# Proceedings of the Eleventh ESSLLI Student Session 

Janneke Huitink \& Sophia Katrenko (editors)

June 20, 2006, Málaga, Spain

## Preface

This volume contains the papers that will be presented at this year's Student Session of the European Summer School in Logic Language and Information (ESSLLI), to be held July 31- August 11, in Málaga, Spain.

The Student Session of ESSLLI provides a platform where students at any level, undergraduates as well as graduates, can present their own work to an interdisciplinary audience. Since its creation in 1996, the popularity of the Student Session is ever increasing. This year, we received a record number of 88 submissions. Of these, 64 papers were submitted for oral presentation, and 24 papers for poster presentation. The overall quality of the submissions was very high. We have selected 16 papers as talks, and 8 as posters. The accepted papers are all included in this volume. They are ordered according to the familiar ESSLLI categories Logic \& Language, Logic \& Computation and Language \& Computation.

We are very grateful to our program committee for their contribution to the organization of the Student Session. We would like to thank the co-chairs in particular for coordinating the reviewing process, and the area experts for their helpful advice. In addition, we are grateful to our reviewers, who have provided our authors with extremely valuable comments on their work. As in previous years, the Student Session is supported by Springer with special awards for the Best oral presentation and Best poster. This year, Springer has generously agreed to increase the Best poster award. We thank them for their support. Finally, our special thanks go to Carlos Areces, Balder ten Cate, Paul Dekker, Judith Gervain, Pablo Lopez, Ernesto Pimentel, Willemijn Vermaat and Andrei Voronkov for all their help in making this edition of the Student Session an inspiring event.

Janneke Huitink
Sophia Katrenko
Chairs of the 2006 Student Session
June 2006, Nijmegen/Amsterdam

## Program Committee

Logic \& Language

Kata Balogh, ILLC, University of Amsterdam
Raffaella Bernardi, Free University of Bozen-Bolzano
Hiroyuki Uchida, University College, London
Logic \& Computation
Ville Nurmi, University of Helsinki
Dmitry Sustretov, Loria, Nancy
Yde Venema, University of Amsterdam

## Language \& Computation

Ann Copestake, University of Cambridge/CSLI, Stanford
Timm Lichte, University of Tübingen
Roelant Ossewaarde, University at Buffalo

## Reviewers

Marco Aiello, Natascha Alechina, Nicholas Allott, Maria Aloni, Avery D. Andrews, Carlos Areces, Alessandro Artale, Hiroshi Aoyama, Roy Bar-Haim, Paul Bennett, Raffaella Bernardi, Henrik Björklund, Andrea Blass, Johan Bos, Peter Bosch, Antal van den Bosch, Torben Braüner, Richard Breheny, Paola Bruscoli, Dirk Bury, Chris Callison-Burch, Balder ten Cate, Jonathan Cohen, Nigel Collier, Jack Copeland, Walter Daelemans, Paul Dekker, Ramón López-Cózar Delgado, Ashwini Deo, Jean-Marc Dewaele, Hans van Ditmarsch, Roy Dyckhoff, Nick Ellis, Peter van Emde Boas, Elisabeth Engdahl, Martin Everaert, Kai von Fintel, Gerhard Fliedner, Jörg Flum, Timothy Fowler, Andrew Gargett, Jelle Gerbrandy, Anastasia Giannakidou, Piklu Gupta, Willem Robert van Hage, Raul Hakli, Chung-hye Han, Lauri Hella, Helen de Hoop, Rosalie Iemhoff, Gerhard Jäger, Samson de Jager, Valentin Jikoun, Dick de Jongh, Laura Kallmeyer, Hans Kamp, Philipp Köhn, Alexander Koller, Marcus Kracht, Angelika Kratzer, Michał Krynicki, Sandra Kübler, Kai-Uwe Kühnberger, Martin Lange, Alex Lascarides, Chungmin Lee, Stéphane Lengrand, Emar Maier, Edgar Meij, Stephan Merz, Jerzy Mycka, Günter Neumann, Joakim Nivre, Juha Oikkonen, Constantin Orasan, Eric Pacuit, Valeria de Paiva, Marc Pauly, Gerald Penn, Douglas Perkins, Ahti-Veikko Pietarinen, George Powell, Alexander Rabinovic, James Rogers, Pamela Rogerson-Revell, Doug Roland, Robert van Rooij, Ricardo Rosati, Leo Ruest, Horacio Saggion, Philippe Schlenker, Sabine Schulte im Walde, Jeremy Seligman, Jennifer Spenader, Lutz Strassburger, Kriszta Szendroi, Hans-Jörg Tiede, Erik Tjong Kim Sang, Robert Truswell, George Tsoulas, Tylman Ule, Willemijn Vermaat, Yannick Versley, Petra Wagner, Maria Angela Weiss, Dag Westerstahl, Holger Wunsch, Adam Wyner, Henk Zeevat, Heike Zinsmeister, Ede Zimmermann, and Sandro Zucchi.

## Contents

I Logic \& Language ..... 1
Houda Anoun ..... 2
Towards a Logical Approach to Nominal Sentences Analysis in Standard Arabic
Michael Franke ..... 14Teleological Necessity and Only
Scott Grimm ..... 27
Subject-Marking in Hindi/Urdu: A Study in Case and Agency
Jana Häussler ..... 39
Disrupted Agreement Checking in Sentence Comprehension
Miltiadis Kokkonidis ..... 51
A Glue/ $\lambda$-DRT Treatment of Resumptive Pronouns
Arndt Riester ..... 64
Only Scalar
Raj Singh ..... 76
Eager for Distinctness
II Logic \& Computation ..... 90
Daniel Andersson ..... 91
An Improved Algorithm for Discounted Payoff Games
Daniel Gorín ..... 99
Hybrid Layering
Jarmo Kontinen ..... 111Zero-One Law and Rational Quantifiers
Sergio Mera ..... 123
Hybrid Logics with Concrete Domains
Magdalena Ortiz ..... 136
A Fully Internalized Sequent Calculus for Hybrid Categorial Logics
Bryan Renne ..... 148Semantic Cut-elemination for Two Explicit Modal Logics
III Language \& Computation ..... 159
Luciana Benotti ..... 160
"Drink me": Handling Actions through Planning in a Text Game Adventure
Alexandru Ceauşu ..... 173
Maximum Entropy Tiered Tagging
Tim van de Cruys ..... 180
Automatically Extending the Lexicon for Parsing
Marieke van Erp ..... 192
Bootstrapping Multilingual Geographical Gazetteers from Corpora
Michael Kaisser ..... 203
Web Question Answering by Exploiting Wide-Coverage Lexical Resources
Tuomo Kakkonen ..... 214
DepAnn - An Annotation Tool for Dependency Treebanks
Fabiana Rosi ..... 226
The Acquisition of Verb Semantics and Morphology in Italian as L2: Towards Computational Simulations
Silke Scheible ..... 239What's the Difference between a Bat and a Mouse? A First Step towards AnsweringComparison Questions in Open-domain QA
Jiří Semecký ..... 250
Automatic Verb Valency Frames Disambiguation for Czech
Reut Tsarfaty263The Interplay of Syntax and Morphology in Building Parsing Models for ModernHebrew

## Part I <br> Logic \& Language

# Towards a Logical Approach to Nominal Sentences Analysis in Standard Arabic 

Houda Anoun<br>Bordeaux 1-LaBRI-Signes (Inria Futurs)<br>anoun@labri.fr


#### Abstract

Standard Arabic (SA) is an extremely rich natural language that has unfortunately received very little interest within computational linguistics literature. We propose in this paper to explore this fertile ground and show the first steps towards the formalization of Arabic syntax and semantics by means of MultiModal Categorial Grammars. We will particularly focus on the analysis of some phenomena related to nominal sentences construction in SA using relevant packages of lexically anchored structural rules.


## 1 Introduction

Standard Arabic (SA) is an extensively used Semitic language: it is considered as the official language of more than twenty countries.
Although SA's grammar has been studied since the $8^{\text {th }}$ century by Sibawayh (Sibawayh 1983) among others, very little thorough research work has been done on the formalization of its syntax and semantics. Moreover, all previous studies related to this field either focus on the syntactic level (e.g., using minimalist program (Abdel 2005; Kremers 2003)), or use linguistic models whose syntax/semantics interface is rather ad-hoc (e.g., using CFG (Haddad 2005)).
To the best of our knowledge, no attempt has yet been made to capture Arabic syntactic and semantic phenomena within a logical setting. We propose in the following paper to explore this promising direction and describe a fragment of SA by means of MultiModal Categorial Grammars (MMCG) (Moortgat 1997). Our initiative is fruitful since it allows us to take advantage of the transparent interface between syntax and semantics guaranteed by Curry-Howard correspondence.
In this paper, we will focus on the analysis of some nominal sentences phenomena in SA (e.g., word order, annexation phenomenon). We will particularly show how to capture such phenomena in an elegant manner using constrained structural reasoning.
Our survey aims at consolidating the interrelation between logic and natural languages which undoubtedly has a dual benefit. On the one hand, the use of a rigorous formalism such as MMCG will help us study SA linguistics in a neat and precise fashion. On
the other hand, the success of this formalization will confirm the linguistic relevance of MMCG model which proves to be readily adaptable to the specificities of a rich language such as SA.

## 2 Preliminaries

### 2.1 What is MMCG?

MMCG (Moortgat 1997; Oehrle 2001) is a logical system well suited to natural language analysis. This model proved relevant as it allows a neat analysis of complex linguistic phenomena occurring in various natural languages (e.g., Dutch verb clusters (Moortgat 1999), multiple Wh-questions in Serbo-Croatian (Vermaat 2005)).

MMCG are composed of two distinct parts: a constant one containing the core logic and a variable one which allows a controlled management of resources.
The first part is represented by a deductive system that describes invariants of grammatical form and meaning composition. Furthermore, it is equipped with a compositional construction process for semantics (represented using a simply typed $\lambda$-calculus) owing to Curry-Howard correspondence. The underlying logical rules of MMCG are universal as they do not depend upon the words (i.e., grammar terminals) of the chosen natural language. They rather express the way in which such words can combine by using their syntactic types. MMCG handles families of binary type constructors $\left(/_{i}, \backslash_{i}, \bullet_{i}\right)$ provided with the structural counterpart $(,)^{i}$ and a set of unary connectives $\left(\diamond_{j}, \square_{j}\right)$ associated with the structure-forming operator $\left\rangle_{j}\right.$. The categorial slashes represent directional forms of the linear implication, they are used to express grammatical incompleteness. For instance, definite adjectives in SA (e.g., al-mufīd-u, i.e., interesting) take the type $\mathbf{n p} \backslash_{0} \mathbf{n p}$ to express their need to combine with a noun phrase to their left to form a modified noun phrase. However, English adjectives (e.g., interesting) take rather the type $\mathbf{n} / 0 \mathbf{n}$ as they will merge with a common noun to their right to yield a modified noun. Moreover, unary operators can be used to encode various features such as morphosyntactic ones (e.g., case, gender, number etc) (Heylen 1999). Indeed, $\square_{j} \mathbf{A}\left(\right.$ resp. $\left.\diamond_{j} \mathbf{A}\right)$ can be seen as a subtype of A with feature $\mathbf{j}$. For example, we can assign the improved type $\square_{s g} \square_{m a} \mathbf{n p} \backslash_{0} \mathbf{n p}$ to the adjective 'al-mufidd-u' to explicitly specify that it requires a singular masculine noun phrase. Thus, we are able to avoid the analysis of some ungrammatical phrases such as **(al-qiṣat-u al$m u f i d d-u$, i.e., the interesting-[mas] story-[fem]) which stems from the combination between the previous adjective and a feminine noun phrase.
The second part of MMCG encapsulates cross-linguistic variation by means of structural rules which allow controlled reconfigurations of contexts. Structural reasoning is constrained thanks to the use of modes of composition that play a crucial role within this framework (Oehrle 2001). Hence, instead of considering a globally available commutativity rule, we can assume that this property is verified by a particular family of connectives marked with mode $\mathbf{c}$. We can control even more this local commutativity by restricting its application to configurations whose left sub-trees are decorated with the structural con-
nective $<>_{j}$. This latter structural rule is formalized by means of the following rewriting rule which can be applied to any appropriate sub-context during a given derivation:

$$
P \diamond(c, j):\left(<\Delta_{1}>_{j}, \Delta_{2}\right)^{c} \rightarrow\left(\Delta_{2},<\Delta_{1}>j\right)^{c}
$$

For the sake of legibility, we will rather present structural rules using their axiomatic form as shown below:

$$
P \diamond(c, j): \diamond_{j} A \bullet_{c} B \rightarrow B \bullet_{c} \diamond_{j} A
$$

Structural reasoning is a powerful tool; it can be used to capture the flexibility of wordorder in a neat fashion (Vermaat 2005). Indeed, it allows to relate the different structural positions that a word can occupy within a phrase, thus limiting its lexical ambiguity. Consequently, the application of relevant structural rules makes it possible to derive all the plausible configurations of a given clause from a single type assignment describing the canonical behavior of each one of its components. This asset will be used subsequently to account for word-order variation within SA nominal sentences.

In this paper, we use the natural deduction presentation of MMCG. For the purpose of completeness, the logical rules of this system are presented in Figure 1.1. We recall that the deduction rules operate on sequents like ( $\Gamma \vdash \mathrm{x}: \mathrm{A}$ ), where $\Gamma$ is a structured binary context, $\mathbf{A}$ is a syntactic category and $\mathbf{x}$ is a simply typed $\lambda$-term that encapsulates the derivational semantics. The interested reader can find an in-depth survey of this deductive system in (Moortgat 1997).

$$
\begin{array}{ccc}
\frac{\Delta \vdash p: A \bullet_{i} B \Gamma\left[(a: A, b: B)^{i}\right] \vdash y: C}{x: A \vdash x: A} A x & \frac{\Delta \vdash}{\Gamma[\Delta] \vdash y\left[a:=\Pi_{1}(p), b:=\Pi_{2}(p)\right]: C} \bullet_{i} E & \frac{\Gamma \vdash f: A /_{i} B \Delta \vdash b: B}{(\Gamma, \Delta)^{i} \vdash(f b): A} /_{i} E \\
\frac{\Gamma \vdash b: B \Delta \vdash f: B \backslash_{i} A}{(\Gamma, \Delta)^{i} \vdash(f b): A} \backslash_{i} E & \frac{\Gamma \vdash a: A \Delta \vdash b: B}{(\Gamma, \Delta)^{i} \vdash(a, b): A \bullet_{i} B} \bullet_{i} I & \frac{(\Gamma, x: B)^{i} \vdash f: A}{\Gamma \vdash \lambda x \cdot f: A /_{i} B} /_{i} I \\
\frac{(x: B, \Gamma)^{i} \vdash f: A}{\Gamma \vdash \lambda x \cdot f: B \backslash_{i} A} \backslash_{i} I & \frac{\Delta \vdash x: \diamond_{j} A \Gamma\left[<a: A>_{j}\right] \vdash c: C}{\Gamma[\Delta] \vdash c[a:=x]: C} \diamond_{i} E & \frac{\Gamma \vdash a: \square_{j} A}{<\Gamma>_{j} \vdash a: A} \square_{j} E \\
\frac{<\Gamma>_{j} \vdash a: A}{\Gamma \vdash a: \square_{j} A} \square_{i} I & \frac{\Gamma \vdash a: A}{<\Gamma>_{j} \vdash a: \diamond_{j} A} \diamond_{j} I & \frac{\left(\Delta_{1} \xrightarrow{R} \Delta_{2}\right) \Gamma\left[\Delta_{2}\right] \vdash x: C}{\Gamma\left[\Delta_{1}\right] \vdash x: C} S_{R}
\end{array}
$$

Figure 1.1: Natural deduction rules for Multimodal Logic

### 2.2 Some Words about SA

We present below two important characteristics of SA which are likely to help with the comprehension of our study. More details about SA grammar can be found in (Blachère 1994; Ryding 2005; Arrajihi 1975).

- SA is a highly inflectional language: cases are generally marked by means of suffixes (e.g., $-u$ : nominative, $-a$ : accusative and $-i$ : genitive). Moreover, definiteness indicators are incorporated within nouns. In fact, the prefix $a l-{ }^{-1}$ is used to form definite nouns while a suffix $-n$ marks indefinite ones. For instance, walad-u-n (i.e., a boy) is an indefinite nominative noun, whereas al-walad-i (i.e., the boy) is a definite genitive one.
- SA is a language with mixed word-order: in fact, word-order in SA can be very flexible in some constructions (e.g., all of SVO, VOS, OVS and VSO orders are generally plausible) but so strict in others (e.g., adjectives always follow their modified nouns). It will be interesting to use controlled structural reasoning to deal with this diversity.

SA writing is built upon a specific alphabet and its direction is from right to left, but for the sake of readability, we will rather use the transliteration given by arabtex ${ }^{2}$ package.

## 3 Syntax \& Semantics of Nominal Sentences in SA

### 3.1 Basic Nominal Sentences Analysis

In contrast with languages such as English or French, we can build nominal sentences in SA that contain no verb (there is no copulative verb in Arabic such as 'to be' or 'to remain'). This construction is frequently used in other Semitic languages, notably in Hebrew (e.g., ha-sepr gadol, i.e., the book is big).

Nominal sentences give descriptions or definitions which are independent of time. They are composed of two components namely a topic realized by a noun phrase with nominative case and a comment which can be either an indefinite noun modifier, an indefinite noun or a prepositional sentence. Examples of grammatical and ungrammatical basic nominal sentences in SA are shown below:
(1) al-mantiq-u/*mantiq-u-n mufīd-u-n
(the) logic-[nom]/logic-[nom] interesting-[ind]
'Logic is interesting'
(2) mufìd-u-n al-mantiq-u
interesting-[ind] logic-[nom]
'Logic is interesting'
(3) al-walad-u fī 'l-bayti
the boy-[nom] in the house
'The boy is in the house'

[^0](4) fī 'l-bayti 'l-walad-u in the house the boy-[nom]
'The boy is in the house'
(5) f̄̂ 'l-bayti walad-u-n, *walad-u-n $f_{\bar{\imath}}{ }^{\prime} l$-bayti in the house a boy
'A boy is in the house'
A definite topic (cf. ex. 1-4) can be placed either before or after its comment. Hence, the canonical order puts the emphasis on the definite topic (ex. 1\&3), while the inverse order makes it possible to underline the comment (ex. 2\&4). However, an indefinite topic can only be used with prepositional comments and should be placed at the end of the sentence (cf. ex. 1\&5). In fact, the predicate which occurs after an indefinite noun is considered as an attributive adjective rather than a comment; the resulting construction is than a nominal phrase (of type $\mathbf{n p}$ ) instead of a nominal sentence (of type $\mathbf{s}^{3}$ ).
To account for the previous constraints that manage word-order between the topic and its comment in SA, we use controlled structural reasoning. Firstly, we assign to each constituent a single refined type which describes its canonical syntactic behavior and encapsulates its relevant morphosyntactic features ${ }^{4}$ by means of $\square$ operator:

| Definite Topic | Indefinite Topic | Prep-Comment | Other Comments |
| :---: | :---: | :---: | :---: |
| al-mantiq- $u$, al-walad- $u$ | mantiqu- $n$, walad- $u-n$ | $f^{\bar{\imath}}{ }^{\prime} l b a y t i$ | muflidu- $n$ |
| $\square_{\text {def }} \square_{\text {nom }} \mathrm{np}$ | $\square_{\text {ind }} \square_{\text {nom }} \mathrm{np}$ | $\mathrm{s} / c^{\prime} \square_{\text {nom }} \mathrm{np}$ | $\square_{\text {def }} \square_{\text {nom }} \mathrm{np} \backslash_{c} \mathrm{~s}$ |

Composition modes $c$ and $c^{\prime}$ used in this lexicon are governed by the postulates below:

$$
P(c): \mathrm{A} \bullet_{c} \mathrm{~B} \longrightarrow \mathrm{~B} \bullet_{c} \mathrm{~A} \quad P \diamond\left(c^{\prime}, d e f\right):\left(\diamond_{d e f} \mathrm{~A}\right) \bullet_{c^{\prime}} \mathrm{B} \longrightarrow \mathrm{~B} \bullet_{c^{\prime}}\left(\diamond_{d e f} \mathrm{~A}\right)
$$

We notice that both modes $c$ and $c^{\prime}$ are used to add local commutativity to our system. The commutativity introduced by mode $c$ makes it possible to combine two expressions whose respective order is unconditionally free. In that case, both type constructors $/ c$ and $\backslash_{c}$ represent the same connective namely the non associative linear implication. On the other hand, mode $c^{\prime}$ 'introduces commutativity in a constrained fashion thanks to the use of the control operator $\diamond_{\text {def }}$. Indeed, its associated structural rule, $P \diamond\left(c^{\prime}, d e f\right)$, cannot be applied unless the first combined expression is definite.
If we consider the grammar provided with the lexicon and postulates above, then we are able to derive the correct examples and predict the ill-formedness of the ungrammatical ones in a straightforward manner. In fact, the underlined idea is as follows. When the comment is a noun modifier (or an indefinite noun), it combines with its definite topic using a commutative mode $c$, thus allowing the envisaged free word-order. On the other hand, a prepositional comment always searches for its topic to the right and combines with it using mode $c^{\prime}$. As the definiteness of the topic required by the prepositional comment is

[^1]underspecified, a potential topic cannot enter the derivation until its definiteness feature is checked (i.e., by means of $\square E$ rule). If the topic is definite (i.e., at this stage, it should be decorated by the structural operator $<>_{d e f}$ ), then it can move to the beginning of the sentence thanks to $P \diamond\left(c^{\prime}, d e f\right)$ postulate; otherwise, no displacement proves to be possible. These steps are illustrated in the derivation of sentence 3 below:
$$
\frac{\frac{\overline{\imath^{\prime}} l \text { bayti } \vdash f: s /{ }_{c}^{\prime} \square_{\text {nom }} n p}{} A x \quad \frac{\overline{\text { alwaladu } \vdash w: \square_{\text {def }} \square_{\text {nom }} n p}}{\frac{<a l w a l a d u>_{\text {def }} \vdash w: \square_{\text {nom }} n p}{<\square_{\text {def }} E} /{ }_{c^{\prime}} E}+\frac{\left(f \imath^{\prime} l b a y t i,<\text { alwaladu }>_{\text {def }}\right)^{c^{\prime}} \vdash(f w): s}{\left(<\text { alwaladu }>_{\text {def }}, f^{\prime} \bar{\imath}^{\prime} l b a y t i\right)^{c^{\prime}} \vdash(f w): s} P \diamond\left(c^{\prime}, \text { de } f\right)}{}
$$

The derivational semantics of a sentence is computed in tandem with its syntactic derivation thanks to Curry-Howard correspondence. For instance, the deduction associated to sentence 3 yields the term $(f w)$, where $f$ (resp. $w$ ) represents the semantics of al-walad$u$ (resp. fi${ }^{\prime}$ 'lbayti). The final semantics of this sentence, namely the logical formula in $(\iota$ man, $\iota$ house $)^{5}$, results from substituting each formal variable representing a linguistic entity by its lexical semantics.

### 3.2 Towards Complex Nominal Sentences

## Annexation Phenomenon

All topics so far have been simple noun phrases. We will see in this section how we can enhance our nominal sentences by using compound topics.
In SA, we can form compound noun phrases by means of annexation phenomenon (Blachère 1994; Kremers 2003). These compound nouns have the following form 'cn=n $n_{1} n_{2} \ldots n_{k}$ ' ( $\mathrm{k} \geq 2$ ), where each $n_{j}(1 \leq \mathrm{j}<\mathrm{k})$ is a noun in construction state (i.e., which has neither the definite nor the indefinite indicator), whereas $n_{k}$ is a noun phrase (either definite or indefinite). The resulting compound noun 'cn' inherits the definiteness feature from $n_{k}$, whereas its case is the same as $n_{1}$ (all the other nouns $n_{j}(j \geq 2)$ take the genitive case). Here are some examples of noun phrases built using annexation:
(6) っibn-u/*al->ibn-u 'l-mudarris-i
son-[nom]/the son-[nom] the teacher-[gen]
'the son of the teacher'
kitāb-u/*kitāb-u-n mudarris-i-n
book-[nom]/a book-[nom] a teacher-[gen]
'the book of a teacher'

[^2](8) kitāb-u $\quad$ :ibn-i $\quad$ 'l-mudarris-i
book-[nom] son-[gen] the teacher-[gen]
'the book of the teacher's son'

To capture annexation phenomenon within MMCG, we assign a suitable syntactic type to each one of the three classes of SA nouns, namely al-nouns (definite simple nouns), csnouns (nouns in construction state) and nn-nouns (indefinite simple nouns).

| al-nouns | nn-nouns | cs-nouns |
| :---: | :---: | :---: |
| $\square_{a l} \square_{\text {case }} \mathrm{np}$ | $\square_{n n} \square_{\text {case }} \mathrm{np}$ | $\square_{c s}\left(\square_{\text {case }} \mathrm{np} / 0 \square_{\text {gen }} \mathrm{np}\right)$ |

Hence, al-nouns and nn-nouns are both complete and self-contained as they can be used in several contexts (as subjects or topics etc). Nevertheless, cs-nouns are incomplete; they are only used to build compound noun phrases. They are assigned a functional type since they require to combine with a noun phrase to their right by means of a rigid composition mode 0 (i.e., non-associative and non-commutative mode) to yield a complete expression. We consider the grammar that supports the following package of postulates, $\mathcal{R}=\mathrm{K}($ def $) \cup$ $\mathrm{K}(i n d) \cup \operatorname{Inc}(d e f, a l) \cup \operatorname{Inc}(i n d, n n)$, where:

$$
\mathrm{K}(\mathrm{j}): \diamond_{j}\left(\mathrm{~A} \bullet_{0} \mathrm{~B}\right) \longrightarrow \diamond_{c s} \mathrm{~A} \bullet_{0} \diamond_{j} \mathrm{~B} \quad \operatorname{Inc}(\mathrm{i}, \mathrm{j}): \diamond_{i} \mathrm{~A} \longrightarrow \diamond_{j} \mathrm{~B}
$$

The structural rule $\mathrm{K}($ def) (resp. $\mathrm{K}($ ind $)$ ) is a kind of strong distributivity postulate (Heylen 1999). Intuitively, this postulate stipulate that a complex constituent is definite (resp. indefinite) if its head is in construction state (e.g., cs-noun) and its complement is definite (resp. indefinite). However, the rule $\operatorname{Inc}(d e f, a l)$ (resp. $\operatorname{Inc}(i n d, n n)$ ) is nothing else but an inclusion principle. It states that all al-nouns (resp. nn-nouns) are inevitably definite (resp. indefinite).

Owing to the package $\mathcal{R}$, only well-formed compound nouns ' $n_{1} \ldots n_{k}$ ' can be derived, they are assigned either type ' $\square_{\text {def }} \square_{c_{1}} n$ ' if $n_{k}$ is an al-noun or type ' $\square_{\text {ind }} \square_{c_{1}}$ np' if $n_{k}$ is a nn-noun ( $c_{1}$ is the case of $n_{1}$ ). In fact, the package $\mathcal{R}$ makes it possible to apply the lock/key strategy (Moortgat 1999) which, in our case, proceeds as the following. Firstly, recursive rules $\mathrm{K}(\mathrm{j})$ are used to open each lock $\square_{c s}$ surrounding nouns $n_{i}(1 \leq \mathrm{i}<\mathrm{k})^{6}$, thus checking that they are all in construction state. Secondly, Inc(i, $j$ ) rules are used to check definiteness feature of noun $n_{k}$. Finally, the derivation can be completed by a succession of /o elimination. Therefore, it is easy to parse a compound nominal sentence such as ' $s_{1}=$ 'ibnu 'l-mudarris- $i f \bar{\imath}$ 'lbayti' (i.e., the man's son is in the house), the crucial steps of its topic's derivation are sketched below:

[^3]\[

$$
\begin{aligned}
& \frac{\left(<\text { ibnu }: \square_{c s}\left(\square_{\text {nom }} n p / 0 \square_{g e n} n p\right)>_{c s},<^{\prime} l \text { mudarrisi }: \square_{a l} \square_{g e n} n p>_{a l}\right)^{0} \vdash(\text { im }): \square_{\text {nom }} n p}{\left(<i b n u: \square^{\prime}\right.} \operatorname{Inc}(\text { def, al }) \\
& \overline{\left(<i b n u: \square_{c s}\left(\square_{\text {nom }} n p / 0 \square_{\text {gen }} n p\right)>_{c s},<^{\prime} l \text { lmudarrisi }: \square_{a l} \square_{\text {gen }} n p>_{\text {def }}\right)^{0} \vdash(\text { im }): \square_{\text {nom }} n p} K(\text { def }) \\
& \frac{\left\langle\left(\text { ibnu }: \square_{c s}\left(\square_{\text {nom }} n p / 0 \square_{\text {gen }} n p\right),{ }^{\prime} \text { lmudarrisi }: \square_{a l} \square_{\text {gen }} n p\right)^{0}>_{\text {def }} \vdash(\text { im }): \square_{\text {nom }} n p\right.}{\left(\text { ibnu }: \square_{c s}\left(\square_{\text {nom }} n p /{ }_{0} \square_{\text {gen }} n p\right),{ }^{\prime} \text { lmudarrisi }: \square_{\text {al }} \square_{\text {gen }} n p\right)^{0} \vdash(\text { im }): \square_{\text {def }} \square_{\text {nom }} n p} I
\end{aligned}
$$
\]

Finally, we use higher order $\lambda$-terms ${ }^{7}$ to represent the lexical semantics of each class of SA nouns. For instance, singular SA nouns are assigned the following meanings:

| cs-noun: | $\lambda P_{(e \rightarrow t) \rightarrow t} \lambda Q_{e \rightarrow t} \cdot \exists \mathrm{x} \cdot \mathbf{w}_{\text {pred }}(\mathrm{x}) \wedge \mathrm{Q}(\mathrm{x}) \wedge$ <br> $w_{c s}(\operatorname{Rel})$ |
| :---: | :---: |
| $\mathrm{P}\left(\lambda \mathrm{y} . \operatorname{Rel}(\mathrm{x}, \mathrm{y}) \wedge \forall \mathrm{z} \cdot \mathbf{w}_{\text {pred }}(\mathrm{z}) \wedge \operatorname{Rel}(\mathrm{z}, \mathrm{y}) \Rightarrow \mathrm{z}=\mathrm{x}\right)$ |  |,

Note that the semantics of each noun $w_{x}$ is based upon a predicate $\mathbf{w}_{\text {pred }}$ representing a set of individuals that share a specific property (e.g., teacher, son...). Moreover, the meaning of cs-nouns closely depends on a relation Rel that binds these individuals to their annexed objects (e.g., Rel can be either a relation of possession, family-ship ...). Lastly, it is worth noticing that both cs-nouns and al-nouns meanings require uniqueness conditions. For instance, the semantics of a cs-noun $w_{c s}$ stipulates that the intersection between the set of individuals verifying the property $\mathbf{w}_{\text {pred }}$ and the range of entities connected to the annexed object by the relation Rel is nothing else but a singleton.
Using the previous lexical semantics, one can easily check that the meaning of sentence $s_{1}$ is represented by the following first-order formula:

$$
\begin{aligned}
& \exists \mathrm{x} . \operatorname{son}(\mathrm{x}) \wedge \operatorname{in}(\mathrm{x}, \iota \text { house }) \wedge \exists \mathrm{y} \text {. teacher }(\mathrm{y}) \wedge \text { family-ship }(\mathrm{x}, \mathrm{y}) \wedge \\
& (\forall \mathrm{z} \cdot \operatorname{son}(\mathrm{z}) \wedge \text { family-ship }(\mathrm{z}, \mathrm{y}) \Rightarrow \mathrm{z}=\mathrm{x}) \wedge(\forall \mathrm{z} \cdot \operatorname{man}(\mathrm{z}) \Rightarrow \mathrm{z}=\mathrm{x})
\end{aligned}
$$

Our study can also be applied to account for nouns built using annexation in Hebrew. In fact, this phenomenon is managed by the same range of syntactic principles in both Hebrew and SA as shows the following example quoted from (Wintner 2000):
(9) yaldei mnahhel taxnot ha-rakkebt
children-[cs] manager-[cs] stations-[cs] the train
'the train stations manager's children'

## Adjectives

We distinguish two classes of adjectives in SA, namely attributive adjectives and predicative ones. Attributive adjectives are used to modify definite and indefinite nouns, they can be involved in the construction of enhanced topics. These adjectives agree with the head they modify on number, gender, case and definiteness. However, predicative adjectives are used as comments within nominal sentences; they are always indefinite and they agree with their

[^4]topic on gender and number. We present in the following some examples of well-formed and ill-formed SA clauses involving the use of adjectives:
(10) sibnu 'l-mudarris-i 'l-q̆amīl-u
son-[nom] the teacher-[gen] beautiful-[nom]
'The beautiful teacher's son'
(11) * ${ }^{*}$ ibn-u $\quad$ ğamīl-u $\quad$ l-mudarris- $i$
son-[nom] beautiful-[nom] the teacher-[gen]
(12) 'l-mudarris-u 'l-ğamīl-u 'l-‘aynayn-i/*lawn-u 'l-‘aynayn-i
the teacher-[gen] beautiful-[nom] the eyes-[gen]/*color-[nom] the eyes
'the teacher with beautiful eyes'
(13) 'l-mudarris-u ğamīl-u 'l-‘aynayn-i the teacher-[gen] beautiful-[nom] the eyes-[gen] 'the teacher has beautiful eyes'
Unlike some languages where the word-order between adjectives and their head is relatively free (e.g., French), SA attributive adjectives are post-modifiers, they always occur after the noun phrase they modify. Moreover, when the noun phrase is built using annexation, the adjective should be placed at the end of the whole construction (cf. ex. 10) since nouns in construct state cannot be modified (cf. ex. 11). On the other hand, we are able to build enhanced adjectival phrases thanks to annexation phenomenon (cf. ex. 12\&13). In SA, an adjectival phrase has two constituents. The first one (i.e., the head) is either a cs-adjective (adjective in construct state) or an al-adjective (simple definite adjective) whereas the second one (i.e., the complement) is nothing else but a genitive al-noun. Hence, it is forbidden to build an adjectival phrase by combining an adjective with a compound noun phrase (cf. ex. 12).
The following table recapitulates the various syntactic constraints which manage the use of SA adjectives. For the sake of legibility, we only focus on definiteness and case agreements between adjectives and their heads; gender and number agreements can be added in a straightforward fashion.

| SA adjectives | al-adjectives | nn-adjectives | cs-adjectives |
| :---: | :---: | :---: | :---: |
| Predicative | $\times$ | $P_{2}=\square_{\text {def }} \square_{\text {nom }} n p \backslash_{c} s$ | $P_{3}=P_{2} /{ }_{0} \square_{a l} \square_{\text {gen }} n p$ |
| Attributive | $A_{1}=\square_{\text {def }} \square_{\text {case }} n p \backslash_{0} \square_{\text {def }} \square_{\text {case }} n p$ | $A_{2}=\square_{\text {ind }} \square_{\text {case }} n p \backslash_{0} \square_{\text {ind }} \square_{\text {case }} n p$ | $A_{3}=A_{2} /{ }_{0} \square_{a l} \square_{\text {gen }} n p$ |
|  | $A_{1}^{\prime}=A_{1} /{ }_{0} \square_{a l} \square_{\text {gen }} n p$ |  |  |

Thanks to this type assignment, we are able to handle the different uses of SA adjectives in a rigorous manner. Lexical ambiguity is necessary to account for the distinct syntactic behaviors of some adjectives. For instance, al-adjectives can directly combine with their definite nouns (e.g., ' $l-g ̆ a m \bar{\imath} l-u$ in ex. 10 ); in that case we use the syntactic type $A_{1}$. Otherwise, these adjectives can initially form an adjectival phrase by combining with a genitive al-noun to their right and then merge with their head to the left (e.g., ' $l$-ğamāl-u in ex. 12). This latter case is dealt with by means of the second syntactic type $A_{1}^{\prime}$.

The semantics of SA adjectives resembles the semantics of English adjectives at a great extent. The interested reader can find more details about this field in (Chierchia 2000).

## Negation

In contrast with some languages where negation is a slightly complex phenomenon (e.g., negation in French involves a discontinuous constituent ne ... pas), nominal sentences negation in SA is obtained in a straightforward fashion. This is done by adding the particle $m \bar{a}$ at the start of the sentence as illustrated below:

```
mā al-m\overline{a}k-u manbc-u 'lassa`ādat-i
not money-[nom] source-[nom] the happiness-[gen]
'Money is not the source of happiness'
```

Consequently, we can easily deal with nominal sentences negation in MMCG by assigning the syntactic type $\mathbf{s} / 0 \mathrm{~s}$ to the particle $m \bar{a}$.
The negation of SA nominal sentences can also be obtained by using some external governors such as laysa (Blachère 1994; Arrajihi 1975). This latter particle also precedes the nominal sentence but it changes the default case of its comment as shows the following example:
(15) laysa al-māl-u manbac-a $\quad$ lssa ${ }^{-} \bar{a} d a t-i$
not money-[nom] source-[acc] the happiness-[gen]

Since the external governor laysa does not subcategorize for a whole nominal sentence, we will not be concerned with its formal study in this work.

## Conclusion \& Prospects

In this paper, we presented the first steps towards the syntactic and semantic analysis of SA using MMCG formalism. In particular, we showed that we can deal with simple and compound nominal sentence constructions using appropriate structural modules. Moreover, the meaning of these sentences is obtained in a compositional fashion thanks to the use of simply typed $\lambda$-calculus and Curry-Howard correspondence. The complete study also includes a logical treatment of the asymmetry between the different forms of verbal phrases in SA. This latter work will be described in a forthcoming paper.
Our ultimate goal is to build a compact MMCG grammar which handles a decent fragment of SA containing at least the frequent linguistic phenomena, e.g., ellipsis, coordination, wh-questions and anaphora. Moreover, we intend to extend our work and deal with some Arabic dialects used in different speech communities (e.g., Moroccan and Egyptian). We also plan to compare Arabic phenomena with other Semitic languages, notably Hebrew. We hope that the present survey will constitute the pillar of the formal study of crosslinguistic variation between the various forms of Arabic Language.

## Acknowledgements

I would like to thank my father Abdelhamid Anoun as well as my colleagues Youssef Tahir and Dina El kassas for fruitful discussions about SA linguistics. I am also very grateful to the anonymous ESSLLI reviewers for their useful comments and constructive suggestions. Finally, I am deeply indebted to my parents, my sister Soumaya and my fiancé Réda for their psychological support and everlasting care. Of course, all remaining errors and omissions are mine.

## Bibliography

Abdel-Hafiz, A.: Verb Agreement in SA: An analysis in the Minimality Program Journal of Language and Linguistics, Vol 4, (2005)

Arrajihi, A.: alttaṭbīqu 'lnaḥw̄̄ Dar alnnahdat ak arabiyat eds, (1975)
Blachère, R.: Eléments de l'Arabe classique Maisonneuve Et Larose eds, ISBN 2706811293, (1994)

Chierchia, G., McConnell-Ginet, S.: Meaning and Grammar, An Introduction to Semantics The MIT press, (2000)
Haddad, B., Yaseen, M.: A Compositional Approach Towards Semantic Representation and Construction of Arabic LACL, (2005)
Heylen, D.: Types and Sorts Resource Logic for Feature Checking PhD thesis, Utrecht,1999.
Kreemers, J.: The Arabic noun phrase: a minimalist approach PhD thesis, Nijmegen University, (2003)
Moortgat, M.: Categorial Type Logic Chapter 2 in Van Benthem \& ter Meulen (eds), Handbook of Logic and Language, (1997)
Moortgat, M.: Meaningful patterns In Gerbrandy, Marx, de Rijke and Venema (eds), Amsterdam, (1999)
Oehrle, R.T.: Multi-Modal Type Logical Grammar Borsley, R. \& Borjars, K. (eds), Non transformational Syntax, Blackwell, (2001)

Ryding, K.: A Reference Grammar of Modern Standard Arabic Cambridge University Press, (2005)
Sibawayh: ‘ālamu 'l-kitābi Bayrout edition, (1983)
Vermaat, W.: The Logic of Variation, A cross-linguistic account of wh-question formation PhD thesis, Utrecht, (2005)
Wintner, S.: Definiteness in the Hebrew Noun Phrase Journal of Linguistics, Vol. 36, (2000)

# Teleological Necessity and Only 

Michael Franke<br>Universiteit van Amsterdam, ILLC

M.Franke@uva.nl


#### Abstract

According to von Fintel and Iatridou (2005a) teleological sufficiency statements, i.e. sentences of the form "In order for $p$, only have to $q$ ", pose a problem of compositionality: it is not clear how to account for their intuitive meaning in terms of a standard theory of only and the meaning of the embedding sentence "In order for $p$, have to $q$ ". Therefore von Fintel and Iatridou resort to a non-standard analysis of only. The aim of this paper is to show that this is not necessary.


## 1 Introduction

Intuitively (1) means that going to Haarlemmerstraat is a way of getting German bread which is comparatively or unexpectedly easy for a means of getting German bread, or, in other words, that going to Haarlemmerstraat is sufficient for achieving the given goal, for reasons of which I will speak of a teleological sufficiency statement (TSS).
(1) In order to get German bread, you only have to go to $[\text { Haarlemmerstraat }]_{F}$. In order for $p$, only have to $q$.

The question to be asked then is how this meaning can be derived from the meaning of only paired with the meaning of the embedding sentence (2), the so-called prejacent, a teleological necessity statement (TNS)?
(2) In order to get German bread, you have to go to Haarlemmerstraat.

In order for $p$, have to $q$.
A naive application of a standard account of the meaning contribution of only seems incapable of answering this question. Recall that, according to Horn's influential approach, a sentence like (3a) semantically means (3b) and either strongly presupposes (3c) (Horn 1969) or weakly presupposes (3d) (Horn 1996).
(3) a. Only $[\text { Hans }]_{F}$ came.
b. Nobody other than Hans came.
$\forall x \in \operatorname{Alt}($ Hans $)(\neg \operatorname{came}(x))$
c. Hans came.
came(Hans)
d. Somebody came.
$\exists x$ (came $(x))$

If we naively apply the same idea to TSSs, we would get that (1) semantically means (4a) and either strongly presupposes (4b) or weakly presupposes (4c).
(4) a. In order to get German bread, you don't have to go anywhere else than to HS. $\forall q \in \operatorname{Alt}(\mathrm{HS}) \neg \mathrm{Nec}(\mathrm{GB}, q)$
b. In order to get German bread, you have to go to Haarlemmerstraat.

Nec (GB, HS)
c. There is something that you have to do in order to get German bread.
$\exists q(\operatorname{Nec}(\mathrm{~GB}, q))$
Yet as von Fintel and Iatridou (2005a) notice, the naive approach faces what they call the prejacent problem; if other ways of getting German bread exist, the strong presupposition in (4b) is false, although intuitively (1) does not seem to suffer from a presupposition failure in such cases. The strong presupposition therefore seems too strong for TSSs. Yet a weak presupposition (4c) seems too weak. The presupposition that something is necessary for getting German bread, together with the semantic meaning (4a) does not capture our meaning intuition that (1) says that going to Haarlemmerstraat is sufficient for getting German bread. Suppose that, if you bring your purse, German bread is to be had in Leidsestraat (LS) and Utrechtsestraat (US), but not in Haarlemmerstraat (HS). Then (4a) is true, because for a set of alternative $\operatorname{Alt}(\mathrm{HS})=\{\mathrm{LS}, \mathrm{US}\}$ we get that $\forall x \in \operatorname{Alt}(\mathrm{HS}) \neg \mathrm{Nec}(\mathrm{GB}, x)$ is true; all alternatives to going to Haarlemmerstraat are not necessary for getting German bread. As bringing your purse is necessary, we do not infer from the weak presupposition and the semantics that going to Haarlemmerstraat is sufficient for getting German bread and so a weak presupposition seems too weak to account for intuitions. So neither a strong nor a weak presupposition seems appropriate in a naive application of an established approach of the meaning of only to account for intuitions.

## 2 Previous Analyses

### 2.1 Modal Split

In order to solve the prejacent problem von Fintel and Iatridou (2005a) suggest a modalsplit analysis of only. In analogy to languages like French where a TSS such as (1) is expressed by a separate negation and an exceptive quantifier $(\exists x \in \operatorname{Alt}(X))$ as in (5), von Fintel and Iatridou suggest to regard only as analogously comprising these two elements.
(5) tu n' as qu' à aller à HS.
you not have except-to go to HS
"You only have to go to HS."
It is then argued that the prejacent problem can be solved if the necessity modal in (1) takes intermediate scope in between negation and the exceptive quantifier, again in analogy to the French example. The proposed semantic meaning of a TSS with modal-split only is then the following:

$$
\begin{align*}
\text { 'In order for } p \text {, only have to } q . & \approx \neg \square_{p} \exists q^{\prime} \in \operatorname{Alt}(q) q^{\prime} \text { is true } \\
& \approx \neg \forall w\left(w \in p \rightarrow \exists q^{\prime} \in \operatorname{Alt}(q) w \in q^{\prime}\right) \\
& \approx \exists w\left(w \in p \wedge \forall q^{\prime} \in \operatorname{Alt}(q) w \notin q^{\prime}\right) \tag{2.1}
\end{align*}
$$

Part of the prejacent problem is solved, because according to (2.1) sentence (1) is no longer predicted true in case there are two alternatives to going to Haarlemmerstraat where German bread can be bought. Yet in order to account for the meaning component of (1) that German bread is on sale in Haarlemmerstraat, it has to be made sure that the witness of (2.1) is actually a world where we went to Haarlemmerstraat. This can be derived if we assume that the set of considered means $\operatorname{Alt}(q) \cup\{q\}$ exhausts the goal-worlds. This exhaustifity requirement is fulfilled, according to von Fintel and Iatridou, by a weak presupposition which is of the form (2.2).

$$
\begin{equation*}
\square_{p} \exists q^{\prime} \in \operatorname{Alt}(q) \cup\{q\} q^{\prime} \text { is true } \tag{2.2}
\end{equation*}
$$

Taken together (2.1) and (2.2) let us derive the overall meaning of a TSS in (2.3).

$$
\begin{equation*}
\exists w \in W\left(w \in p \wedge w \in q \wedge \forall q^{\prime} \in \operatorname{Alt}(q) w \notin q^{\prime}\right) \tag{2.3}
\end{equation*}
$$

Although this analysis overcomes the noted prejacent problem, it still suffers from some insufficiencies. It is not only that the only genuine argument for modal split of only is that it helps solve the prejacent problem, and that therefore, if possible, a non-split treatment of only would clearly be preferred, but it is also that (2.3) is too weak to account for sufficiency and that the scalar meaning component, $q$ 's relative ease for achieving $p$, are not captured. These latter two points of criticism have been taken up by Huitink (2005) and Krasikova and Zhechev (2005) respectively and form the basis of their alternative accounts.

### 2.2 Modal Concord

Huitink (2005) notices that von Fintel and Iatridou's prediction (2.3) does not capture the transitivity of TSSs. Intuitively, the following argument is clearly valid, but this intuition is not borne out in (2.3):

In order to pass logic, you only have to be able to do derivations.
In order to be able to do derivations, you only have to know the rules of thumb.
$\therefore$ In order to pass logic, you only have to know the rules of thumb.
At the heart of Huitink's criticism lies the realization that the existential in (2.3) is too weak to capture sufficiency. Von Fintel and Iatridou (2005a) intended to parry Huitink's charge by pointing out the difference between (6a) and (6b).
(6) a. In order to find out what Morris is working on, you only have to go to the SC.
b. You only have to go to the SC, and you'll find out what Morris is working on.

Whereas (6a) does not mean that it is a direct and immediate result of going to the SC that the addressee finds out about Morris' work, this is the intuitive meaning of (6b). Hence, so the conclusion of von Fintel and Iatridou, TSSs express something short of sufficiency. Yet although this indeed seems to be the case, the worry remains that von Fintel and Iatridou's analysis falls too short of sufficiency. The problem clearly surfaces in the erroneous predictions about sentences such as (7).
(7) In order for this fair coin to come up heads, you only have to toss it.

For all we know about fair coins, (7) should be false, but is rendered true by the analysis of von Fintel and Iatridou in (2.3).

In the light of the shortcomings of von Fintel and Iatridou's analysis Huitink proceeds to propose an alternative account of TSSs. She proposes to see a modal concord phenomenon in only have to constructions. only is considered a universal modal quantifier alongside of have to. Since intuitively in (1) only one universal modal quantifier seems to be operative, Huitink suggests to analyze only have to as a modal concord phenomenon where only simply reverses the relation in (2.4) to yield (2.5).

$$
\begin{align*}
\text { 'In order for } p \text {, have to } q . & \approx \forall w(w \in p \rightarrow w \in q)  \tag{2.4}\\
\text { 'In order for } p \text {, only have to } q . & \approx \forall w(w \in q \rightarrow w \in p) \tag{2.5}
\end{align*}
$$

This analysis can account for the transitivity of TSSs, as desired. But again it relies on a non-standard treatment of only, may even seem ad hoc from a distance and clearly raises the question whether it is not actually too strong. It is not the case that (1) means that in all worlds where one goes to Haarlemmerstraat automatically or immediately German bread is obtained. So it seems that an appropriate intermediate notion of sufficiency has to be met to account for the meaning of (1) situated in between the too weak notion of von Fintel and Iatridou and the too strong notion of Huitink.

### 2.3 Scalar Only

Both von Fintel and Iatridou and Huitink pay attention to, but do not focus on the intuitive meaning aspect of (1) that going to Haarlemmerstraat is comparatively easy. Krasikova
and Zhechev (2005) put this intuition center stage and suggest a scalar analysis of TSSs. Accordingly, (1) is said to mean (2.6) semantically and to weakly presuppose (2.7).

$$
\begin{align*}
& \forall q^{\prime} \in \operatorname{Alt}(q)\left(q^{\prime}>q \rightarrow q^{\prime} \text { is not necessary for } p\right)  \tag{2.6}\\
& \text { All ways more effortful than } q \text { are not necessary for } p .
\end{align*}
$$

$$
\begin{equation*}
\exists q^{\prime} \in \operatorname{Alt}(q)\left(q^{\prime} \text { is necessary for } p\right) \tag{2.7}
\end{equation*}
$$

There is something which is necessary for $p$.
Effort of a proposition is defined in terms of probability degrees: More effortful ways are less probable. $D(p) \in[0,1]$ is the probability degree of proposition $p$. With this (2.6) can be rephrased as (2.8).

$$
\begin{equation*}
\forall q^{\prime} \in \operatorname{Alt}(q)\left(D\left(q^{\prime}\right)<D(q) \rightarrow q^{\prime} \text { is not necessary for } p\right) \tag{2.8}
\end{equation*}
$$

In order to derive the sufficiency of $q$ for $p$ which is intuitively expressed by (1) and not to succumb to the prejacent problem, probability degrees are themselves considered necessary or sufficient for a proposition $p$. A probability degree $d$ is necessary (sufficient) for $p$ iff there is a proposition $q$ such that $q$ is necessary (sufficient) for $p$ and $D(q)=d$. Necessity and sufficiency then interrelate in various ways via probability degrees, e.g. as in (2.9).

$$
\begin{equation*}
d \text { is sufficient for } p \text { iff } \forall d^{\prime}<d \text { ( } d^{\prime} \text { is not necessary for } p \text { ) } \tag{2.9}
\end{equation*}
$$

There is no proposition less likely / more effortful than degree $d$ necessary for $p$.
According to the authors (2.8) and (2.9) together yield that some proposition $q^{\prime}$ with $D\left(q^{\prime}\right)=D(q)$ is sufficient for $p$. By a strengthening implicature, this $q^{\prime}$ is identified with $q$ and the sufficiency of $q$ for $p$ is ensured.

It needs to be noted on the side that the inference from (2.8) and (2.9) to the existence of some proposition $q^{\prime}$ sufficient for $p$ with $D\left(q^{\prime}\right)=D(q)$ is a non-sequitur, unless, implausibly, $\operatorname{Alt}(q)$ contains all propositions for each degree $<D(q)$. Disregarding the details of the formalization, it appears that the main point of criticism is that the motivation for talking about probability degrees remains utterly mysterious, although the intuitions about scalar readings are, I claim, basically on the right track.

## 3 Teleological Necessity

The main thesis of this paper is that to account for the meaning of TSSs a standard theory of the meaning of only can be pulled off effortlessly, if only the correct notion of teleological necessity, is supplied. The questions to be addressed in this section are: (i) what is teleological necessity, (ii) what information is conveyed by teleological modals and (iii) what reading of the prejacent TNS may we assume for TSSs?

### 3.1 Teleological Necessity $=$ Logical Necessity + Dependency

It is clear that a TNS like (2) does not only express $q$ 's logical necessity for $p$. If "In order for $p$, have to $q$ " was a feasible sentence for all propositions $p, q$ such that $p$ logically entails $q$, then we'd expect all instantiations where $q$ is a result of $p$ to be legitimate instantiations. But this is not the case. (8) should be true and felicitous if teleological necessity was just logical necessity, but, for all we know about kangaroos, it is marked:
?In order for Kanga to lose her tail, she has to topple over.
To see what is at stake for a requirement on $p, q$ pairs for instantiation in "In order for $p$, have to $q$ ", the following coin-flip scenario is illuminating.
Coin Flip Scenario Suppose Hans bet on tails and we are about to flip a fair coin.
(9) a. In order for Hans to win, the coin has to come up tails.
b. ?In order for the coin to come up tails, Hans has to win.

Now suppose that the coin was flipped, and it came up heads.
(10) a If the coin had come up tails, Hans had won.
b ?If Hans had won, the coin would have come up tails.
The parallel between (9) and (10) suggests that the same notion of dependency between events is needed for feasibility of TNSs that informs our intuitive judgements about counterfactuals. Having no intention to model these here, I will just assume the correct dependencies to be given. It is then required for pragmatic felicity of "In order for $p$, have to $q$ " that $p$ depends on $q$. Leaving $p$ 's dependence on $q$ implicitly understood, we can define the notion of teleological necessity $\operatorname{Nec}(p, q)$ simply as $\operatorname{logical}$ necessity: $\operatorname{Nec}(p, q)$ is true in $w$, $w \in \operatorname{Nec}(p, q)$, if all $p$-worlds that are contextually accessible from $w$ are $q$-worlds. There is moreover serious reason for hope that a suitable rendering of dependency will account for temporal matters naturally.

### 3.2 Information Conveyed by Teleological Modals

To say that $\operatorname{Nec}(p, q)$ is true in a world if all contextually accessible $p$-worlds are $q$-worlds, leaves open the question what kind of accessibility relation teleological modals require. According to von Fintel and Iatridou's (2005a) analysis of TNSs in terms of Kratzer's (1991) theory of modality, $\operatorname{Nec}(p, q)$ is true in a world $w$ relative to a circumstantial modal base $f(w)$ if all worlds in $f(w)$ where $p$ is true are $q$-worlds. As circumstantial modality seems to be an appropriate candidate for our running example (2) and most others, I will follow von Fintel and Iatridou here.

In restricting ourselves to circumstantial modality we restrict ourselves to cases where TNSs and TSSs are used for predictions about future courses of events. A teleological modal informs us about how the future will evolve. It reduces epistemic uncertainty about what the state of affairs is at present by telling us that the real world $w$ faces a particular future, i.e. is associated with a modal base $f(w)$. If we separate epistemic and circumstantial modality in this way, we may assume that for a contextually given set of alternatives $Q$ and the set of possible worlds $W$, the modal base $f(w)$ of each $w \in W$ is partitioned by $Q$
into singleton sets, i.e. we assume that for each $w \in W$ there is a bijection $f_{w}: Q \rightarrow f(w)$. The idea is that for a given conceivable possible world $w$ there will be just one way $w$ will develop if $q \in Q$ takes place. For epistemic uncertainty how the future will develop under $q \in Q$, we feature worlds $w, w^{\prime}$ with $f(w) \neq f\left(w^{\prime}\right)$.

It is here that we have to solve a problem that we came across earlier. In section 2.2 we saw that von Fintel and Iatridou's analysis of TSSs appeared too weak, while Huitink's amendment appeared too strong. The root of the problem is that our analysis of teleological modality has to leave room for a remote chance of sheer luck in achieving $p$ without $q$ and a remote chance of bad luck in not achieving $p$ despite $q$. One way of dealing with this problem is to assume a restriction to normal courses of events. Another possibility is to think of $f(w)$ as the state of affairs right after $q \in Q$ has become true. As the slack that we want to incorporate into the model stems from nature's mysterious ways, what is at stake in $w^{\prime} \in f(w)$ is not whether $p$ is true or false, but whether the proposition $p_{\delta}^{*}$ is, a proposition that says that it is within the hands of the addressee to bring about $p$ with a sufficiently high probability $\delta$. Notice that in some cases $\delta$ might just be 1 and there is nothing further that the agent has to do to achieve $p$. The coin flip scenario (9a) is an example of such a situation. In other examples, however, amongst which (2), the slack parameters are needed to account for normal courses of events and normal behaviour of goal-oriented agents. We thus define:

$$
\begin{equation*}
w \in \operatorname{Nec}(p, q) \text { iff } \forall w^{\prime} \in f(w)\left(w^{\prime} \in p_{\delta}^{*} \rightarrow w^{\prime} \in q\right) \tag{3.1}
\end{equation*}
$$

### 3.3 Kinds of Teleological Necessity

In this section I will argue that the meaning of a TNS is context-dependent in interesting ways. $\operatorname{Nec}(p, q)$ might be the basic case, but the presence of scalar only forces a particular reading of the prejacent TNS in TSSs, which is however also available outside of TSSs, if only the circumstances are appropriate. Eventually I will propose that in TSSs the underlying prejacent TNS gets a scalar 'at least'-reading.

Here is an example situation which provides evidence that a TNS may be pragmatically enriched to include further contextually salient goals in addition to the mentioned.
Shanghai Scenario A customer of a travel agency declares his wish to fly to Shanghai. There are three airways available. $A$ and $B$ fly to Shanghai, $C$ heads for Tokyo. Assume that $B$ and $C$ are comfortable to travel with, unlike $A$.
(11) In order to fly comfortably, you have to fly with $B$.

The travel agent may say (11) in this situation without running risk of untruthfulness, because it is understood that the contextually salient goal to fly to Shanghai is implicitly assumed. The case suggests that for contextually salient goals $r$ a TNS may be pragmatically enriched from $\operatorname{Nec}(p, q)$ to $\operatorname{Nec}(p \wedge r, q)$.

This context dependence may also be made responsible for scalar readings of teleological necessity. In a situation where it is mutually known that the addressee wants to minimize his effort in achieving a certain goal, we might assume that the further wish to be economical creeps into the reading of teleological necessity, just as other additional
salient goals do. By way of illustration, suppose that we consider three locations where German bread might possibly be obtained: Leidsestraat (LS), Haarlemmerstraat (US) and Utrechtsestraat (US) with a preference order, based on walking distance, for instance, LS ${ }^{\mathrm{E}} \mathrm{HS}<{ }^{\mathrm{E}}$ US, i.e. LS preferred over HS etc. Now it seems that (2) may be said truly and felicitously even if German bread is available in Utrechtsestraat if only the effort scale is sufficiently salient in the discourse. This clearly speaks for a scalar reading of TNSs.

Of course, the minimization of effort cannot be accounted for simply by adding a further proposition $r$ to yield $\operatorname{Nec}(p \wedge r, q)$, as was the case with example (11). Minimization of effort in realizing $p$ requires comparison with other possible ways of realizing $p$. This can be achieved, for example, in a Kratzerian vein by taking into account an additional ordering source $g(w)$. Let $\operatorname{Nec}_{\mathrm{Sc}}(p, q)$ be the proposition expressed by a TNS under its scalar reading and let $\operatorname{Nec}_{\text {Sc }}(p, q)$ be true in $w$ relative to a circumstantial modal base $f(w)$ and ordering source $g(w)$, which now takes care of the additional wish to minimize effort, if all the $g(w)$-best worlds in $f(w)$ where $p$ is true are $q$-worlds. The $g(w)$-best worlds are minimal worlds according to the ordering $\prec$ defined as usual: $v \prec u$ iff $\{p \in g(w) \mid u \in p\} \subset$ $\{p \in g(w) \mid v \in p\}$.

Let $\left\langle Q, \leq^{\mathrm{E}}\right\rangle$ be a preference order on the set of possible means. Barring clear intuitions about effort-incomparable alternatives, I will assume throughout the paper that all preference orders are linear, but not necessarily strict. I will furthermore assume that $Q$ is finite. $g(w)$ is meant to capture the goal of minimizing effort in achieving $p$. That means that $g(w)$ will contain the proposition $p_{\delta}^{*}$ and each proposition $q^{\uparrow}$ for all $q \in Q$ where $q^{\uparrow}$ is true in a world $w$ if $w \in q_{w}$ and $q_{w} \leq^{\mathrm{E}} q$. This yields:

$$
\begin{equation*}
w \in \operatorname{Nec}_{\mathrm{Sc}}(p, q) \text { iff } \forall u \in f(w)\left(\left(u \in p_{\delta}^{*} \wedge \neg \exists v \in f(w)(v \prec u)\right) \rightarrow u \in q\right) \tag{3.2}
\end{equation*}
$$

It is clear from the way $g(w)$ is defined here that if a world $u$ makes $p_{\delta}^{*}$ true, then any world $v$ for which $v \prec u$ holds has to make $p_{\delta}^{*}$ true as well. Furthermore $v$ has to make strictly more propositions from the set $\left\{q^{\uparrow} \mid q \in Q\right\}$ true which just means that if $v \in q_{v}$ and $u \in q_{u}$, then $q_{v}<{ }^{\mathrm{E}} q_{u}$. Based on the assumption that each $f(w)$ is partitioned by $Q$, this suggests that we can consider an easier alternative ordering $<{ }^{\mathrm{E}}$ between worlds $u, v \in f(w)$ defined as $v<{ }^{\mathrm{E}} u$ iff $v \in q_{v}$ and $u \in q_{u}$ and $q_{v}<{ }^{\mathrm{E}} q_{u}$. With this it is easily seen that (3.2) is equivalent to (3.3).

$$
\begin{equation*}
w \in \operatorname{Nec}_{\mathrm{Sc}}(p, q) \text { iff } \forall u \in f(w)\left(\left(u \in p_{\delta}^{*} \wedge \neg \exists v \in f(w)\left(v \in p_{\delta}^{*} \wedge v<^{\mathrm{E}} u\right)\right) \rightarrow u \in q\right) \tag{3.3}
\end{equation*}
$$

Although not crucial, it pays to assume a scalar 'at least'-reading of TNSs in TSSs, because this way fewer amendments to the theory of only by means of which I want to compute the meaning of TSSs in the next section have to be made. As a suggestive example of what a scalar 'at least'-reading is and as evidence that such readings can be justified for TNSs also outside of TSSs, consider again the German bread scenario with alternatives $\mathrm{LS}<{ }^{\mathrm{E}} \mathrm{HS}<{ }^{\mathrm{E}}$ US. What (12) now seems to be saying is that there is no German bread at Leidsestraat, for sure, and that the addressee has to go to Haarlemmerstraat at least.
(12) In order to get German bread, you have to go to HS, if not even to US.

In order to model the scalar 'at least'-reading, let $q$ ' be a proposition that says that $q$ or more is the case: $w \in q^{\downarrow}$ iff $w \in q_{w}$ and $q_{w}>^{E} q$ or $q_{w}=q$ for some $q_{w} \in Q$. Then let $\operatorname{Nec}^{\prime}{ }_{\mathrm{Sc}}(p, q)$ be the proposition expressed by a TNS under its scalar 'at least'-reading.

$$
\begin{equation*}
w \in \operatorname{Nec}^{\prime}{ }_{S c}(p, q) \text { iff } \forall u \in f(w)\left(\left(u \in p_{\delta}^{*} \wedge \neg \exists v \in f(w)\left(v \in p_{\delta}^{*} \wedge v<^{\mathrm{E}} u\right)\right) \rightarrow u \in q^{\downarrow}\right) \tag{3.4}
\end{equation*}
$$

Given a set $Q$ of alternatives, define $\operatorname{Alt}(q)=Q \backslash\{q\}$ for each $q \in Q$. If we further define what it means for a $q \in Q$ to enable $p$ in a world $w: w \in \operatorname{Enable}(p, q)$ iff there is a $w^{\prime} \in f(w)$ such that $w^{\prime} \in q$ and $w^{\prime} \in p_{\delta}^{*}$, then it can be proved ${ }^{1}$ that (3.4) has a much more intelligible equivalent in (3.5) which I will be using hereafter.

$$
\begin{equation*}
w \in \operatorname{Nec}^{\prime}{ }_{\mathrm{Sc}}(p, q) \text { iff } \forall q^{\prime} \in \operatorname{Alt}(q)\left(q^{\prime} \leq^{\mathrm{E}} q \rightarrow w \notin \operatorname{Enable}\left(p, q^{\prime}\right)\right) \tag{3.5}
\end{equation*}
$$

## 4 Teleological Sufficiency

In the beginning we saw that an analysis of a TSS into semantic component (4.1) and pragmatic component (4.2) was subject to the prejacent problem: a TSS can be true and felicitous even if there are more ways of achieving $p$ than just $q$. Weakening the pragmatic component came at the price of loosing $q$ 's sufficiency for $p$.

$$
\begin{gather*}
\left\{w \in W \mid \forall q^{\prime} \in \operatorname{Alt}(q) w \notin \operatorname{Nec}\left(p, q^{\prime}\right)\right\}  \tag{4.1}\\
\operatorname{Nec}(p, q) \tag{4.2}
\end{gather*}
$$

To overcome this problem, I suggest similarly to Krasikova and Zhechev that only in teleological sufficiency statements is scalar. Instead of excluding all of the alternatives we only exclude more relevant alternatives. What is more relevant in turn is based on an underlying effort or preference ordering of the considered alternatives. The presence of scalar only hands down scalarity to the underlying notion of teleological necessity and justifies an 'at least'-reading. Taken together, I suggest that a TSS comprises (4.3) as its semantic and (4.4) as its pragmatic component.

$$
\begin{equation*}
\left\{w \in W \mid \forall q^{\prime} \in \operatorname{Alt}(q)\left(q<^{\mathrm{E}} q^{\prime} \rightarrow w \notin \operatorname{Nec}_{\mathrm{Sc}_{c}}\left(p, q^{\prime}\right)\right)\right\} \tag{4.3}
\end{equation*}
$$

I will show presently how the analysis in (4.3) and (4.4) can be derived from a recent independent account of the meaning of only. I tend to believe, however, that nothing

[^5]hinges crucially on the precise theory of only that is used to calculate the meaning of a TSS as in (4.3) and (4.4), as long as it can handle scalar readings. It is therefore in order to elaborate informally first how this analysis overcomes the problems that we want it to. In particular, we want to see that the prejacent problem of von Fintel and Iatridou is avoided and that our intuitions about sufficiency are met.

If we apply (4.3) and (4.4) to our main example (1), we would like to see what the predictions are in case there are less, equally and more preferred alternatives to going to Haarlemmerstraat. So assume that $Q=\{$ LS, NDS, HS, US\}, where NDS is short for Nieuwe Doelenstraat and preferences are: $\mathrm{LS}<{ }^{\mathrm{E}} \mathrm{NDS}={ }^{\mathrm{E}} \mathrm{HS}<{ }^{\mathrm{E}}$ US. With this:

From (4.3) we get that $\neg \mathrm{Nec}^{\prime}{ }_{\mathrm{Sc}}$ (GB, US) which means that there is some $q \in \operatorname{Alt}(\mathrm{US})$ with $q \leq{ }^{\mathrm{E}}$ US such that Enable (GB, $q$ ) is true. (4.4) yields that $\operatorname{Nec}^{\prime}{ }_{S c}(\mathrm{~GB}, \mathrm{HS})$, so that all $q \in \operatorname{Alt}(\mathrm{HS})$ with $q \leq{ }^{\mathrm{E}}$ HS do not enable GB. Together, Enable(GB, HS) must be true.
Notice that the prejacent problem does not arise. Given the analysis in (4.3) and (4.4), (1) may be true and felicitous, even if there are alternative successful means of getting German bread, as long as these are not preferred to going to Haarlemmerstraat. Also, our intuitions about sufficiency are met: going to Haarlemmerstraat turned out to be a means of getting German bread. Moreover, it might be said that (4.3) alone vindicates our intuitions about sufficiency. Von Fintel and Iatridou (2005a) ascribe to Beck and Rullmann (1999) the observation that " $q$ being sufficient for $p$ " has a natural paraphrase in "for $p$, it's not necessary to do more than $q$ ". The very same intuition most certainly also motivated Krasikova and Zhechev to relate sufficiency and necessity of probability degrees as in (2.9).

The question remains, whether the meaning intuitions in (4.3) and (4.4) can be accounted for with a standard theory of only in a straight-forward manner. In what follows I will shortly show that van Rooij and Schulz's (2006a) recent background-alternatives approach to the meaning of only does the trick nearly effortlessly. Given a sentence "Only $B(F)$ ", with $B$ a background predicate and $F$ the focus applying to it, the basic idea of this approach is to assimilate the workings of only to exhaustification and to say that the meaning of "Only $B(F)$ " is the meaning of $B(F)$ interpreted in worlds that are minimal with respect to the extension of the background predicate. A world $v$ is more minimal with respect to background $B$ than world $w, v<_{B} w$, if $v$ is exactly like $w$, except that $B[v] \subset B[w]$, i.e. the extension of the background predicate in $v$ is included in the extension of the background predicate in $w$. With this, van Rooij and Schulz compute the overall meaning impact of a sentence "Only $B(F)$ " as (4.5) and identify (4.6) as its semantic component.

$$
\begin{gather*}
\operatorname{ONLY}(F, B)=\left\{w \in W \mid w \in B(F) \wedge \neg \exists v \in W\left(v \in B(F) \wedge v<_{B} w\right)\right\}  \tag{4.5}\\
\left.\operatorname{only}(F, B)=\left\{v \in W \mid \exists w \in \operatorname{ONLY}(F, B) v \leq_{B} w\right)\right\} \tag{4.6}
\end{gather*}
$$

For the calculation of our example (1) once again assume that $Q=\{\mathrm{LS}, \mathrm{NDS}, \mathrm{HS}, \mathrm{US}\}$ and that LS $<{ }^{\mathrm{E}} \mathrm{NDS}={ }^{\mathrm{E}} \mathrm{HS}<{ }^{\mathrm{E}}$ US. We are interested in the case where the background predicate $B$ is $\left.\mathrm{Nec}^{\prime} \mathrm{Sc}^{(\mathrm{GB}, \cdot}\right)$ and where the extension of the background predicate is $B[w]=$ $\{q \in Q \mid w \in B(q)\}$. Clearly, the relevant possibilities to be considered are the sixteen possible distributions of truth-values of Enable (GB, $q$ ) for $q \in Q$ :

|  | $w_{1}$ | $w_{2}$ | $w_{3}$ | $w_{4}$ | $w_{5}$ | $w_{6}$ | $w_{7}$ | $w_{8}$ | $w_{9}$ | $w_{10}$ | $w_{11}$ | $w_{12}$ | $w_{13}$ | $w_{14}$ | $w_{15}$ | $w_{16}$ |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| LS | $\sqrt{ }$ | $\sqrt{ }$ | $\sqrt{ }$ | $\sqrt{ }$ | $\sqrt{ }$ | $\sqrt{ }$ | $\sqrt{ }$ | $\sqrt{ }$ | - | - | - | - | - | - | - | - |
| NDS | $\sqrt{ }$ | $\sqrt{ }$ | $\sqrt{ }$ | $\sqrt{ }$ | - | - | - | - | $\sqrt{ }$ | $\sqrt{ }$ | $\sqrt{ }$ | $\sqrt{ }$ | - | - | - | - |
| HS | $\sqrt{ }$ | $\sqrt{ }$ | - | - | $\sqrt{ }$ | $\sqrt{ }$ | - | - | $\sqrt{ }$ | $\sqrt{ }$ | - | - | $\sqrt{ }$ | $\sqrt{ }$ | - | - |
| US | $\sqrt{ }$ | - | $\sqrt{ }$ | - | $\sqrt{ }$ | - | $\sqrt{ }$ | - | $\sqrt{ }$ | - | $\sqrt{ }$ | - | $\sqrt{ }$ | - | $\sqrt{ }$ | - |

Unfortunately, if we rely on the pure extensional ordering $<_{B}$ defined above we do not make the right predictions. In particular, with $<_{B}$ we get as the semantic meaning of (1) that only $(\mathrm{HS}, B)=\left\{w_{1}, \ldots, w_{10}, w_{13}, w_{14}\right\}$. But for (4.3) to be vindicated, we want $w_{11}$ and $w_{12}$ to be in only (HS, $B$ ) as well. The reason why $w_{11}$ and $w_{12}$ are not in only $(\mathrm{HS}, B)$ under the ordering $<_{B}$ is that worlds $w_{11}$ and $w_{12}$ are not comparable with worlds $w_{13}$ and $w_{14}$, in turn because the propositions $\operatorname{Nec}^{\prime}{ }_{\mathrm{Sc}}(\mathrm{GB}, \mathrm{NDS})$ and $\mathrm{Nec}^{\prime}{ }_{\mathrm{Sc}}(\mathrm{GB}, \mathrm{HS})$ do not entail one another.

Fortunately, the problem already has an established solution. To account for contextsensitivity of exhaustification, as needed for scalar reasoning, domain restriction and answers to mention-some questions, van Rooij and Schulz (2006b) suggest to consider not a pure entailment-based, but a relevance-based ordering on worlds $<_{B}^{\mathrm{r}}$. The very same idea, of course, then applies to their theory of only. In our present example, a nonstrict linear order on $Q$ gave us, so conceived, a mention-some case in the middle of a scale: options NDS and HS are equally preferred and therefore $\mathrm{Nec}^{\prime}{ }_{S c}(\mathrm{~GB}, \mathrm{NDS})$ and $\mathrm{Nec}^{\prime}{ }_{\mathrm{Sc}}(\mathrm{GB}, \mathrm{HS})$ should be equally good propositions for any natural measure of relevance. Consider for instance the addressee's decision problem where to go to get German bread with possible actions $Q$. Prior to inquiry assume that all possibilities are equiprobable. Van Rooij (2004) suggests to measure the relevance of a proposition $P$ as the change in utility value, i.e. the expected utility of the best action, that learning $P$ brings about. With this the relevance order $<^{\mathrm{r}}$ on propositions is straight-forward: ${ }^{2}$ $\operatorname{Nec}^{\prime}{ }_{S c}(\mathrm{~GB}, \mathrm{LS})<{ }^{\mathrm{r}} \mathrm{Nec}^{\prime}{ }_{\mathrm{Sc}}(\mathrm{GB}, \mathrm{NDS})={ }^{\mathrm{r}} \operatorname{Nec}^{\prime}{ }_{\mathrm{Sc}}(\mathrm{GB}, \mathrm{HS})<{ }^{\mathrm{r}} \mathrm{Nec}^{\prime}{ }_{S c}(\mathrm{~GB}, \mathrm{US})$. Based on $<^{\mathrm{r}}$, define a relevance order on worlds $<_{B}^{\mathrm{r}}$ as usual:

$$
\begin{equation*}
v<_{B}^{\mathrm{r}} w \text { iff }\{u \in W \mid B[v] \subseteq B(u)\}<^{r}\{u \in W \mid B[w] \subseteq B(u)\} \tag{4.7}
\end{equation*}
$$

This yields: $w_{1}={ }_{B}^{\mathrm{r}} \ldots={ }_{B}^{\mathrm{r}} w_{10}<_{B}^{\mathrm{r}} w_{11}={ }_{B}^{\mathrm{r}} \ldots={ }_{B}^{\mathrm{r}} w_{14}<{ }_{B}^{\mathrm{r}} w_{15}={ }_{B}^{\mathrm{r}} w_{16}$ and if we now use $<_{B}^{\mathrm{r}}$ instead of ${<_{B}}_{B}$ in the calculation of (4.5) and (4.6) we get: $\operatorname{ONLY}(\mathrm{HS}, B)=\left\{w_{13}, w_{14}\right\}$ and $\operatorname{only}(\mathrm{HS}, B)=\left\{w_{1}, \ldots, w_{14}\right\}$. This is the correct prediction. Proposition $\left\{w_{1}, \ldots, w_{14}\right\}$ is identified as the semantic meaning of (1) and this corresponds to (4.3). The overall meaning of (1) is predicted to be $\left\{w_{13}, w_{14}\right\}=\left\{w_{1}, \ldots, w_{14}\right\} \cap \operatorname{Nec}^{\prime}{ }_{\mathrm{Sc}}(\mathrm{GB}, \mathrm{HS})$ as desired.

## 5 Conclusion

The main aim of this paper has been to substantiate the idea that a standard account of the meaning contribution of only to TNSs is all that it takes to explain our intuitions

[^6]about TSSs. The present proposal indeed did not suffer from the prejacent problem, which previous analyses were chiefly concerned with, and moreover managed to provide an adequate analysis of sufficiency as expressed in TSSs.

## Acknowledgements

I would like to thank Floris Roelofson, Samson de Jager, Fabrice Nauze, Christian Plunze, Magdalena Schwager, Henk Zeevat, Kai von Fintel and Robert van Rooij for help, comments and discussion.

## Bibliography

Beck, S. and H. Rullmann (1999). A flexible approach to exhaustivity in questions. Natural Language Semantics 7(3), 249-298.
von Fintel, K. and S. Iatridou (2005a). Anatomy of a modal. ms, MIT.
von Fintel, K. and S. Iatridou (2005b). What to do if you want to go to harlem: Anankastic conditionals and related matters. ms, MIT.
Horn, L. R. (1969). A presuppositional approach to Only and Even. In Chicago Linguistics Society, 5, pp. 98-107.

Horn, L. R. (1996). Exclusive company: Only and the dynamics of vertical inference. Journal of Semantics 13(1), 1-40.
Huitink, J. (2005). Analyzing anakastic conditionals and sufficiency modals. In Proceedings of Console XIII.
Krasikova, S. and V. Zhechev (2005). Scalar use of Only in conditionals. In P. Dekker and M. Franke (Eds.), Proceedings of the 15th Amsterdam Colloquium, pp. 137-142.
Kratzer, A. (1991). Modality. In A. von Stechow and D. Wunderlich (Eds.), Semantik: Ein internationales Handbuch der zeitgenössischen Forschung, pp. 639-650. Berlin: Walter de Gruyter.
van Rooij, R. (2004). Utility, informativity and protocols. Journal of Philosophical Logic 33(4), 389-419.
van Rooij, R. and K. Schulz (2006a). Only: Meaning and implicatures. In M. Aloni, A. Butler, and P. Dekker (Eds.), Questions in Dynamic Semantics, pp. 193-223. Amsterdam, Singapore: Elsevier.
van Rooij, R. and K. Schulz (2006b). Pragmatic meaning and non-monotonic reasoning: The case of exhaustive interpretation. To appear.

# Subject-Marking in Hindi/Urdu: A Study in Case and Agency 

Scott Grimm<br>Universitat Konstanz<br>Scott.Grimm@uni-konstanz.de


#### Abstract

Semantic parameters of agency and affectedness have long been known to affect the realization of case-marking. This paper proposes an approach which decomposes agency and affectedness into semantic properties, loosely based on Dowty's proto-role theory, but conceived in terms of privative opposition and organized into a lattice. This results in a structured framework which is capable of modelling both case systems and case alternations, as is demonstrated by an account of subject-marking in Hindi/Urdu.


## 1 Introduction

Modulation of semantic parameters such as agency and affectedness are known to affect the realization of case-marking (Hopper and Thompson 1980). For instance, subjects low in agency and/or affected are cross-linguistically at risk to be marked by an alternate case from that of canonical subjects, should the language dispose of a sufficiently rich case system. Yet, explicitly connecting individual parameters with the semantics of case alternations has largely proven elusive. One difficulty is that a single case, e.g., the dative in Hindi/Urdu, can serve to mark a variety of semantic distinctions, including both experiencer and recipient arguments. The semantic content of a case then must both account for its diverse uses and display the interrelation among them. A second complication is that realizations of case, and a fortiori, case alternations, often cannot be attributed to one sole parameter, but arise only in the context of the interaction of several, necessitating an account of this interaction in precise terms. In what follows, a method is proposed to define case in terms of the parameters of agency and affectedness - reanalyzed as emergent properties dependent on more primitive, constituent properties and hierarchized in a lattice structure. This provides a structured, explanatory device for how a language marshals morphological resources to indicate subtleties of argument realization, as demonstrated in an application to the subject-marking patterns in Hindi/Urdu.

## 2 Decomposing Agency and Affectedness

Whatever view one holds on the central function of case-marking, whether it serves to index an argument with a semantic property or differentiate between arguments (cf. (Song 2001) for the debate), it is clear that languages which develop case systems use them at least to mark subjects and objects. A large body of research conducted on argument structure has demonstrated that subject and object selection is largely determined by the thematic content associated with the arguments of a given predicate, e.g., which participants are agents and which are patients. Since case is responsible for marking such arguments, there is clearly a relation between thematic content and the eventual marking patterns in case languages.

Investigations of argument structure have made use in one way or another of the notions of agency and affectedness as those underlying what determines which participant is an agent or a patient, and ultimately, subject or object, of transitive constructions; however, taking the concepts of 'agent' and 'patient' in themselves as primitives has proven to be too unwieldy to account for more fine-grained syntactic and morphological behavior. The work of (Dowty 1991) provided a theoretical advance by decomposing the larger notions of 'agent' and 'patient' into constituent properties, permitting 'agent' and 'patient' to become emergent properties, and amplify the level of detail in argument structure analyses.

On Dowty's account, thematic roles emerge from a set of "Proto-Properties", eventbased properties entailed by the verb, relativized to Proto-Agents and Proto-Patients. Proto-Agent properties include "causally affecting another entity", "motion (relevant to another participant)" while representative Proto-Patient properties are "causally affected by another entity" and "stationary (relevant to another participant)". While the ProtoRole theory indeed provides a more suitable account of what it is to be an 'agent' or 'patient', the choice and organization of the primitives limits its application. As can be seen by the above Proto-Properties, a two-participant transitive situation is taken as given. This assumption leads to difficulties in treating constructions which deviate from the transitive paradigm, such as middles and intransitives. Further, the properties of (Dowty 1991) include the complex notions of 'affectedness' and 'causation' taken as primitive. Affectedness, while used in a variety of ways in the literature, has generally been conceded to be not a binary concept, but a three-way distinction between unaffected, partially or totally affected. ${ }^{1}$ Causation implies at least two participants and some sort of direct link between them, and taking such a property as primitive reinforces the bias towards transitive situations. ${ }^{2}$ An increase in simplicity and empirical reach can be gained by reformulating the properties without reference to other participants and complex notions.

The approach here retains the use of event-based properties entailed by the verb, as in (Dowty 1991), to capture the parameters of agency and affectedness; yet, rather than using two distinct sets of properties for agents and patients, I use one set of properties

[^7]which gives rise to a privative opposition between agents and non-agents.
I assume a set of properties which refer to modes of participation in events: instigation, motion, sentience, volition, and different degrees of persistence. Instigation entails any argument effecting the event designated by the predicate. Motion is entailed just in case the argument is required to be in motion. Sentience designates conscious involvement in the event (Rozwadowska 1988) while volition designates deliberate engagement in the event. Agents, then, will typically possess one or more of these properties.

Persistence is a two-tiered notion, for something can persist existentially, that is, its essence remains the same throughout the event/state, or it can persist qualitatively-i.e., it persists in all its particulars. Either of these can obtain at the beginning and/or the end of the event - in terms of features, we have the following set: existential persistence (beginning), existential persistence (end), qualitative persistence (beginning), and qualitative persistence (end).

| Agentive | Non-Agent ('Patient') |
| :---: | :---: |
| volitional | -volition |
| sentience | - sentience |
| instigation | - instigation |
| motion | - motion |
| existential persistence(beginning) | - existential persistence(beginning) |
| existential persistence(end) | - existential persistence(end) |
| qualitative persistence(beginning) | - qualitative persistence(beginning) |
| qualitative persistence(end) | - qualitative persistence(end) |

Table 1.1: : Agency Properties
As shown in Table 1.1, the set of properties above establishes a privative opposition between agents and non-agents (of whom patients are special subset), rather than equipollent opposition between agents and patients. This yields a continuum of agency, from maximal agents to non-existent entities. This move is motivated in as much as agents can stand in opposition to arguments which do not strictly qualify as patients, e.g., objects of statements of negative existence, incorporated/cognate objects ("sing a song") or narrowscope objects of verbs such as "seek"-i.e., arguments which do not entail any existence independent of the event at hand. In contrast, patients are typically affected by the event, which presupposes existence prior to the event, thus they minimally entail existential persistence (beginning). Therefore, the opposition between agents and patients falls out from this feature system in that agents will possess total persistence along with other agency properties while patients will generally possess no properties save initial persistence and possibly existential persistence (end).

Parallel to the manner that the gradations of agency can be accounted for by different combinations of the participant properties above, affectedness is defined over a range of combinations holding in common a lack of persistence. Affectedness, in its most basic
semantic sense, designates that an affected object is altered by the event in some manner, i.e., "changed or moved" (Anderson 1979). ${ }^{3}$ Without any loss of descriptive power, the concept of affectedness can be inverted and recast in terms of persistence. Further, this feature configuration is able to capture the different degrees of affectedness with respect to existence. Totally affected patients, e.g., of verbs of destruction/consumption ('destroy', 'eat'), entail that their object argument persists existentially at the beginning of the event, but not at the end. Patients which are partially affected (e.g., objects of verbs such as 'damage' or 'move') persist existentially throughout the event, but do not persist qualitatively, i.e., they are changed in some manner. Unaffected entities, most often agents, persist both existentially and qualitatively throughout the event.

While the above choice of properties has attempted to avoid taking complex notions as primitive, such as cause or control, the feature set remains conservative with respect to the advances made by the Proto-Role theory. If a descriptive need of such complex properties arises, they can be defined in terms of the stated primitives. Causation can be defined over pairs of arguments where the causer entails instigation and the causee is restricted from qualitative persistence (end). The notion of control has also been found useful as a descriptive label for distinctions made in case alternations to be treated below. External control, where one argument controls another, can be defined over entailments for two arguments, (ArgX: [+ instigation,+ sentient, + volition], ArgY: [- instigation, - volition $]$ ). Internal control, normally used with intransitives where the argument has control over the event, can be defined for single arguments as clusters containing [+ instigation, + sentient, + volition]. These definitions make apparent that cause and external control are relations between participants, in contrast to the other properties which are defined only with respect to the event.

### 2.1 Constructing the Agency Lattice

Merely by positing these primitives in the manner above, a combinatorial argument ensues. Since verbs may entail various combinations of properties, a natural question is which combinations are possible given the set of properties. Eight binary properties lead to a total of 256 possible combinations. In the following, I will show how these possible combinations can be at once constrained from the possible combinations to the logically and conceptually valid combinations and structured via a lattice, a move inspired by (Aissen 2003).

[^8]Logical entailments among the features constrain the combinations possible. For instance, volition entails sentience, since only sentient beings are capable of volition, and - existential persistence (end) entails - qualitative persistence (end), since if an entity does not exist at the end of the event, clearly none of its qualities do either. The possible combinations are further constrained by the conceptual impossibility of arguments designating entities which do not possess at least the feature existential persistence (beginning) combining with agency properties such as motion or sentience.

The remaining combinations can then be given greater structure. One can regard the participant properties as atoms from which "proto-roles" are composed. These atoms and their combinations can be ordered in terms of inclusion-i.e., both motion and instigation are included in the composite term motion^instigation. This set of atomic elements, ordered by inclusion (i.e., a partial order), induces a mathematical structure, a lattice, shown in Figure 1.1 and referred to henceforth as the agency lattice. ${ }^{4}$

The lattice makes the privative opposition holding among the properties visible: the highest node possesses all the properties (the maximal agent) and the lowest node possesses none, not even independent existence. Further, agents are upwards closed in the lattice while patients are downward closed. That is to say, if some node $x$ of the agency lattice is an agent relative to a given predicate, then all the nodes higher than $x$ are as well, and conversely, if some node $y$ of the agency lattice is considered a patient relative to a predicate, then all the nodes lower than $y$ are as well. ${ }^{5}$ This property of the agency lattice guarantees that if the agent (patient) argument of a predicate is satisfied when instantiated by an entity of a given level of agency, it will also be satisfied when instantiated by an entity possessing a higher (lower) level of agency. ${ }^{6}$

This lattice then provides a structure upon which argument structures can be mapped. The focus now turns to case-marking, demonstrating how the lattice structure can represent different cases as continuous regions of the lattice, and in the process, bring forth the commonality between canonical and non-canonical uses of a given case.

### 2.2 Connecting Case and Agency

In explaining the behavior of a given case, one is confronted with both syntactic uses, i.e., marking the arguments of a predicate, and often also semantic uses, e.g., case alternations. Case is often seen as primarily syntactic, therefore the question arises concerning the origin of the semantic properties which underlie case alternations. Further, what is the connection between a syntactic and a semantic use?

The above lattice provides a way to capture the semantic space of argument structure via agency properties. A case, in its syntactic function will refer to a region of this space,

[^9]

Figure 1.1: The Agency Lattice
since it marks a delimited class of arguments. By associating case with the region of its primary use, and hence the semantic properties therein, a general answer is provided to the above questions: the semantic properties of its primary, syntactic use provide the semantic content for extended uses. In more concrete terms, if a case marks a class of arguments, say indirect objects, then the case marker is associated with the semantic properties of that class, here, recipients and beneficiaries. But then, a case marker, equipped with these semantic properties, can be used to express notions and relations appropriate to these properties beyond its primary syntactic function.

A second way in which a case-marker is connected to semantic content is due to being historically conditioned. Case-markers generally originate in other lexical material, having been recruited to express the requisite syntactic function, and thereby have undergone a
grammaticalization process (e.g., the verb 'give' can be recruited as a marker of beneficiaries, cf. (Lord 1989)). Given that case-markers originate from other lexical material, case-markers come into being associated with one or more nodes of the lattice appropriate to the original lexical material. Here, too, the agency lattice provides predictions of constraints on the grammaticalization process of markers of verbal arguments. The process of grammaticalization-weakening of the original sense, generalization to grammatical function, and picking up other senses - can be seen as spreading to other nodes. The lattice, however, predicts that this will only occur with connected nodes, restricting the types of grammaticalization patterns that should be observed.

These two manners in which cases are connected to semantic content are, in fact, intrinsically linked. Whatever lexical item is recruited to become a case-marker is presumably recruited because its sense coheres with that of the syntactic function that is in need of representation, therefore, it is expected that the semantic content of the lexical item with respect to participant properties should fall within the relevant region of the syntactic function. In the other direction, the particular properties inherent to the recruited lexical item will constrain the semantic space which is actually instantiated by the case-marker, determining its possible grammaticalization trajectories.

The methodology for modelling usages of case via the lattice follows directly from the above considerations. First, we map a case to the region corresponding to its primary use and/or that of the lexical material from which it was recruited. It is then incumbent on the semantic properties of that region to provide an explanation for extended uses of the case, as non-canonical subject markers (e.g., experiencers) and in case alternations. Should the case exemplify highly grammaticalized uses, such as becoming an all-purpose subjectmarker, this should be consistent with spreading among connected nodes. It will be shown that the results of this methodology coincide with the descriptive accounts familiar from the literature. Thus, an explanatory account of semantic uses of a case can be derived from its syntactic use without further stipulation. I now turn to applying this methodology to the four cases in Hindi/Urdu relevant for subject-marking.

## 3 The Case-Marking of Subjects in Hindi/Urdu

In this section, I will map the Hindi/Urdu case system to regions of the lattice. The mappings will be established by examining the primary uses of the cases. These mappings will then be shown to correlate with the marked values that uses of the cases assume in opposition to the unmarked nominative.

### 3.1 The Dative

The core function of the dative is to mark the indirect object, which is canonically a recipient/beneficiary. With respect to the agency properties above, clearly a recipient/beneficiary is 'consciously involved' in and is affected qualitatively by the event. As such, the dative will be ascribed the sole property of sentience and be located on the Qualitative Persistence
(Beginning) branch of the lattice, as shown in Figure 1.2.
The dative has an extended use, marking subjects of certain experiential and psychological predicates: physical sensations/conditions, psychological/mental states, wanting/needing and obligation or compulsion. Such predicates clearly require the subject to be sentient, and further, indicate that they are affected in some manner, correlating with the semantic properties ascribed to the dative's primary use.

Further, (Masica 1991) observes that these verbs with dative subjects share the trait that their subjects are non-volitional, in opposition to nominative subjects, which are unmarked for volitionality. This is exemplified by the pair in (1) (from (Mohanan 1994)), where while the nominative subject of (1a) permits both volitional and non-volitional readings, the dative subject of (1b) can only be taken as non-volitional.
a. tusaar
$\mathrm{k}^{h}$ uš
huaa
Tushar.NOM happy.NOM become.PERF
Tushar became happy.
b. tuṣaar=ko $\mathrm{k}^{h}$ ušii huii

Tushar.DAT happiness.NOM happen.PERF
Tushar became happy. (Lit. To Tushar happiness happened.)
This coincides with the region assigned above to the dative, which does not extend to nodes with the property of volition, since this is not relevant for beneficiaries/recipients.


Figure 1.2: Subject Marking in Hindi/Urdu

### 3.2 The Instrumental

The core use of the instrumental is to mark instruments involved in an event. In mapping the instrumental case, note that prototypical instruments are not sentient, although capable of motion and instigation (at least co-instigation along with an understood agent), and are viewed as persisting throughout the event-i.e., if an axe is used to cut, the axe persists throughout the cutting event. Therefore, prototypical instrumentals are located on the Total Persistence branch of the lattice, yet restricted from the nodes containing sentience, as shown in Figure 1.2.

In its use as a subject marker, the instrumental principally marks a demoted (or passive) agent. The location of the instrumental accords with the general function of the passive agent, as a source of instigation of the event, while properties such as volition, or even sentience, are generally not at issue for passive agents.

### 3.3 The Nominative

In Hindi/Urdu, the nominative is not morphologically marked and used for both subjects and objects. In contrast to the other cases, the nominative can mark any level of agency, i.e., the nominative is unmarked for agency; thus, the nominative is not associated with any particular region of the lattice.

### 3.4 The Ergative

The examination of the ergative must begin with its use as a subject marker. In Hindi/Urdu, the subject is obligatorily marked ergative in perfective transitive sentences, whereas intransitive verbs generally require the nominative. Yet, there are a small number of transitive verbs which allow both, such as 'jānnā', designating 'to know' with a nominative subject and 'to find out' with an ergative subject (see discussion and further examples (Mohanan 1994)). Note that the latter is an event over which the subject has internal control-which in terms of the semantic properties assumed here reduces to volition. The canonical region for the ergative thereby is mapped on the lattice to the region containing the feature volition, and constrained to the Total Persistence branch, since the ergative only marks agents, which are prototypically unaffected.

The ergative also enters into an alternation with the nominative in intransitive verbs, as in (2) (see (Butt and King 2005)).
(2) $\quad \operatorname{ram}(=n e)$
$\mathrm{k}^{h}$ as-a
Ram.M.Sg.NOM(ERG) cough-Perf.M.Sg
Ram coughed (purposefully).
(Butt and King 2005) states that the relevant criterion here too is internal control (i.e., volition), for which the ergative is marked and the nominative unmarked, which is precisely what follows from the above mapping.

The ergative does, however, occur in instances where volitionality appears to be a nonissue, as an anonymous reviewer pointed out. Natural forces constitute the main class of exceptions, in phrases such as "The storm broke the glass" (see (Mohanan 1994)). Given that the ergative has developed into the case of the subject in perfective clauses, such an extension of meaning beyond the canonical region is expected, which is in turn consonant with the representation of the grammaticalization process discussed in section 2.2.

## 4 Equipollent Case Alternations

The above has examined instances of marked cases alternating with the unmarked nominative. Hindi/Urdu also disposes of alternations between two marked cases, i.e., equipollent alternations. These are a particularly challenging use of case for which to account, since although the data below are minimal pairs, differing only in case-endings, they display a complex interaction of properties. Yet, we will explain these alternations directly from the cases' position on the lattice, without further stipulation.

### 4.1 Ergative/Dative Alternation

The ergative-dative alternation occurs in this construction found in the Lahori and Delhi dialects (Butt and King 2005):
a. nadya=ne zu ja-na he Nadya.F.Sg.ERG zoo.M.Sg.OBL go-Inf.M.Sg be.Pres.3.Sg. Nadya wants to go to the zoo.
b. nadya=ko ju ja-na he Nadya.F.Sg.DAT zoo.M.Sg.OBL go-Inf.M.Sg be.Pres.3.Sg. Nadya has to/wants ${ }^{7}$ go to the zoo.

By associating cases with complexes of features, as above, the base requirements for such modal uses of case are secured: the ergative is associated with volitionality, and its minimal interpretation is the lowest node of its region, containing only volition and sentience, the semantic prerequisites for volitive modals, while the dative is restricted from volition nodes, is marked as sentient and is qualitatively affected with respect to the event, all of which are semantic prerequisites for deontic modals.

[^10]
### 4.2 Instrumental/Dative Alternation

In Hindi/Urdu, the causee of causative constructions is normally marked by the instrumental; however, ingestives ('eat', 'drink'), verbs of motion, perception ('see', 'hear'), but also 'write', require the dative to mark the causee, and certain verbs alternate between the two, as in (4) (from (Butt 1998)).

> a. anjvm-ne saddaf=ko masala $\mathrm{cak}^{h}$-va-ya
> Anjum.F.Erg Saddaf.F.Acc spice.M.Nom taste-Caus-Perf.M.Sg
> Anjum had Saddaf taste the seasoning.
> b. anjvm-ne saddaf=se masala $\mathrm{cak}^{h}$-va-ya
> Anjum.F.Erg Saddaf.F.Inst spice.M.Nom taste-Caus-Perf.M.Sg
> Anjum had the seasoning tasted by Saddaf.
(4a) entails that the causee is affected by the event and to some degree "consciously involved", while the causee in (4b) is unaffected and is only indirectly involved (Butt 1998). Both distinctions fall out from the position of the cases on the lattice. The instrumental case, located on the Total Persistence branch, is viewed as unaffected, while the dative, on the Qualitative Persistence (Beginning) branch, as affected. Second, the dative is associated with sentience, indicating conscious involvement, whereas the instrumental is restricted from this property. Therefore, the semantic content of the instrumental and dative cases' regions on the lattice delivers an account of their participation in this alternation.

## 5 Conclusion

A re-working of the approach of (Dowty 1991) into one set of features, in terms of privative opposition, and hierarchized in a lattice has led to a structured framework which can account for the fundamental syntactic and semantic distribution of case. The efficacy of the system has been shown to allow for well-grounded and effective explanations of a set of data that has proved to be recalcitrant for linguistics analysis, e.g., case alternations in Hindi/Urdu.

## Acknowledgements

I would like to thank Miriam Butt and Henk Zeevat for comments and discussion, as well as the ESSLLI Student Session reviewers.

## Bibliography

Judith Aissen. Differential object marking: Iconicity vs. economy. Natural Language and Linguistic Theory, 21:435-483, 2003.
Mona Anderson. Noun Phrase Structure. PhD thesis, University of Conneticut, 1979.
Miriam Butt. Constraining argument merger through aspect. In Erhard Hinrichs, Andreas Kathol, and Tsuneko Nakazawa, editors, Syntax and Semantics: Complex Predicates in Nonderivational Syntax, volume 30. Academic Press, 1998.
Miriam Butt and Tracy Holloway King. The status of case. In Veneeta Dayal and Anoop Mahajan, editors, Clause Structure in South Asian Languages. Springer Verlag, 2005.

David Dowty. Thematic proto-roles and argument selection. Language, 67:547-619, 1991.

Scott Grimm. The lattice of case and agentivity. Master's thesis, Universiteit van Amsterdam, 2005. ILLC Master of Logic Series MoL 2005-9.
Paul J. Hopper and Sandra A. Thompson. Transitivity in grammar and discourse. Language, 56:251-299, 1980.
Manfred Krifka. Thematic relations as links between nominal reference and temporal constitution. In Ivan Sag and Anna Szabolcsi, editors, Lexical Matters. CSLI, 1992.
Beth Levin and Malka Rappaport Hovav. Argument Realization. Cambridge University Press, 2005.
Carol Lord. Syntactic Reanalysis in the Historical Development of Serial Verb Constructions in Languages of West Africa. PhD thesis, UCLA, 1989.
Colin Masica. The Indo-Aryan Languages. Cambridge University Press, 1991.
Tara Mohanan. Argument Structure in Hindi. CSLI, 1994.
Åscild Naess. What markedness marks: the markedness problem with direct objects. Lingua, 114:1186-1212, 2004.
John T. Platts. A dictionary of Urdu, classical Hindi, and English. W. H. Allen and Co., 1884.
Boz̀ena Rozwadowska. Thematic restrictions on derived nominals. In Wendy Wlikins, editor, Syntax and Semantics, volume 21, pages 147-165. Academic Press, 1988.
Jae Jung Song. Linguistic Typology: Morphology and Syntax. Longman, 2001.

# Disrupted Agreement Checking in Sentence Comprehension 

Jana Häussler<br>University of Konstanz<br>jana.haeussler@uni-konstanz.de


#### Abstract

The process of computing syntactic representations during language comprehension includes the checking of dependencies like subject-verb agreement. Prior studies have shown that so-called attraction errors arise when a complex subject phrase contains an NP mismatching the head noun in number as in the key to the cabinets. Occasionally people produce agreement with the modifying NP instead of agreement with the subject, and respectively detect a (seeming) agreement violation in comprehension. Attraction errors only occur with singular subjects containing a plural modifier. According to a prominent account, attraction errors are due to percolation of the plural feature of the modifier NP, while the checking process itself is assumed to work basically flawlessly. I will discuss experimental evidence that at least in certain configurations the checking mechanism is to blame for the attraction error. (i) Contrary to the asymmetry mentioned above, in relative-clause constructions attraction errors are not restricted to singular subjects. (ii) Objects can cause attraction error as well, with no asymmetry between singular and plural subjects. (iii) Attraction has no effect for pronoun resolution. In all these cases percolation is rather unlikely. I will suggest an interference account instead: In order to check subject-verb agreement the parser has to retrieve the subject when encountering the verb. Sometimes the interfering NP is retrieved instead of the subject. If this interfering NP does not match the subject in number, an attraction error results. In sum, I will argue that we have to distinguish two types of attraction errors, with different properties and different sources: percolation and interference.


## 1 Introduction

During language comprehension, the parser has not only to build and extend phrase structure representations but also to check various dependencies between lexical items. Subjectverb agreement is one of these dependencies. Despite of the simplicity of the agreement rule - singular subjects require singular verbs and plural subjects require plural verbs, agreement errors occur. So-called attraction errors occur in configurations where the head noun of the subject phrase is separated from the verb by an intervening noun mismatching the head noun in number, as illustrated in (1) (taken from Bock \& Miller, 1991: 46).
(1) *The readiness of our conventional forces are at an all-time low.

In (1) the verb erroneously agrees with the embedded NP conventional forces. In a sense, the head noun readiness has attracted the number feature of the embedded noun, such that a plural marking on the finite verb results. We will call the head noun of a subject NP (readiness) the Agreement controller and the NP which is assumed to be
responsible for the agreement error (forces) the Distractor. The term attraction is used as purely descriptive term for this specific kind of error, with no further implications with respect to the underlying processes.

Attraction errors were first observed in language production (Bock \& Miller, 1991; cf. Bock et al., 2001 for review), but have been attested for language comprehension as well. Attraction-related difficulties in comprehension are reflected by increased reading times on the verb (e.g. Branigan et al., 1995; Nicol et al., 1997; Pearlmutter et al., 1999, Pearlmutter, 2000), and to a substantial number of judgment errors in a speededgrammaticality judgment procedure (the method of the experiments to be described below).

We can distinguish two potential sources of attraction errors in sentence comprehension: (i) the representation of the subject might be error-prone by the presence of the intervening distractor, (ii) the checking process itself might be disrupted due to the interference of the distractor. I will discuss both options below.

## 2 The Computation of Subject-Verb Agreement during Sentence Comprehension

### 2.1 Subject Integration

To integrate the subject into the current partial phrase marker (CPPM) the parser has two tasks: building up a phrase structure representation, and determining the features of the whole subject NP from the features of the words which are dominated by the subject NP. If something goes wrong during the second subtask, the subject might end up with a wrong number specification. If then the verb is checked against this wrong representation, agreement seems to be violated.

A prominent implementation of this idea is the percolation account proposed by Nicol et al. (1997). According to this account, attraction errors result from some kind of erroneous feature migration during the computation of the subject noun phrase. For subject noun phrases containing a distractor mismatching the controller in number, like the key to the cabinets, the plural feature on the distractor erroneously percolates to the NP headed by the controller. As a result, the singular subject NP turns into a plural NP on processing the plural distractor. Thus an agreement error will be detected when the singular verb is integrated and checked against the subject noun phrase.

A percolation account along the lines of Nicol et al. (1997) easily explains two major findings with regard to attraction errors, findings which hold for both production and comprehension. First, it explains the observed asymmetry between singular and plural. Attraction errors only occur with singular subjects containing a plural distractor (the key to the cabinets), but not with the reverse pattern (the keys to the cabinet). For singular distractors, the rate of agreement errors is not higher than the baseline rate that is observed when controller and distractor match in number (the keys to the cabinets). Under the assumption that only plural is specified by a feature while singular is unspecified, as proposed by Eberhard (1997), percolation cannot occur with singular distractors, since
there is nothing to percolate. Plural distractors, in contrast, have a plural feature in their representation and therefore percolation has a chance to apply.

Secondly, percolation accounts can explain why the hierarchical distance between controller and distractor is a main determinant for the occurrence of attraction errors whereas linear distance between distractor and verb is not. In a language production experiment (discussed in Nicol et al., 1997), Vigliocco and Nicol found more attractions errors for (2a) than for (2b).
(2) a. The telegram [to the friends [of the soldier]]
b. The telegram [to the friend [of the soldiers]]

Since the distractor friends in (2a) is closer to the head noun telegram but further from the verb than the distractor soldiers in (2b), finding more errors for sentences like (2a) suggests that the interfering effect of the distractor is mediated by the subject head noun, as predicted by the percolation account. Comparable results for language comprehension have been provided by Nicol et al. (1997) and Pearlmutter (2000).

In sum, an important feature of the percolation account is that attraction errors are attributed to the process of integrating the distractor into the ongoing phrase structure representation whereas the agreement checking processes themselves can be assumed to work basically flawlessly. This, however, is not a necessary assumption. Given the complexity of the mental processes involved in checking subject-verb agreement, these checking processes might be disrupted by the presence of an intervening distractor bearing a number feature different from that of the controller.

### 2.2 Agreement Checking

Checking subject-verb agreement requires the following steps: after the finite verb has been encountered and its number feature has been registered, the number specification of the subject has to be retrieved from the syntactic structure build up so far. The retrieved number specification of the subject has then to be compared with the number specification of the verb. If a distractor intervenes between the head noun of the subject and the verb, errors might occur due to interference. In particular, instead of retrieving the number specification of the agreement controller, the number specification of the distractor might be retrieved, leading to an agreement error if distractor and controller differ in their number specifications.

Attributing attraction errors to interference would be in line with recent evidence showing that syntactic processing can be disrupted by interfering items. Interference effects have been found in configurations where the parser needs to retrieve some earlier information from the CPPM in order to integrate the current word. For example, Gordon and his colleagues have shown that interference contributes to the increased complexity of object-extracted relative clauses in contrast to subject-extracted relative clauses (cf. Gordon, Hendrick \& Johnson, 2001, 2004; Gordon, Hendrick \& Levine, 2002). In addition, van Dyke and Lewis (2003) have observed that recovery from a garden path can become
particularly difficult when the ambiguous region contains interfering items with similar properties like the actual retrieval target. Prima facie, the arguments cited above in support for percolation seem to discredit the idea of interference during the checking phase. First, this hypothesis seems hard to reconcile with the finding that the hierarchical proximity between controller and distractor is crucial for the occurrence of attraction errors, and not the linear distance between distractor and verb. If subject retrieval is achieved by a linear, backward search through the sentence, distractors being close to the verb should cause more attraction errors than distractors being more distant. However, this is a rather implausible scenario. McElree (2000) and McElree et al. (2003) have provided evidence that retrieval during sentence comprehension is mediated by a direct access mechanism, not by a search process. Under the assumption of a direct access mechanism, linear distance should be irrelevant for number attraction.

The second finding which seems to argue against interference during the checking phase being responsible for attraction errors is the observed asymmetry between singular and plural distractors. It is not obvious whether such an interference mechanism of number attraction would predict this asymmetry, even if we follow Eberhard's (1997) assumption that plural is specified by the presence of a plural feature whereas singular is specified by the absence of such a feature. Given this kind of number representation, the basic rule for subject-verb number agreement could be formulated as follows: A verb which is marked for singular agrees with a subject which is either completely unmarked or specifically marked for singular (e.g. by a singular quantifier like one); a verb which is marked for plural agrees with a subject which is also marked for plural.

Given this agreement rule, the occurrence of an asymmetry will depend on the particular retrieval cues for accessing the subject. If we assume that number is not among the retrieval cues, no asymmetry between singular and plural subjects is expected. Interference will then be possible with both singular and plural distractors. When a distractor intervenes between verb and controller, sometimes the distractor might be erroneously retrieved as the target for agreement. In this case the number specification of the distractor will be checked instead of the number specification of the controller. If controller and distractor do not have the same number specification, an agreement error results. Under the assumption that number is not among the retrieval cues, the probability of erroneously retrieving the distractor is independent of any number specification and thus no asymmetry between singular and plural is predicted.

## 3 Evidence for Disrupted Checking Processes

### 3.1 Relative Clause Constructions

Some piece of evidence that under certain circumstances attraction errors might indeed be due to processes occurring during the checking phase comes from attraction effects in relative-clause constructions (Häussler, Bader, and Bayer 2004). Examples are given in (3) and (4). Here, controller and distractor are separated by a clause boundary which should
block percolation. Nevertheless, experiments using the method of speeded-grammaticality judgments revealed attraction errors in both constructions.
a. dass der Professor, dessen Assistentinnen ich getroffen habe, angerufen hat that the professor whose assistants I met have called has '(I know) that the professor whose assistants I met has called.'
b. dass die Professoren, deren Assistentin ich getroffen habe, angerufen haben that the professors whose assistant I met have called have '(I know) that the professors whose assistants I met have called.'
a. Ich traf die Professoren, deren Assistentin angerufen hat.

I met the professors whose assistant called has
'I met the professors whose assistant called.'
b. Ich traf den Professor, dessen Assistentinnen angerufen haben.

I met the professor whose assistants called have
'I met the professor whose assistants called.'
In (3) the controller is the subject of an embedded clause with the finite auxiliary in clause-final position. The controller is modified by a relative clause attached to it. The distractor is the object of this relative clause. Whenever controller and distractor differed in number (professor whose assistants and professors whose assistant), participants made more judgment errors than for corresponding sentences with controller and distractor having the same number specification. Importantly, an attraction effect occurred for both singular and plural subjects. While a percolation account in combination with the assumption of an asymmetric number representation can explain the attraction effect with singular subjects and plural distractors, it fails for the reverse constellation.

In (4) the distractor is the head noun of the relative clause and precedes the controller, which is the head noun of relative-clause subject NP. If we assume that percolation applies in the process of integrating the distractor into the current partial phrase structure marker, no attraction effect should be observed in (4), since the processing of the distractor takes place before the subject is encountered. However, attraction errors occurred, and they occurred for plural subjects as well. Again, in the light of the evidence for an asymmetric representation of number, it is difficult to reconcile the attraction effects with plural subjects (i.e. singular distractors) with a percolation mechanism. For the checking phase, it is in principle conceivable that both singular and plural distractors interfere with the retrieval of the subject.

### 3.2 Mismatching Objects

Further evidence for checking being error-prone comes from simple subject object constructions reported in Häussler et al. (2005). Experiments investigating sentence comprehension have shown that attraction errors occur when subject and object differ in number, as illustrated in (5). Again, no asymmetry between singular and plural subjects was observed.
a. dass die Assistentin auch gestern die Professorinnen angerufen hat that the assistant also yesterday the professors called has '(I know) that the assistant called the professors yesterday too.'
b. dass die Assistentinnen auch gestern die Professorin angerufen haben that the assistants also yesterday the professor called have '(I know) that the assistants called the professor yesterday too.'

Note that both NPs in (5) are case-ambiguous. Further experiments have shown that unambiguous case marking of the subjects eliminates attraction errors and unambiguous case marking of the object reduces the amount of attraction errors. These results challenge the percolation account for at least three reasons: (i) when processing of the subject is completed, further material outside the subject NP should not affect its representation, (ii) given the asymmetric representation of number, only plural distractors should cause attraction, (iii) percolation should not be sensitive to case ambiguity. A checking account on the other hand can deal with these results: (i) checking requires retrieval of the subject and therefore items processed after the processing of the subject can interfere, (ii) interference can be caused by both singular and plural distractors as discussed above, (iii) since the retrieval cue is nominative case, retrieval is easy for unambiguous subjects, whereas for ambiguous subjects the parser sometimes is misguided by an object. The object is always at a disadvantage and therefore only rarely considered to be the subject. Unambiguous case marking reduces the probability even more.

### 3.3 Pronoun Resolution

The experiment investigates whether a plural distractor really turns a singular subject NP into a plural NP. If so, attraction should affect pronoun resolution. And more specifically, it should be possible to pick up a singular subject NP by a plural pronoun. A singular pronoun, on the other hand, should cause difficulties since the controller is no longer a singular NP and therefore does not show agreement with the pronoun. This prediction based on percolation is tested in the experiment by comparing sentences where attraction leads to an error concerning the subject-verb agreement and sentences where attraction leads to an error with respect to the agreement between a pronoun and its antecedent.

## Method

Participants. Forty-eight students of the University of Konstanz participated in the experiment. All of them were native speakers of German and not informed about the purpose

[^11]of the experiment. They received either course credits or were paid for their participation.
Procedure. The experiment was using the method of speeded-grammaticality judgments. Sentences were presented visually on a computer screen in a word by word fashion with each word appearing at the same position (mid-screen). Immediately after the last word of a sentence, participants had to judge the grammaticality of the sentence as quickly and as accurately as possible. Participants indicated their judgment by pressing one of two response buttons. If participants did not respond within 2,000 milliseconds, a red warning line " zu langsam" (too slow) appeared on the screen and the trial was finished.

Materials. Forty sentences were created. All sentences were introduced by a matrix clause. The object of this matrix clause was modified by a relative clause immediately following it. The relative clause was introduced by a possessive relative pronoun co-indexed with the head noun and therefore sharing its number specification and a singular noun which was either the subject or the object of the relative clause. In the first case for which (6) is an example, the relative clause was continued by a prepositional object and a verb plus auxiliary. The other case is illustrated in (7), here the relative clause is continued by a subject NP, a prepositional object, a verb plus auxiliary and a complement phrase containing the singular pronoun er ('he') referring to the relative clause initial NP. Each sentence appeared in eight versions according to the three experimental factors: (i) the distractor (the head noun of the relative clause, respectively the possessive relative pronoun) was either singular or plural, i.e. matched or mismatched the agreement controller in number, (ii) the type of agreement relation was either subject-verb agreement or antecedent-anaphor-agreement, (iii) the whole sentence was either grammatical or not. A set of grammatical examples is given in (6) and (7). Ungrammatical sentences were constructed by replacing the clause-final verb by a plural verb, resulting in an agreement violation at the end of the sentence.
(6) a. Gestern traf ich Maria, deren Sohn über die Abreise informiert wurde. yesterday met I M. whose son about the leaving informed was 'Yesterday I met Maria whose son was informed about the leaving.'
b. Gestern traf ich die Leute, deren Sohn über die Abreise informiert wurde. yesterday met I the people whose son about the leaving informed was 'Yesterday I met the guys whose son was informed about the leaving.'
a. Gestern traf ich Maria, deren Sohn man darüber informiert hat, yesterday met I M. whose son one about-it informed has
dass er abreisen muss. that he leave must
'Yesterday I met Maria whose son was informed that he has to leave.'
b. Gestern traf ich die Leute, deren Sohn man darüber informiert hat, yesterday met I the people whose son one about-it informed has
dass er abreisen muss.
that he leave must
'Yesterday I met the guys whose son was informed that he has to leave.'

## Results

|  | anaphor agreement |  |  | subject-verb agreement |  |
| :--- | :---: | :---: | :---: | :---: | :---: |
|  | Match | Mismatch |  | Match | Mismatch |
| grammatical | 90 | 88 |  | 90 | 81 |
| ungrammatical | 96 | 96 |  | 92 | 75 |

Table 1.1: Percentages of correct judgments in the experiment
Analyses of variance unfolded two main effects: a distractor effect ( $F_{1}=28.8, p<.001$; $F_{2}=26.4, p<.001$ ) and an effect of agreement type ( $F_{1}=22.6, p<.001 ; F_{2}=18.6$, $p<.001$ ), as well as a significant interaction of both factors $\left(F_{1}=12.1, p<.01 ; F_{2}=\right.$ $13.4, p<.001$ ). Grammaticality failed significance as a main effect ( $F_{1}=1.2, p=.28$; $F_{2}=2.6, p=.11$ ), but showed an interaction with the factor agreement type ( $F_{1}=9.6$, $\left.p<.01 ; F_{2}=6.6, p<.05\right)$. In addition, the three-way interaction reached significance ( $F_{1}=3.4, p<.1 ; F_{2}=3.2, p=.1$ ). The interaction of distractor and grammaticality failed significance (both $F s<1.1$ ).

As can be seen in Table 1.1, the anaphor-conditions and the subject-verb-conditions pattern differently. Planned comparisons revealed that attraction errors only occurred in the subject-verb agreement-conditions $\left(t_{1}=5.7, p<.001 ; t_{2}=6.0, p<.001\right)$, but not in the anaphor-conditions $\left(t_{1}<1, p=.46 ; t_{2}<1, p=.44\right)$.

## Discussion

The main finding of the experiment is the occurrence of an attraction effect for subjectverb agreement, and the absence of such an effect for the antecedent-anaphor relation. While a preceding distractor can cause errors in the agreement checking inside a relative clause, it does not affect anaphor resolution. Thus, the plural distractor does not turn the singular controller into a plural NP. As a consequence I assume that attraction does not occur during the construction phase of the relative clause subject, but rather during the checking phase of subject-verb agreement. The representation of the subject NP remains intact and is therefore available for a pronoun.

There are two potential objections to this conclusion: (i) the point at which the seeming agreement violation occurs is different for the two agreement-type conditions (the end of the sentence vs. the middlefield of the embedded clause), (ii) the grammatical status of agreement differs (obligatory subject-verb agreement vs. possible free interpretation of the pronoun). Both objections will be investigated in further experiments - (i) by using a self-paced reading procedure, and (ii) by testing anaphors in the sense of binding theory (principle A).

## 4 Summary and Conclusions

Prior research focused on leftward attraction in complex subject NPs, in which a singular subject is modified by plural distractor following the subject. The attraction error was attributed to a percolation process from the distractor NP to the subject NP. The experiments discussed above show that attraction errors can also be observed in configurations where they are not expected if percolation is the source of the error. Attraction occurs in relative-clause constructions, and in fact in both directions: leftward from a distractor inside the relative clause to a subject NP being the head noun of this relative clause, and rightward from a distractor being the head noun of the relative clause to the relative-clause subject following the distractor. Furthermore mismatching objects can cause attraction errors as well. Thus, attraction is not restricted to complex subject NPs. However, I would like to distinguish two kinds of attraction errors: the one found in prior research and the one described here. Despite the superficial similarity (occurring when subject and distractor differ in number), the former differs from the latter, especially with respect to the asymmetry between singular and plural subjects. The differences naturally follow if we assume different mechanisms underlying attraction errors. I would like to argue that we have evidence for two sources of attraction errors: (i) errors occurring during the phase of subject integration and (ii) errors occurring during the checking phase. Errors during the subject integration can be attributed to percolation mechanism. A crucial argument for percolation is the asymmetry between singular and plural. While this asymmetry was observed in complex subject NPs with the distractor being a modifier NP or an NP inside a PP, it did not show up in the relative-clause construction and in sentences with a distractor object. From the absence of the asymmetry I conclude that percolation is not the source of the error, but instead interference effects during the checking phase are responsible for attraction in these constructions. Further evidence for this claim was provided by the experiment in section 3.3 showing that this kind of attraction has no effect for pronoun resolution. The representation of subject NP remains intact.

Errors during the checking phase can be attributed to interference. Checking subjectverb agreement requires a retrieval of the subject. This retrieval is subject to interference of other items have similar properties as the actual retrieval target and therefore (at least partially) match the retrieval cue. Instead of retrieving the subject sometimes the distractor NP is retrieved. This results in an agreement error if subject and distractor do not match in number. From the pattern described in 3.2, we can derive further properties of the retrieval mechanism. In the object construction, attraction errors can be found with both singular and plural subjects. This is compatible with an asymmetric representation of number if we assume number not to be a retrieval cue. Furthermore, object attraction is sensitive to case marking. Unambiguous case marking on the subject facilitates subject retrieval and eliminates attraction. Therefore, we conclude that nominative is one of the retrieval cues.

An obvious question that comes up at this point is whether we really need to assume two mechanisms underlying attraction errors. I already argued that a percolation account cannot easily explain the findings for the constructions discussed here, neither attraction in
the relative-clause construction nor attraction emanating from objects. Thus, percolation cannot be the only explanation for attraction effects. Another option to reduce the number of mechanism is to explain all observed attraction errors with errors occurring during the checking phase. Such an account has difficulties to explain the asymmetry observed in the modifier construction. I therefore conclude that both mechanisms apply. This is not a costly assumption, since I do not assume an additional mechanism - checking has to be done anyway. I rather claim that there is just one further mechanism which is error prone. In addition to erroneous feature percolation, the checking mechanism itself can be disrupted by interference.

## Bibliography

Bock, K., K. M. Eberhard, J. C. Cutting, A. S. Meyer, and H. Schriefers (2001). Some attractions of verb agreement. Cognitive Psychology 43, 83-128.
Bock, K. and C. A. Miller (1991). Broken agreement. Cognitive Psychology 23, 45-93.
Branigan, H. P., S. P. Liversedge, and M. J. Pickering (1995). Verb agreement in written comprehension and production. Paper presented at the 8th Annual CUNY Sentence Processing Conference, Tucson, AZ.

Eberhard, K. M. (1997). The marked effect of number on subject-verb agreement. Journal of Memory and Language 36, 147-164.
Gordon, P. C., R. Hendrick, and M. Johnson (2001). Memory interference during language processing. Journal of Experimental Psychology: Learning, Memory, and Cognition 27, 1411-1423.
Gordon, P. C., R. Hendrick, and M. Johnson (2004). Effects of noun phrase type on sentence complexity. Journal of Memory and Language 51, 97-114.

Gordon, P. C., R. Hendrick, and W. H. Levine (2002). Memory-load interference in syntactic processing. Psychological Science 13, 425-430.

Hartsuiker, R. J., I. Antón-Méndez, and M. van Zee (2001). Object attraction in subjectverb agreement construction. Journal of Memory and Language 45, 546-572.

Häussler, J., M. Bader, and J. Bayer (2004). Number attraction effects - evidence from german relative clauses. Poster presented at the $1^{17}$ th Annual CUNY Sentence Processing Conference, College Park, MD.
Häussler, J., M. Bader, and J. Bayer (2005). A checking account of the number attraction effect. Poster presented at the 18th Annual CUNY Sentence Processing Conference, Tucson, AZ.

Hemforth, B. and L. Konieczny (2003). Proximity in agreement errors. In Proceedings of the 25th Annual Conference of the Cognitive Science Society.

Konieczny, L., S. Schimke, and B. Hemforth (2004). An activation-based model of agreement errors in production and comprehension. In Proceedings of the 26th Annual Conference of the Cognitive Science Society.

McElree, B. (2000). Sentence comprehension is mediated by content-addressable memory structures. Journal of Psycholinguistic Research 29, 111-123.

McElree, B., S. Foraker, and L. Dyer (2003). Memory structures that subserve sentence comprehension. Journal of Memory and Language 48, 67-91.

Nicol, J. L., K. I. Forster, and C. Veres (1997). Subject-verb agreement processes in comprehension. Journal of Memory and Language 36, 569-587.

Pearlmutter, N. J. (2000). Linear versus hierarchical agreement feature processing in comprehension. Journal of Psycholinguistic Research 29, 89-98.

Pearlmutter, N. J., S. M. Garnsey, and K. Bock (1999). Agreement processes in sentence comprehension. Journal of Memory and Language 41, 427-456.

Van Dyke, J. and R. Lewis (2003). Distinguishing effects of structure and decay in attachment and repair: A cue-based parsing account of recovery from misanalyzed ambiguities. Journal of Memory and Language 49, 285-316.

# A Glue/ $\lambda$-DRT Treatment of Resumptive Pronouns 

Miltiadis Kokkonidis<br>Computational Linguistics Group, University of Oxford<br>miltiadis.kokkonidis@clg.ox.ac.uk


#### Abstract

Anaphora and resumption, phenomena which intuitively seem to challenge resource sensitivity, have been argued to support its role in Glue. The resource deletion treatment of resumption in LFG Glue presented by Asudeh (2004) was designed to work together with the Glue resource duplication treatment of anaphora (Dalrymple et al., 1999). However, as 'resumptive pronouns' are mere pronouns, the choice of a treatment of anaphora that has long been found to be inadequate creates an issue. A modular treatment of pronouns using LFG, Glue, and $\lambda$-DRT (Kokkonidis, 2005) is the most promising alternative. A treatment of resumption for this treatment of pronouns is presented together with new insights about resource management, the new and the old treatment of resumption, and the intuitions behind them.


## 1 Introduction

It has been more than a decade since Dalrymple et al. (1993) presented Glue (Dalrymple, 2001; Kokkonidis, 2006), a compositional semantics framework for LFG (Kaplan and Bresnan, 1982; Dalrymple, 2001) based on linear logic (Girard, 1987), a resource-sensitive logic. Having a resource logic driving semantic composition suits the unordered nature of fstructure, while preserving the fundamental semantic principle (underlying also Montaguestyle compositionality) that the meaning of a phrase is formed by combining its semantic contributions using each of them once. A semantic version of the completeness and coherence principles (Dalrymple, 2001) is built into the formal system for meaning assembly.

Anaphora is a linguistic phenomenon that at first seems to provide counter-evidence for a hypothesis of linguistic resource-sensitivity: there is no limit to how many times an antecedent (a semantic resource) can be referred to (re-used). However, Dalrymple et al. (1999) argued that there is actually no conflict between the phenomenon of anaphora and the goal of a resource-sensitive semantic composition framework and gave a resource management analysis of pronouns in Glue. This analysis was interesting, but not without problems. As a result, a number of alternative analyses have appeared (Crouch and van Genabith, 1999; Dalrymple, 2001; van Genabith and Crouch, 1997; Kokkonidis, 2005).

Asudeh (2004) defines resumption as the existence of a surplus pronominal resource. This again creates a situation that at first seems to challenge the ability of a resourcesensitive approach to deal with the linguistic phenomenon at hand: if all semantic resources
must be used exactly once, what happens when surplus resources appear in the sentence being analysed? It turns out there is an answer to this question also. Moreover, Asudeh (2004) puts forward the proposition that his resource management analysis given in a resource-sensitive semantic composition framework owes much to the very fact that it is given in such a framework. Resource-sensitivity is thus once again argued to be a strength, not a weakness of Glue.

Kokkonidis (2005) provides an analysis of anaphora based on the classic DRT (Kamp, 1981; Kamp et al., 2005) anaphoric resolution mechanism. This analysis bypasses Glue and its resource sensitivity and is accompanied by argumentation that suggests that anaphoric resolution is not best addressed at the semantic composition layer. But as that analysis of anaphora avoids resource management while the analysis of resumption (and copy raising) of Asudeh (2004) owes much to it, the two would seem to be at odds with each other. This conflict leads to an unsatisfactory state of affairs. Ideally we would want to have the benefits of both analyses, but it is not immediately obvious how the two can be combined. Far from that, there seem to be statements in the work of Asudeh (2004) that indicate that something like that may not be possible at all. The present work shows it is.

While Kokkonidis (2005) argues that the Glue resource management approach creates more problems than it solves for anaphora and that anaphoric context management should be left to a dynamic semantic representation instead, not Glue, Asudeh (2004) attributes the success of his theory to it (p. 11):

The resource management theory is a unified theory of resumption that accounts for both resumptive pronouns and copy raising in resource logical terms, while maintaining key differences between the two phenomena that have blocked unified analyses.
While for Kokkonidis (2005) a pronoun has a Glue type like any other noun phrase, Asudeh (2004) claims that the fact that in the treatment of anaphora he uses (Dalrymple et al., 1999) pronouns have Glue types of a different form than any other semantic contribution is essential (p. 13):

Only pronouns can be used in resumption because they are the only things that have the correct form to be consumed by manager resources.
Asudeh uses a very simple analysis of pronouns. In the case of sophisticated analyses of pronouns (or indeed epithets or other noun phrases that can appear where resumptive pronouns can, assuming they are meant to be covered by the same analysis that covers resumptive pronouns) one would want their semantics incorporated into the meaning rather than deleted (p. 13):

Pronominal elements can be consumed by manager resources because it is precisely these elements whose removal is recoverable from elsewhere in the semantics.

Sections 2 and 3 outline the analyses of anaphora of Dalrymple et al. (1999) and Kokkonidis (2005) respectively. Section 4 presents a slightly improved version of the original treatment of resumption by Asudeh (2004). Section 5 shows that superficial differences between types can be deceiving, an essential insight useful throughout this paper. Section 6 presents an analysis of resumption which works in the LFG/Glue/ $\lambda$-DRT setting of Kokkonidis (2005). Section 7 summarises the new technical results and insights obtained.

## 2 The Resource Duplication Analysis of Pronouns

The point of using linear, rather than, say, intuitionistic, types in Glue is that linear logic is resource sensitive. However, a discourse referent that is in the current context can be referenced any number of times. There may seem to be a problem with trying to treat anaphora within Glue. The antecedent-entity duplication treatment of anaphora of Dalrymple et al. (1999), the first ever analysis performing Glue resource management, addresses this issue, but this issue alone, which is why alternative treatments appeared (Crouch and van Genabith, 1999; Dalrymple, 2001; van Genabith and Crouch, 1997; Kokkonidis, 2005).

The idea behind it was simple: anaphoric resolution does not come down to free re-use of resources; it is the pronoun (overt or not) that actually triggers and manages this 'reuse'. If the pronoun corresponds to label $p$ and its antecedent to label $a$ then its type will be $e_{a} \multimap e_{a} \otimes e_{p}$. The meaning of a pronoun is that of a duplicating function $\lambda x$. $(x, x)$. What this achieves is a controlled duplication of the antecedent in the Glue context. This analysis in particular and the general resource management idea behind it respectively are what Asudeh (2004) takes as the setting and the intuition for his treatment of resumption.

It is now possible to return to the question of how anaphora can be reconciled with resource sensitivity. In a sentence such as (1) it is not the case that the semantic contribution of 'John' can be used twice. ${ }^{1}$ Glue's resource sensitivity would not allow it. However, the semantic contribution of the pronoun has a complex meaning $(\lambda x .(x, x))$. This clearly shows that the semantic representation language, in which the meanings of semantic contributions, larger units, and eventually the sentence are expressed, is not resource-sensitive. Meaning expressions themselves have no resource-sensitivity constraints, but the composition of meaning expressions into more complex ones does. Glue's resource sensitivity does not affect at all the ability of expressing anaphoric binding within the chosen semantic representation.
(1) John likes himself.

Glue typing judgement
john: $e_{s}$, like : $\left(e_{s} \otimes e_{o}\right) \multimap t_{f}$, himself $: e_{s} \multimap e_{s} \otimes e_{o} \vdash$ like (himself john) : $t_{f}$
FOL SEmantic representation expression
LIKE (JOHN, JOHN)
(2) Everyone likes himself.

## Glue typing Judgement

```
everyone : \(\forall \alpha\). \(\left(e_{s} \multimap t_{\alpha}\right) \multimap t_{\alpha}\),
    like \(:\left(e_{s} \otimes e_{o}\right) \multimap t_{f}, \vdash\) everyone \(\lambda\). like (himself \(\left.x\right): t_{f}\)
    himself: \(e_{s} \multimap e_{s} \otimes e_{o}\)
    FOL SEmantic representation expression
    \(\forall x\). PERSON \((x) \Rightarrow \operatorname{LIKE}(x, x)\)
```

[^12]With reference to LFG, an advantage of this approach (and those of Crouch and van Genabith (1999) and Dalrymple (2001) which also place anaphoric resolution at the semantic composition level) over the one combining Glue with CDRT described by van Genabith and Crouch (1997) is that it can capture syntactic constraints on anaphora. This is done by a constraint on what can be the pronoun's antecedent. However it has problems in treating intrasentential anaphora (Dalrymple et al., 1999; Dalrymple, 2001; Kokkonidis, 2005). As resumption is an intrasentential phenomenon, Asudeh (2004) whose concern was to best present his treatment of resumption did well to choose the simplest analysis of anaphora available at the time that enabled him to express antecedent constraints needed in his analysis even if that analysis has problems in dealing with intersentential anaphora and covering the spectrum of pronoun interpretation. But, as resumptive pronouns are ordinary pronouns, the analysis of pronouns used as the basis of an analysis of resumption must be one that covers anaphora as fully as possible. The state of the art is not very advanced, but the following analysis is possibly a good starting point for future work.

## 3 A Glue $+\lambda$-DRT analysis of pronouns

Kokkonidis (2005) presented a modular analysis of pronominal anaphora using LFG, Glue and $\lambda$-DRT that deals with a number of issues more successfully and arguably more elegantly than preceding analyses.

If there is a pronoun and it makes up a noun phrase labelled $p$, then its type will be $\forall \alpha$. $\left(e_{p} \multimap t_{\alpha}\right) \multimap t_{\alpha}$. The dynamic semantics will take care of the anaphora. The meaning assigned to a pronoun ${ }^{2}$ found at f -structure $p$ will be:

$$
\lambda P . \begin{array}{|l|}
\hline \mathrm{x}: \mathrm{e}_{\hat{p}} \\
\hline \mathrm{x}=? \\
\hline
\end{array} \sqcup P(x)
$$

It will be up to DRT (augmented with a simple type system for discourse referents) to resolve ? to an accessible discourse referent of a type compatible with the mandates of any syntactic constraints captured in terms of labels and the ${ }^{\wedge}$ function mapping f-structures to anaphoric indices. Kokkonidis (2005) provides more details. Here is an example of a sentence and its anaphorically resolved DRS, whereby the ? has been replaced by the only accessible discourse referent that obeys the constraint imposed by the syntactic properties of 'himself', namely that it is co-indexed with 'everyone': ${ }^{3}$

Glue typing Judgement

```
everyone : \(\forall \alpha .\left(e_{s} \multimap t_{\alpha}\right) \multimap t_{\alpha}\),
    like \(:\left(e_{s} \otimes e_{o}\right) \multimap t_{f}, \vdash\) everyone \(\lambda x\). himself \(\lambda y\). like \(x y: t_{f}\)
    himself: \(\forall \beta .\left(e_{o} \multimap t_{\beta}\right) \multimap t_{\beta}\)
```

[^13]
## DRT SEMANTIC REPRESENTATION EXPRESSION



Note that the combination of LFG syntactic constraints and DRT semantic form constraints rules out the reading whereby 'himself' outscopes 'everyone'. It also rejects the co-indexing of (3). There the constraint preventing co-indexation is one of accessibility. The idea of combining syntactic and semantic form constraints is an elegant and powerful one.
(3) ${ }^{*}$ Nobody $_{1}$ came. $H e_{1}$ laughed.

## 4 The deletion treatment of resumption

For our discussion of resumption we will be using the following example from Irish. In Irish there are two different versions of ' $a$ ', referred to as a $L$ and a $N$, which behave differently with respect to resumption. Oversimplifying, one can say that a $N$ licences resumptive pronouns while $a L$ does not. Here we have an example involving $a N$ and a resumptive pronoun (é).
(4) an scríbneoir a molann na mic léinn é the writer a $N$ praise the students him the writer who the students praise (him)

$$
z:\left[\begin{array}{ll}
S P E C & \text { 'THE' } \\
P R E D & \text { 'WRITER' } \\
A D J & \left\{c:\left[\begin{array}{lll}
P R E D & \text { 'PRAISE' } \\
T O P I C & w:[P R E D & \text { 'PRO' } \\
S U B J & s:\left[\begin{array}{ll}
\text { "the students" }]
\end{array}\right] \\
O B J & o:\left[\begin{array}{ll}
P R E D & ' P R O ' \\
P E R S & 3 \\
N U M & \text { sg } \\
G E N D & \text { masc }
\end{array}\right]
\end{array}\right]\right.
\end{array}\right]
$$

$$
\begin{aligned}
& t h e_{z}: \forall \alpha .\left(e_{z} \multimap t_{z}\right) \multimap\left(e_{z} \multimap t_{\alpha}\right) \multimap t_{\alpha}, \\
& \text { writer : } e_{z} \multimap t_{z} \text {, } \\
& \text { rel : }\left(e_{w} \multimap t_{c}\right) \multimap\left(\left(e_{z} \multimap t_{z}\right) \multimap\left(e_{z} \multimap t_{z}\right)\right) \text {, } \\
& \Gamma={ }^{\text {the }}: ~ \forall \beta .\left(e_{s} \multimap t_{s}\right) \multimap\left(e_{s} \multimap t_{\beta}\right) \multimap t_{\beta}, \\
& \text { students }: e_{s} \multimap t_{s} \text {, } \\
& \text { praise }: e_{s} \multimap e_{o} \multimap t_{c} \text {, } \\
& \text { him : } e_{w} \multimap e_{w} \otimes e_{o} \text {, } \\
& \text { mngr: }\left(e_{w} \multimap e_{w} \otimes e_{o}\right) \multimap\left(e_{w} \multimap e_{o}\right)
\end{aligned}
$$

Asudeh (2004) regards resumptive pronouns as 'surplus resources': a Glue derivation that would have been possible with a gap is no longer possible when a extraneous resumptive pronoun appears. In places where they are not meant to appear, this is exactly what he wants his analysis to predict. Indeed, he relies on Glue to complement syntax. ${ }^{4}$ However, where resumptive pronouns are allowed, Glue derivations should be possible. If the treatment for, say, relative clauses is such that no resumptive pronouns are expected then when they appear, they indeed become extraneous for Glue. This is the setup Asudeh (2004) assumes. The solution he proposes is simple: as the surplus resource (the resumptive pronoun) prohibits a Glue derivation, its presence should be neutralised if and only if it is permitted according to the rules of the language. This task falls upon the resumption licenser. There are two flavours of deletion that I am aware of. In the one proposed by Asudeh (2004) it is the pronoun itself that is 'deleted', whereas in a variation of that (Mary Dalrymple, 2006, personal communication), it is its effect that is 'deleted' instead. Only Asudeh's version will be considered here.

As a matter of fact what is called here a deletion treatment of resumption is not meant to be literally taken as involving deletion of anything either from the f-structure or from the Glue typing context. What happens instead is that a semantic contribution, a manager resource as Asudeh calls them, is added to the typing context, courtesy of the resumption licenser. The only role of this resource manager is to consume the resumptive pronoun and return nothing or something very similar to nothing. 'Nothing' in this context is ()$: 1$ as found in both linear and intuitionistic logic, but an alternative value, something very similar to 'nothing', the identity function ( $\lambda x . x$ ) with glue type $e_{a} \multimap e_{a}$ (where $a$ is the label of the antecedent of the resumptive pronoun) is actually a better choice in terms of not introducing additional complexity in Glue and in the resulting terms. So with reference to example (4), Asudeh's resource manager would be mngr : $\left(e_{w} \multimap e_{w} \otimes e_{o}\right) \multimap\left(e_{w} \multimap e_{w}\right)$. Its semantics would be $\lambda p . \lambda x . x$ where one can see that $p$ i.e. the semantic expression corresponding to the resumptive pronoun is consumed and made no use of, which is why I call this a deletion treatment of resumption.

But the fact that there is a surplus contribution for a resumptive pronoun in the typing context is not the only way that that pronoun manifests itself in a way that affects the Glue derivation. Asudeh's resource manager will combine with that resource giving an identity function that affects neither the derivation nor the semantics. As far as resource

[^14]accounting goes it will be as if that surplus resource never existed. But the presence of the resumptive pronoun is also evident in the f-structure where it fills a position that would have otherwise been linked to another part of the f-structure. In the resumption-less version of (4) the object $o$ of the inner clause $c$ is the topic $w$ i.e. $o=w$. In (4) where we have a resumptive pronoun in $o$, the f-structures for $o$ and $w$ are distinct. Because of this difference in the f-structure, the effect of the resumptive pronoun is manifest in the Glue typing context in a way that Asudeh's resource manager does not do anything about. It does consume the contribution of the resumptive pronoun, but its presence is still evident in the Glue type of the relative clause. If we consider the first six elements (that is ignoring the resumptive pronoun and its corresponding resource manager) of the typing context for (4) we see that rel expects an argument of type $e_{w} \multimap t_{c}$ but $e_{w}$ is mentioned nowhere else among those six elements. Asudeh's solution is to introduce a relabeler resource relab : $\left(e_{o} \multimap t_{c}\right) \multimap\left(e_{w} \multimap t_{c}\right)$ with semantics $\lambda P$. P. A simpler version of that is relab : $e_{o} \multimap e_{w}$ with semantics $\lambda x$.x.

Simplifying Asudeh's treatment one step further, one can simply provide instead of a contribution that deletes the resumptive pronoun and another that performs relabelling a single resource manager that does both. This has the exact same semantics as Asudeh's but its Glue type is a bit different: $\left(e_{w} \multimap e_{w} \otimes e_{o}\right) \multimap\left(e_{w} \multimap e_{o}\right)$. This is what was used in the typing context for (4). It has the form (resumptive pronoun) $\multimap$ (relabeler). This is preferable to the proposal of Asudeh (2004) of having a combination of a resource manager ((resumptive pronoun) $\multimap$ (identity function)) and a relabeler, both because there is no need to introduce an identity function in the process of 'deleting' the resumptive pronoun and because one instead of two semantic contributions are made by the resumption licenser.

Asudeh (2004) notes that in order to account for what are called mixed chains in Irish a $N$ may need to only contribute a relabeler rather than both that and a resource manager. This is expressed naturally also in our view: either a $N$ consumes a resumptive pronoun and does relabelling (mngr : $\left.\left(e_{w} \multimap e_{w} \otimes e_{o}\right) \multimap\left(e_{w} \multimap e_{o}\right)\right)$ or it only does relabelling (relab : $e_{o} \multimap e_{w}$ ).

## 5 The issue of form

Pronouns in the antecedent-entity duplication approach of Dalrymple et al. (1999) have Glue types of a different form than any other semantic contribution. According to Asudeh (2004) this difference in form prohibits, at the Glue layer, anything other than a pronoun being licensed by resumption licensers. Such a difference in form between pronouns and other phrases does not exist when Glue is combined with $\lambda$-DRT (Kokkonidis, 2005). A Glue treatment of $\lambda$-DRT resumptive pronouns must ideally offer all the benefits of the treatment proposed by Asudeh (2004) for antecedent-entity duplicating pronouns. An $a$ priori criticism that emerges for any Glue treatment of $\lambda$-DRT resumptive pronouns is that, as pronouns will have types similar in form to those of other noun phrases, a resource manager will not be able to distinguish a resumptive pronoun from, say, a proper name: if we take any grammatical sentence with a resumptive pronoun and replace that pronoun
with a proper name, their types will be the same and Glue will not be able to tell the difference and reject that sentence. This is certainly true. But that would also have been the case with the original analysis of Asudeh (2004). Despite appearances, antecedent entity duplicating pronouns are not "the only things that have the correct form in the resource logic to be consumed by manager resources" in Asudeh's analysis either.

To understand why Glue can not make this distinction, we should note that both a quantifier $\forall \alpha$. $\left(e_{x} \multimap t_{\alpha}\right) \multimap t_{\alpha}$ and a pronoun $e_{a} \multimap e_{a} \otimes e_{x}$ (in the antecedent entity duplication analysis of Dalrymple et al. (1999)) are indirect (and somewhat constrained) ways of offering $e_{x}$. If we have an entity $\mathrm{e}_{a}$ corresponding, say, to a proper name, it is possible to derive an expression with a quantifier type (good old type raising):

$$
\text { alonso }: e_{x} \vdash \lambda P \text {. P alonso }: \forall \alpha .\left(e_{x} \multimap t_{\alpha}\right) \multimap t_{\alpha} .
$$

Similarly it is possible to obtain an expression with a pronoun type:

$$
\text { alonso }: e_{x} \vdash \lambda x .(x, \text { alonso }): e_{a} \multimap e_{a} \otimes e_{x} .
$$

The power of Glue makes superficial differences in the form of types irrelevant. If it is only pronouns that we want resumption licensers to allow, this can be specified by means of a syntactic form constraint.

## 6 A Glue/ $\lambda$-DRT treatment of resumption

The two new kinds of semantic contributions Asudeh (2004) introduced, namely resource managers and relabelers, were fairly simple and corresponded well to his intuition. The problem with simple solutions is that they are best appreciated when presented after confused complex alternatives. The Glue/ $\lambda$-DRT treatment of resumption is also trivial, but a number of less trivial alternatives were rejected before it emerged. What is also interesting is how similar it is to a treatment of resumption in the setting of Dalrymple et al. (1999) not considered previously.
(5) an scríbneoir a molann na mic léinn é the writer a $N$ praise thee students him the writer who the students praise (him)

$$
z:\left[\begin{array}{ll}
S P E C & \text { 'THE' } \\
P R E D & \text { 'WRITER' } \\
A D J & \left\{c:\left[\begin{array}{lll}
P R E D & \text { 'PRAISE' } \\
T O P I C & w:[P R E D & \text { 'PRO' }] \\
S U B J & s:\left[\begin{array}{ll}
\text { "the students" }]
\end{array}\right. \\
O B J & o:\left[\begin{array}{ll}
P R E D & ' P R O ' \\
P E R S & 3 \\
N U M & \text { sg } \\
G E N D & \text { masc }
\end{array}\right]
\end{array}\right]\right.
\end{array}\right]
$$

$$
\begin{aligned}
& \text { the }: \forall \alpha \cdot\left(e_{z} \multimap t_{z}\right) \multimap\left(e_{z} \multimap t_{\alpha}\right) \multimap t_{\alpha}, \\
& \\
& \text { writer }: e_{z} \multimap t_{z}, \\
& \\
& \text { rel }:\left(e_{w} \multimap t_{c}\right) \multimap\left(\left(e_{z} \multimap t_{z}\right) \multimap\left(e_{z} \multimap t_{z}\right)\right), \\
& \Gamma= \\
& \text { the }: \forall \beta \cdot\left(e_{s} \multimap t_{s}\right) \multimap\left(e_{s} \multimap t_{\beta}\right) \multimap t_{\beta}, \\
& \text { students }: e_{s} \multimap t_{s}, \\
& \\
& \text { praise }: e_{s} \multimap e_{o} \multimap t_{c}, \\
& \\
& \text { him }: \forall \gamma \cdot\left(e_{o} \multimap t_{\gamma}\right) \multimap t_{\gamma}, \\
& \\
& \text { mngr }: ? ? ?
\end{aligned}
$$

Let us recap. In example (6) we have a common noun ("scríbhneoir", Glue: writer : $e_{z} \multimap t_{z}$ ), which corresponds to $R$, a function from entities to DRSs that is to be modified by the relative clause. We also have a complete inner clause ("molann na mic léinn é", Glue: the students $\lambda$ s.him (praise $s$ ) or him $\lambda h$. the $e_{s}$ students $\lambda$ s.praise $s h$ ). That inner clause corresponding to English "the students praise him" is complete. It can be given two Glue readings both of which correspond to the same DRS C. Note that in this view the resumptive pronoun is not a surplus resource; its contribution namely a new discourse referent, let us call it $\chi$, and a condition $\chi=$ ? have been incorporated into $C$. Note also that in this view the presence of a resumptive pronoun does not necessitate any mention of relabelling. Finally, the meaning of $R$ modified by the relative clause is simply $\lambda x .(R x) \sqcup C$ where ? is somehow instantiated to whatever $x$ is.

We know that $x$ will be the discourse referent introduced by the meaning of $t h e_{z}$. Let us call it $\zeta: e_{\hat{z}}$. Using the coindexation constraints of Kokkonidis (2005) and knowing that $\hat{w}=\hat{z}$, all that needs to be added is a constraint $\hat{o}=\hat{w}$. Even without a discourse referent for the relativiser, the transitivity provided by this setup for syntactic constraints on anaphora, guarantees that $\hat{o}=\hat{z}$ which means that in $\chi=$ ?, ? can only be $\zeta$ (or a discourse referent equal to $\zeta$ ). This takes care of one issue. The other is actually merging those two DRSs.

This is the job of the meaning of rel : $\left(e_{w} \multimap t_{c}\right) \multimap\left(\left(e_{z} \multimap t_{z}\right) \multimap\left(e_{z} \multimap t_{z}\right)\right)$ with semantics $\lambda F . \lambda R$. $\lambda x$. $(F x) \sqcup(R x)$. But $C: t_{c}$ is complete as it is, it does not depend on $x$ in the sense a function from entities to DRSs would normally depend on its argument although it does in a different way as its metavariable ? will be equal to whatever $x$ is thanks to the $\hat{w}=\hat{z}$ constraint. If we want to avoid changing mngr, what we can do is introduce

$$
m n g r: t_{c} \multimap e_{w} \multimap t_{c}
$$

with semantics $\lambda C . \lambda x$. $C$. Notice that what we would normally want, to ensure that $C$ depends on $x$ is not done through function application here, but through anaphoric resolution within the DRS $C$. As $x$ is not used in the semantics, the new analysis given resembles a resource deletion analysis.

Let us provide such an analysis also. We will take something as being a surplus resource. But instead of claiming that this is the entity of the resumptive pronoun, the way we have described things it is the entity the relativiser offers to fill the gap no longer there in the relative clause that is a surplus resource now. This explanation, to repeat the point once again, comes with no need to refer to a concept of relabelling. The manager resource needs
to have a type that consumes $e_{w}$ and return nothing or, for practical reasons, something close to nothing i.e. an identity function. One fairly attractive possibility is

$$
m n g r: e_{w} \multimap t_{c} \multimap t_{c}
$$

If we regard this and the previous version of the manager resource as functions of two arguments, the only difference is the order of the arguments. The corresponding semantics for this version is $\lambda x \lambda C . C$. Another possibility, with identical (modulo types) semantics i.e. $\lambda x \lambda y . y$, is

$$
m n g r: e_{w} \multimap e_{o} \multimap e_{o}
$$

This is interesting for another reason. The effect deletion treatment of resumption (Mary Dalrymple, 2006, personal communication) in the setting with the pronouns of Dalrymple et al. (1999) for this example would give a resource manager

$$
m n g r:\left(e_{w} \otimes e_{o}\right) \multimap e_{w}
$$

If we were to apply the same idea of combining that with the relabeler as we did for the proposal of Asudeh (2004), we would get

$$
m n g r:\left(e_{w} \otimes e_{o}\right) \multimap e_{o}
$$

The curried version of that is

$$
m n g r: e_{w} \multimap e_{o} \multimap e_{o}
$$

The pronouns were different, the idea was to delete rather than use the pronoun meaning, what was considered as a 'surplus resource' was the $e_{o}$ added by the resumptive pronoun rather than the $e_{w}$ supplied by the relative clause construction, yet the result is the same! This brings us back to the question of form: pronouns in both approaches are there to offer an entity, of type $e_{o}$ in the given example. It should not come as a surprise that resource managers designed for one kind of pronoun will work with the other.

What really changes when $\lambda$-DRT pronouns are used is the anaphoric coindexing constraint for the benefit of label-sensitive DRT anaphoric resolution (Kokkonidis, 2005) that replaces a similar constraint expressed in terms of an ANTECEDENT feature used with the antecedent-entity duplication treatment of anaphora (Asudeh, 2004). This difference becomes more interesting when mixed chains in Irish are investigated, but space limitations prohibit such a venture here.

## 7 Conclusions

The aim of this work was to provide a Glue treatment of resumption that would work with $\lambda$-DRT pronouns, but would retain all advantages of the one Asudeh (2004) gave for antecedent-entity duplicating pronouns (Dalrymple et al., 1999). This seemed to be a challenge; some apparent obstacles seemed easy to overcome, some not.

Before contemplating a treatment of resumption for the LFG/Glue/ $\lambda$-DRT setting, it seemed that it would be impossible for any such treatment to have a certain property Asudeh's treatment had hitherto been assumed to have. However, further investigation showed Asudeh's system did not have the property that a pronoun is the only kind of Glue semantic contribution a resource manager may consume as its argument either. The only difference is that this is obvious from the start in the new setting, but obscured by the superficial difference in form between the types typically associated with pronouns as opposed to those typically associated with other noun phrases in the original setting. That pronouns are a syntactic concept means that that if it is only them that we want resumption licensers to allow, syntax should have been where we would normally first try to place such a constraint anyway. That Glue can not help here is not really an issue.

It is possible to argue, even convincingly perhaps, that there is no conflict between the aversion of Kokkonidis (2005) to resource management in the antecedent-entity duplication treatment of anaphora and a treatment of resumption along the lines of Asudeh's original resource management (deletion) treatment because the later manages (deletes) resumptive pronouns whatever they may be. Something we have not seen is that trying to consume a $\lambda$-DRT pronoun will only work sometimes, that is in some Glue readings. What we have seen though is that resource managers designed for the LFG/Glue/ $\lambda$-DRT setting can be independently motivated in the antecedent-entity duplication setting and vice versa. This is a much stronger response to the question of what problems there will be when trying a resource management approach for resumption in a setting where anaphora does not involve resource management at the Glue level: there is no issue and no difference whatsoever. The new treatment is, as far as the resource managers go, identical to a treatment deleting the entity resource contributed by a pronoun. Even Asudeh's original 'pronoun consuming' resource managers can be transferred without modification to the new setting. It is Glue's flexibility that makes exact form irrelevant in most cases and the fact that pronouns in both settings offer an entity resource that is the reason of this similarity. Also the fact that one can choose not to take the approach that the pronoun is surplus and still get the same results is due to the fact than in the presence of resumption there are two equal entities competing for one place so which one is chosen is irrelevant. Different intuitions lead to the same result.

Finally, given their similarity an expectation that the new analysis that did not try to delete the resumptive pronoun but use its meaning would have an advantage over the 'pronoun deletion' approach of Asudeh (2004) when the pronoun is replaced by something that does carry more meaning, such as an epithet, is now easily seen not to be true. An independent clue could have been that his resource managers only involve entities, not statements about those entities, in effect deleting one. The whole meaning expression of a pronoun may not be recoverable from elsewhere but this entity, being equal to another, is.

The goal of obtaining a Glue analysis for $\lambda$-DRT pronouns was achieved not only thanks to the Glue analyses the paper concentrated on, but also thanks to the simple and elegant way of syntactically restricting DRT anaphoric resolution of Kokkonidis (2005). It is thanks to that that only correct DRT readings are obtained after anaphoric resolution. In retrospect it is this part of the analysis that is most important rather than the trivial

Glue resource managers. With respect to Glue, the insight developed by investigating the topic from a different perspective was the main benefit, more important than the new simpler resource managers surprisingly applicable in both the old and the new settings. The original goal was achieved, albeit in an unexpected manner. We set out to sail to the end of the world only to come back where we came from; but we now know that it is round.

## Acknowledgements

I wish to thank Mary Dalrymple and Ash Asudeh for a number of interesting discussions that have greatly influenced this paper and the two anonymous reviewers for their comments and suggestions.

## Bibliography

Asudeh, A. (2004). Resumption as resource management. Ph.D. dissertation, Stanford University.
Crouch, R. and J. van Genabith (1999). Context change, underspecification, and the structure of glue language derivations'. See Dalrymple (1999).
Dalrymple, M. (Ed.) (1999). Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach. MIT Press.
Dalrymple, M. (2001). Lexical Functional Grammar. Number 42 in Syntax and Semantics Series. Academic Press.

Dalrymple, M., J. Lamping, F. C. Pereira, and V. Saraswat (1999). Quantification, anaphora, and intensionality. See Dalrymple (1999).

Dalrymple, M., J. Lamping, and V. Saraswat (1993). LFG semantics via constraints. In Proceedings of the Sixth Meeting of the European ACL, University of Utrecht, pp. 97-105. European Chapter of the Association for Computational Linguistics.
Fry, J. (1999). Proof nets and negative polarity licensing. See Dalrymple (1999).
Girard, J.-Y. (1987). Linear logic. Theoretical Computer Science 50, 1-102.
Kamp, H. (1981). A theory of truth and representation. Formal Methods in the Study of Language.
Kamp, H., J. van Genabith, and U. Reyle (2005). Discource representation theory. Draft of chapter in the forthcoming Gabbay, D. and Guenther, F. (eds.) Handbook of Philosophical Logic, Vol.2, Springer.
Kaplan, R. M. and J. Bresnan (1982). Lexical functional grammar: A formal system for grammatical representation. In J. Bresnan (Ed.), The Mental Representation of Grammar Relations, pp. 173-281. MIT Press.
Kokkonidis, M. (2005). Why glue your donkey to an f-structure when you can constrain and bind it instead? In M. Butt and T. H. King (Eds.), Proceedings of the LFG05 Conference. CSLI Publications.

Kokkonidis, M. (2006). First-order Glue. Journal of Logic, Language and Information. To appear.
van Genabith, J. and R. Crouch (1997). How to glue a donkey to an f-structure or porting a dynamic meaning representation language into LFG's linear logic based glue language semantics. In Proceedings of the International Workshop for Computational Semantics, Tilburg, pp. 52-65.

# Only Scalar 

ARndt Riester<br>IMS, Universität Stuttgart<br>arndt.riester@ims.uni-stuttgart.de


#### Abstract

In this paper I want to present a new proposal for a unified meaning of the particle only which accounts for the traditional distinction between a quantificational and a scalar only. It therefore avoids an unintuitive lexical ambiguity and tries to capture a generalization missed so far. For this purpose, different stages in the history of only will be reviewed, portraying the increasing complexity of the matter. As I will attempt to reduce the different meanings of only to a single one, a contextual parameter is necessary which imposes different orders on the associated alternative set. It is these different orders from which the different readings derive.


## 1 Introduction: Defining Only

It is a well-known fact that stating the meaning of only as defined in Horn (1969) and taken up in early versions of Alternative Semantics (Rooth (1985); Rooth (1992)), cf. (1), leads to wrong predictions for sentences containing focus on conjunctions as in (2a).

$$
\begin{equation*}
\operatorname{only}(\phi, C)=\{w \in \phi \mid \neg \exists \psi \in C[w \in \psi \wedge \psi \neq \phi]\} \tag{1}
\end{equation*}
$$

(2) a. John only kissed [Sue and Mary $]_{F}$.
b. John kissed Sue and Mary.
c. John kissed Sue.

Even if the context predicate $C$ is set to the focus alternative value, as Rooth proposed, alternatives like in (2c) are excluded which leads, in combination with what has sometimes been called the sentence's presupposition ${ }^{1}(2 b)$, to a contradiction.

Krifka (1993) managed to avoid this problem by introducing a solution (3) which is sensitive to subset relations among the alternatives.

[^15](3)
\[

$$
\begin{aligned}
& \text { only }\langle B, F\rangle \\
= & \left\{w \in B(F) \mid \forall F^{\prime} \in \operatorname{ALT}(F)\left[w \in B\left(F^{\prime}\right) \rightarrow B(F) \subseteq B\left(F^{\prime}\right)\right]\right\}
\end{aligned}
$$
\]

The proposal makes use of the theory of Structured Meanings, cf. Krifka (1992), going back to work by von Stechow (1982) and Jacobs (1983). The problem in (2) is solved because, now, propositions which are entailed by (2b) and which are therefore less informative, like (2c), are not excluded anymore.

The following tree (5), a derivation of the sentence in (4), briefly recalls the system of compositional focus semantics according to Krifka (1992). Applying an $F$ feature to some constituent (here the object DP) yields a background-focus structure $\langle B, F\rangle$, which ensures the identification of and the access to the focus even after the focused constituent has undergone semantic composition. It is to such representations that operators like the one defined in (3) apply.
(4) John ate $[\text { an APPLE }]_{F}$.


In the following section I will introduce the reader to several aspects of the notion "scale" as it appears at various subareas of the semantics of focus and only. This is necessary in order to solve a number of known problems with the approach introduced so far.

## 2 On the Nature of Scales

## Scalar Implicatures

Rooth (1992) claims that a sentence with a free (unbound) focus is able to trigger a Gricean quantity implicature. Assuming a universe of two individuals and their sum we can establish a partial order as in (6). By uttering the sentence (7a), not only its semantic content (7b) is asserted but it is furthermore implied that the statement was the strongest the speaker was able to make in comparison to alternative statements obtainable by means of replacing the focused constituent with other elements from the partially ordered set in (6). In particular, we may conclude (7c) and thus derive (7d).
(6)

a. $\mathrm{CARL}_{F}$ passed.
b. Carl passed. (from (a), assertion)
c. It is not true that Carl $\oplus$ Fred passed. (from (a), scalar implicature)
d. Fred didn't pass. (from (b),(c))

## Several Readings of Only?

Another role played by scales directly concerns the meaning of only. Various authors, e.g. Bonomi and Casalegno (1993) or Krifka (1993), have pointed out that only is ambiguous between a "quantificational" and a "scalar" reading. Consider the following case (8) of focus on an indefinite, which has the two readings exemplified by (9a) and (9b), whereas the latter is, in addition, underspecified with regard to how the focused phrase is compared to its alternatives.
(8) John only ate $[\text { an APPLE }]_{F}$.
(9) a. There was an apple $x$ which John ate, and John didn't eat anything but $x$. (quantificational)
b. What John ate was an apple and nothing more substantial/nutritious/expensive/healthy/toxic... (scalar)

Krifka (1993) proposes to analyse the former reading differently from cases with focus on proper names as seen above in (2). He suggests an analysis in which the indefinite takes scope over the only operator, leaving behind a focused trace. This, however, would mean that the analysis given before in (3) is far less general than previously thought.

Furthermore, the "scalar" reading (9b) doesn't get formally specified in Krifka (1993) at all and has received much less attention in the literature than quantificational only. It is therefore certainly not only $m y$ intuition that what should be pursued is the establishment of a unified semantics for only which on the one hand gives the right results when applied to a broad range of categories and on the other hand derives both the quantificational and the various scalar interpretations. For that reason, I would like to elaborate on the following two hypotheses.

Hypothesis 1. Only is always scalar (in a sense to be specified).
Hypothesis 2. The different readings of only-sentences are due to different scales associated with the focused elements.

## Scale for the "Quantificational Reading" (All-properties Scale)

In order to express the "quantificational" reading of only using a definition based on Krifka (1993) it might be helpful to take into account that an individual may be represented as the set of all its properties. When comparing different individuals, it is normally impossible to establish an order on these unless we decide beforehand what the ordering is supposed to express. However, if sums of individuals are taken into account as well, a partial order becomes salient naturally, viz. an order as in (10), which is of course related to the one in (6) above.

Think of a universe consisting of three individuals: the president ( $\mathbf{p}$ ), the vice-president ( $\mathbf{v}$ ) and the secretary of state ( $\mathbf{s}$ ) and their sums. They are represented in terms of generalised quantifiers (sets of properties) as the nodes of the graph. Here, 'the president' et al. are being treated as names. An arrow (' $\rightarrow$ ') should be read as ' $\supseteq$ '. Note, furthermore, that the null element - if we want to include it at all - does not correspond to the quantifier 'nobody' as one might perhaps assume but rather to the set $\mathcal{P}$ of all properties.


The graph expresses statements like the following, that the set of properties which the president, the vice-president and the secretary of state have in common is smaller or equal than the set of properties that the president and the vice-president have in common, and that set is smaller than the set of properties the president has. If only, in the spirit of (3), is to operate on such a graph it will keep the quantifiers which are ranked lower than the one in focus but throw out those ranked higher.

## Scale for the "Scalar Reading"(One-dimensional Scale)

There is, however, a possibility to compare individuals in a more direct way. This is what corresponds to the so-called "scalar interpretation" of only-sentences. In order to perform such a comparison, some contextual or otherwise salient information is necessary, which indicates the aspects according to which the different individuals are to be compared. For example, we might take a set $\mathcal{A}$ denoting authorities or powers. Then the subset $\lambda P . P(\mathbf{p}) \cap \mathcal{A}$ will represent the set of properties of the president that pertain to his authorities, for instance, the right to appoint ministers. I will write such an intersection as $[\lambda P . P(\mathbf{p})]^{\mathcal{A}}$. If we assume that there is always a way in which individuals can be ranked in terms of subsets of certain qualities, which they have (or don't have), and if the set $\mathcal{A}$ therefore is chosen in the appropriate way, then an ordering arises as in (11). ${ }^{2}$

$$
\begin{equation*}
[\lambda P . P(\mathbf{s})]^{\mathcal{A}} \subseteq[\lambda P . P(\mathbf{v})]^{\mathcal{A}} \subseteq[\lambda P . P(\mathbf{p})]^{\mathcal{A}} \tag{11}
\end{equation*}
$$

The obtained scale is a one-dimensional, a total order. It expresses that the set of relevant properties (e.g. authorities) that the secretary of state has is contained in that of the vice-president which is in turn contained in that of the president. ${ }^{3}$

[^16]What we are doing here is to make a selection from the domain of properties．This selection emphasises a certain aspect with regard to which individuals are compared．The so－called＂quantitative reading＂of only－sentences is nothing more than the generalisation of that comparison to all properties that make up the individuals．This will then lead back to the situation in which two individuals do not stand in the $\subseteq$ relation as could be read off the figure in（10）where it held，for instance，that $\lambda P . P(\mathbf{v}) \nsubseteq \lambda P . P(\mathbf{p})$ ．

What remains to be done is to spell out the uniform reading of only in terms of a context parameter $\mathcal{C}$ ，which accounts for the limitation towards a certain aspect of an individual．In this paper，I shall only concentrate on focus on nominals．

```
        only(\langleB,F\rangle,\mathcal{C})
= {w\inB(F)|\forall\mp@subsup{F}{}{\prime}\in\operatorname{Alt}(F)[w\inB(F')->(F\cap\mathcal{C})\subseteq(\mp@subsup{F}{}{\prime}\cap\mathcal{C})]}
```

When applying this definition to example（4）we obtain the following result：

```
{w\in\lambdaw.(\existsx[\operatorname{apple}(x)\wedge ate(\mathbf{j}x)](w))|\mathcal{Q}\in Alr\llbracketan apple\rrbracket
[w\in(\llbracketJohn\rrbracket(\llbracketate\rrbracket(\mathcal{Q})))->(\llbracketan apple\rrbracket\cap\mathcal{C})\subseteq(\mathcal{Q}\cap\mathcal{C})\rrbracket}
```

If we instantiate for $\mathcal{C}$ either the set $\mathcal{P}$ of all properties or the set $\mathcal{H}$ of all qualities concerning the healthiness of food we arrive at the two readings in（14）．

$$
\begin{align*}
\text { a. } & \{w \in \lambda w .(\exists x[\operatorname{apple}(x) \wedge \text { ate }(\mathbf{j} x)](w)) \mid \forall \mathcal{Q} \in \text { Alt } \llbracket \text { an apple } \rrbracket  \tag{14}\\
& {[w \in(\llbracket j o h n \rrbracket(\llbracket a t e \rrbracket(\mathcal{Q}))) \rightarrow \llbracket a n \text { apple】 } \subseteq \mathcal{Q}]\} } \\
\text { b. } & \{w \in \lambda w .(\exists x[\text { apple }(x) \wedge \text { ate }(\mathbf{j} x)](w)) \mid \forall \mathcal{Q} \in \text { Alt } \llbracket \text { an apple } \rrbracket \\
& {[w \in(\llbracket j o h n \rrbracket(\llbracket a t e \rrbracket(\mathcal{Q}))) \rightarrow(\llbracket \text { an apple } \cap \mathcal{H}) \subseteq(\mathcal{Q} \cap \mathcal{H})] }
\end{align*}
$$

Disappointingly，neither of them gives us a correct result．The meaning in（14a）should be compatible with（15）．
（15）John ate a green apple．
However，as the latter is not entailed by John＇s eating an apple（or rather 【an apple】 is not a subset of $\llbracket a$ green apple $\rrbracket$ ），it will be excluded and likewise for all other colors；which results in the absurd claim that John＇s apple seems to have had no color at all．

The reading（14b）is compatible with all quantifiers denoting things which which are healthier than an apple，e．g．kiwis．${ }^{4}$ This，however，is what should have been excluded by the statement that John only ate an apple．
democracies feature a head of state equipped with less authorities than e．g．their head of government．In those cases the selection of properties establishing the hierarchy between them can not always be figured out as easily．
${ }^{4}$ Nutritionists may forgive me if I am mistaken here．

## 3 A Solution in Terms of Background Alternatives

There is a systematic reason why the above readings are bad. The authors van Rooij and Schulz (2005) point out that approaches quantifying over focus alternatives run into trouble if the focused constituent is an indefinite or a disjunction.
a. John only kissed [Jane or Mary] ${ }_{F}$.
b. John kissed Jane or Mary.
c. John kissed Jane.
d. John kissed Mary.

Sentence (16a) results in the exclusion of both (16c) and (16d), which, in combination with (16b), yield a contradiction. This is essentially the same problem as with (15) above. To overcome this serious shortcoming, van Rooij and Schulz (2005) (quoting von Stechow (1991) on an idea by Groenendijk and Stokhof (1984)), propose an account in terms of so-called background alternatives. The formulation in (17) is an adaptation of one of their definitions. As a qualification, the approach is only supposed to apply to upward monotonic quantifiers ${ }^{5}$ whose alternatives are likewise upward monotonic. This is in line with assumptions made in von Stechow and Zimmermann (1984) and von Stechow (1991).

$$
\begin{align*}
& \text { only }\langle B, F\rangle \\
= & \{w \in W \mid(B(w))(F(w)) \wedge \neg \exists v \in W[(B(v))(F(v)) \wedge B(v) \subset B(w)]\} \tag{17}
\end{align*}
$$

Applied to our example, this yields (18).

$$
\begin{align*}
& \text { only }\langle\lambda \mathcal{Q} . \llbracket J o h n \rrbracket(\llbracket \text { ate } \rrbracket(\mathcal{Q})), \llbracket \text { an apple } \rrbracket \\
= & \left\{w \in W \mid \llbracket J o h n \text { ate an apple} \rrbracket^{w} \wedge\right.  \tag{18}\\
& \left.\neg \exists v\left[\llbracket J o h n \text { ate an apple } \rrbracket^{v} \wedge \lambda y \cdot[\operatorname{ate}(\mathbf{j} y)(v)] \subset \lambda y \cdot[\operatorname{ate}(\mathbf{j} y)(w)]\right]\right\}
\end{align*}
$$

What is going on here? The approach quantifies over possible worlds and only allows those worlds $w$ to get added to the meaning of the only-sentence if the matrix clause $\llbracket J o h n$ ate an apple』 holds in them and the extension of the background predicate "being eaten by John" in $w$ is minimal among all worlds in which the matrix clause holds. For (18) this will mean that the meaning of the sentence consists of only those worlds in which John ate an apple and not more than that. This approach also solves the disjunction problem from example (16a), cf. (van Rooij and Schulz (2005), albeit maybe not as nicely as one would have hoped. ${ }^{6}$

[^17]The problem with (18) for us is now that we are once more talking about extensions of predicates, i.e. sets of individuals, but as we saw in section (2), in order to describe both readings of only we need to be able to talk about quantifier meanings and sets of them. So we consider an equivalent representation to the one in (18), namely (19), which we obtain when we type-raise the background predicate to denote a function from quantifiers to truth values.

```
    only \(\langle\lambda \mathcal{Q} . \llbracket J o h n \rrbracket(\llbracket a t e \rrbracket(\mathcal{Q})), \llbracket a n\) apple \(\rrbracket\rangle\)
\(=\left\{w \in W \mid \llbracket J o h n\right.\) ate an apple \(\rrbracket^{w} \wedge\)
    \(\neg \exists v\left[\llbracket J o h n\right.\) ate an apple \(\rrbracket^{v} \wedge\)
    \(\lambda \mathcal{Q} \cdot[\mathcal{Q}(\lambda y \cdot \operatorname{ate}(\mathbf{j} y))(v)] \subset \lambda \mathcal{Q} \cdot[\mathcal{Q}(\lambda y \cdot \operatorname{ate}(\mathbf{j} y))(w)]]\}\)
```

While the ordering relation in (18) involved extensions containing individuals, e.g. $B\left(w_{1}\right)=$ $\{a\} ; B\left(w_{2}\right)=\{a, b\} \models B\left(w_{1}\right) \subset B\left(w_{2}\right)$, we are now dealing with extensions of predicates of type $\langle\langle\langle e, t\rangle, t\rangle, t\rangle$, like in (20).

$$
\begin{align*}
& B^{\prime}\left(w_{1}\right)=\{\llbracket \text { an apple } \rrbracket, \llbracket \text { something } \rrbracket, \ldots\}  \tag{20}\\
& B^{\prime}\left(w_{2}\right)=\{\llbracket \text { an apple and a kiwi } \rrbracket \text {, } \text { an apple } \rrbracket, \llbracket \text { a kiwi } \rrbracket, \\
&\llbracket \text { something } \rrbracket, \llbracket \text { at least two things } \rrbracket, \ldots\} \\
&=\quad B^{\prime}\left(w_{1}\right) \subset B^{\prime}\left(w_{2}\right)
\end{align*}
$$

It doesn't matter what the exact extensions are, what is important is the fact that the subset relations are preserved. The proof for this runs as follows: Assume two predicates $A, B$ of type $\langle e, t\rangle$ and their type-raised counterparts $A^{\prime}, B^{\prime}$ of type $\langle\langle\langle e, t\rangle, t\rangle, t\rangle$, functions from upward monotonic quantifiers to truth values. We want to show that $A \subset B$ iff $A^{\prime} \subset B^{\prime}$.

First assume $A \subset B$. We define $A^{\prime}:=\lambda \mathcal{Q} \cdot \mathcal{Q}(A), B^{\prime}:=\lambda \mathcal{Q} \cdot \mathcal{Q}(B)$ and assume a quantifier $\mathcal{Q}_{1} \in A^{\prime}$. It holds that $[\lambda \mathcal{Q} \cdot \mathcal{Q}(A)]\left(\mathcal{Q}_{1}\right)$ and therefore $\mathcal{Q}_{1}(A)$. By monotonicity and our initial assumption it follows that $\mathcal{Q}_{1}(B)$ from which we get, by $\lambda$-abstraction, $[\lambda \mathcal{Q} . \mathcal{Q}(B)]\left(\mathcal{Q}_{1}\right)$ or, equivalently, $\mathcal{Q}_{1} \in B^{\prime}$. We have, therefore, shown that $A \subset B \models A^{\prime} \subset$ $B^{\prime}$.

In the reverse direction, we assume $A^{\prime} \subset B^{\prime}$ and $x \in A$. The quantifier $\mathcal{Q}_{2}=\lambda P . P(x)$ thus holds for $A$, i.e. $\mathcal{Q}_{2}(A)$. By $\lambda$-abstraction this is equivalent to $(\lambda \mathcal{Q} \cdot \mathcal{Q}(A))\left(\mathcal{Q}_{2}\right)$ or

[^18]$A^{\prime}\left(\mathcal{Q}_{2}\right)$. From our initial assumption we get $B^{\prime}\left(\mathcal{Q}_{2}\right)$, therefore $\mathcal{Q}_{2}(B)$ and thus $x \in B$. We have proved that $A^{\prime} \subset B^{\prime} \models A \subset B$.

After many detours we have reached a satisfactory stage concerning the formulation of what used to be the "quantificational" reading of only. But what about the "scalar" reading? In section (2), I argued for the introduction of a contextual variable that tells us whether to take the entire quantifier meanings of the focused constituent into account or to limit our view to certain classified properties contained in that quantifier. A similar move will be proposed below although the approach is still not as homogeneous as what one probably would like to achieve eventually. My final definition for only is (21).

```
        only \((\langle B, F\rangle, \mathcal{C})\)
\(=\left\{w \in W \mid(B(w))(F(w)) \wedge \neg \exists v\left[\left(B(v)(F(v)) \wedge[B(v)]^{\mathcal{C}} \subset[B(w)]^{\mathcal{C}}\right]\right\}\right.\)
```

$B$ applied to some world $w$ is again a set of quantifiers. But it may become subject to some modifications. First, we define the quantifier intersection of $B(w)$ with a context variable $\mathcal{C}$ of the type of a set of predicates.

$$
[B(v)]^{\mathcal{C}}:=\left\{\begin{align*}
\text { if } \mathcal{C}=\mathcal{P}: & \left\{\mathcal{Q}^{\mathcal{C}} \mid \mathcal{Q} \in B(v)\right\}=\{\mathcal{Q} \cap \mathcal{C} \mid \mathcal{Q} \in B(v)\}  \tag{22}\\
\text { otherwise }: & \bigcup\left\{\mathcal{Q}^{\mathcal{C}} \mid \mathcal{Q} \in B(v)\right\} \\
& =\left\{\left(\mathcal{Q}_{1} \cap \mathcal{C}\right) \cup \cdots \cup\left(\mathcal{Q}_{n} \cap \mathcal{C}\right) \mid \mathcal{Q}_{i ; 1 \leq i \leq n} \in B(v)\right\}
\end{align*}\right.
$$

If $\mathcal{C}$ is the set of all properties we will receive the meaning in (19). However, for $\mathcal{C}=\mathcal{H}$, a set of properties denoting certain qualities we receive for $[B(v)]^{\mathcal{H}}$ the union of the properties contained in the quantifiers intersected with $\mathcal{H}$. This should give us one of the "scalar" interpretations for "John only ate an APPLE $F_{F}$ ", namely (23).

$$
\begin{align*}
& \text { only }\langle\lambda \mathcal{Q} . \llbracket \text { John } \rrbracket(\llbracket \text { ate } \rrbracket(\mathcal{Q})), \llbracket \text { an apple } \rrbracket\rangle \\
= & \left\{w \in W \mid \llbracket J o h n \text { ate an apple } \rrbracket^{w} \wedge\right.  \tag{23}\\
& \neg \exists v\left[\llbracket J o h n \text { ate an apple } \rrbracket^{v} \wedge \mathbf{Y}_{v} \subset \mathbf{Y}_{w} \rrbracket\right\}
\end{align*}
$$

Here, $\mathbf{Y}_{w}$ stands for the set $\bigcup\{\mathcal{Q} \cap \mathcal{H} \mid \mathcal{Q} \in \lambda \mathcal{Q} .[\mathcal{Q}(\lambda y \cdot$ ate $(\mathbf{j} y))(w)]\}$, i.e. the union of all the property-denoting sets obtained by intersecting the quantifiers in the denotation of the background in world $w$ with the set $\mathcal{H}$ of health properties.

The following assumptions shall be made: we assume again that John ate an apple in $w_{1}$, an apple and a kiwi in $w_{2}$, as well as, an apple and a peanut in $w_{3}$. All sets $B\left(w_{i}\right)$ for $1 \leq i \leq 3$ will contain at least the objects $\llbracket$ something $\rrbracket^{\mathcal{H}}$ and $\llbracket a n$ apple $\rrbracket^{\mathcal{H}}$. $B\left(w_{2}\right)$ will additionally contain at least $\llbracket a$ kiwi $\rrbracket^{\mathcal{H}}$ and $\llbracket$ an apple and a kiwi $\rrbracket^{\mathcal{H}}$. Furthermore, $B\left(w_{3}\right)$ will contain at least $\llbracket a$ peanut $\rrbracket^{\mathcal{H}}$ and $\llbracket$ an apple and a peanut $\rrbracket^{\mathcal{H}}$. The set $\mathcal{H}$ will impose the scale in (24) on the quantifiers.

$$
\begin{equation*}
\llbracket a \text { peanut } \rrbracket^{\mathcal{H}} \subseteq \llbracket a n \text { apple } \rrbracket^{\mathcal{H}} \subseteq \llbracket a \text { kiwi } \subseteq \rrbracket^{\mathcal{H}} \tag{24}
\end{equation*}
$$

If we now take the unions of the elements in $B\left(w_{i}\right)$ as prescribed by (21) and (22) we obtain the following results. If there is at least one health property $P$ which is an element
of $\llbracket$ a kiwi $\rrbracket^{\mathcal{H}}$ but not of $\llbracket$ an apple $\rrbracket^{\mathcal{H}}$ we get that $\mathbf{Y}_{w_{1}} \subset \mathbf{Y}_{w_{2}}$; in other words, world $w_{2}$ is going to be excluded.

On the other hand, as for every health property $P^{\prime} \in \llbracket a$ peanut $\rrbracket^{\mathcal{H}}$ it also holds that $P^{\prime} \in \llbracket a n ~ a p p l e \rrbracket^{\mathcal{H}}$, we get $\mathbf{Y}_{w_{1}}=\mathbf{Y}_{w_{3}}$ which includes world $w_{3}$.

Our reading (23) would thus be compatible with (25a) but not (25b) which is the desired outcome for the scalar interpretation of "John only ate an apple".
a. John ate a peanut.
b. John ate a kiwi.

## 4 Summary

I presented an approach for a unified meaning definition for only accounting for both the "quantificational" and the "scalar" reading. In order to do this certain deliberations were necessary concerning quantifiers and possibilities of how to rank them. If no further information is given, it is possible to rank quantifiers and their sums according to their mereological order. If a certain aspect is known according to which quantifiers shall be compared, i.e. a certain class of properties highlighting the desired mode of comparison, this can be spelled out in terms of intersections between the quantifiers. In both cases only operates on the available scales. In order to make things work and to avoid problems with disjunctions and indefinites a framework based on background alternatives, taken from von Stechow (1991) and van Rooij and Schulz (2005), is being used.

## Acknowledgements

This paper has profited from comments by Ágnes Bende-Farkas, Regine Eckardt, Hans Kamp, Doris Penka, Arnim von Stechow, three anonymous reviewers as well as audiences in Stuttgart and Moscow. I wish I would have been able to integrate more of their observations. Instead only a part of them must do. Thanks a lot!

## Bibliography

David Beaver. Five Only Pieces. Theoretical Linguistics, 30(1):45-64, 2004.
Andrea Bonomi and Paolo Casalegno. Only: Association with Focus in Event Semantics. Natural Language Semantics, 2(1):1-45, 1993.
Bart Geurts and Rob van der Sandt. Interpreting Focus. Theoretical Linguistics, 30(1):1-44, 2004.

Jeroen Groenendijk and Martin Stokhof. Studies in the Semantics of Questions and the Pragmatics of Answers. PhD thesis, Universiteit van Amsterdam, 1984.

Larry Horn. A Presuppositional Analysis of Only and Even. In Chicago Linguistics Society 5, pages 97-108, 1969.

Joachim Jacobs. Fokus und Skalen. Zur Syntax und Semantik von Gradpartikeln im Deutschen. Niemeyer, Tübingen, 1983.

Manfred Krifka. A Compositional Semantics for Multiple Focus Constructions. In Joachim Jacobs, editor, Informationsstruktur und Grammatik, pages 17-53. Westdeutscher Verlag, Opladen, 1992. Also in Proceedings of SALT 1. Cornell Working Papers in Linguistics 10. 1991.
Manfred Krifka. Focus and Presupposition in Dynamic Interpretation. Journal of Semantics, 10:269-300, 1993.
Fred Landman. Groups I+II. Linguistics and Philosophy, 12(5,6):559-605,723-744, 1989.

Craige Roberts. Only and Conventional Presupposition. ms., 2005
Robert van Rooij and Katrin Schulz. Only: Meaning and Implicatures - The Very Incomplete Short Version. In Emar Maier, Corien Bary, and Janneke Huitink, editors, Proceedings of SuB9, pages 314-324, 2005. Longer version to appear.
Mats Rooth. Association with Focus. PhD thesis, University of Massachusetts, Amherst, 1985.
Mats Rooth. A Theory of Focus Interpretation. Natural Language Semantics, 1(1): 75-116, 1992.

Arnim von Stechow. Structured Propositions. Technical report, Universität Konstanz, 1982. Arbeitspapiere des SFB 99.

Arnim von Stechow. Focusing and Backgrounding Operators. In Werner Abraham, editor, Discourse Particles, pages 37-84. Benjamins, Amsterdam, 1991.

Arnim von Stechow and Thomas Ede Zimmermann. Term Answers and Contextual Change. Linguistics, 22:3-40, 1984.

# Eager for Distinctness 

Raj Singh<br>Massachusetts Institute of Technology<br>singhr@mit.edu


#### Abstract

. I provide evidence that the interpretation of disjunctive sentences is sensitive to the linear order of the disjuncts. I argue that the asymmetry is rooted in informational redundancies that are present in one order that are not present in the other. I propose a constraint that ensures informational distinctness between earlier and later disjuncts. The constraint is checked at a surprising point in the left-right interpretation of the sentence, hence making the dynamics of the interpretive process of crucial importance in accounting for the observed asymmetry.


## 1 The Puzzle

(Hurford 1974) observes that disjunctions $\ulcorner X$ or $Y\urcorner$ where one disjunct entails the other are infelicitous. Following (Simons 2000), call such sentences "entailing disjuncts."
(1) \#John was born in Paris or in France
(2) \#John was born in France or in Paris

Let us call whatever it is that rules out entailing disjuncts "Hurford's Constraint" (henceforth HC). Observe that although (3) and (4) are entailing disjuncts, they are nonetheless judged felicitous:

Question: Who (of John and Mary) came to the party?
(3) (John or Mary) or Both [came to the party]
(4) John or (John and Mary) [came to the party]

Hurford uses the felicity of (3) and (4) along with HC to argue that English or is ambiguous between an inclusive and an exclusive reading. For example, if the first disjunct in (3) is read exclusively, then there is no longer any entailment between the disjuncts. As such, HC is avoided, and the sentence is judged felicitous.

Against the conclusion that English or is ambiguous, (Gazdar 1979) and (Simons 2000) observe that or is always read exclusively in the scope of negation:
(5) I didn't eat beef or pork at the party (= I didn't eat beef and I didn't eat pork)

If or had an exclusive meaning, (5) should be judged true if I ate both beef and pork at the party. Such a reading is unattested. Further, n-ary disjunctions $\left\ulcorner\mathrm{X}_{1}\right.$ or $X_{2}$ or $\ldots$ or $\left.\mathrm{X}_{n}\right\urcorner$ are normally interpreted as 'only $X_{1}$ or $\ldots$ or only $X_{n}$.' Let us call this "the onlyone reading." Simons shows that an exclusive or has no way of generating the only-one reading.

Simons offers a solution to these puzzles. She develops a system whereby or is unambiguously inclusive (truth-conditionally). She derives the exclusive reading by exhaustiyfing each disjunct. Along with a pragmatically motivated constraint against entailing disjuncts, her system solves all the issues raised above: the infelicity of (1) and (2), the felicity of (3) and (4), the only-one reading of n-ary disjunctions, and the fact that or is unambiguously inclusive in the scope of negation. However, since disjunction is fully symmetric in Simons' system, she predicts that a sentence $\ulcorner X$ or $Y\urcorner$ should be felicitous iff $\ulcorner Y$ or $X\urcorner$ is. As such, she predicts that (6) and (7), which are the same as (3) and (4) but for the order of the disjuncts, should be felicitous. This prediction is incorrect.
(6) \#(John and Mary) or (John or Mary) [came to the party]
(7) \#(John and Mary) or John [came to the party]

The puzzle is: if exhaustifying disjuncts allows us to avoid HC in (3) and (4), why doesn't it allow us to do so in (6) and (7)?

## 2 Symmetric and Asymmetric Disjunction

How would standard theories of disjunction deal with the facts stated above? For purposes of exposition, it is useful to group the various theories into four classes, though I must insist that the grouping need not be read as forming a partition of the set of theories. One class is composed of what I call "global pragmatic systems," of which the state of the art representatives are (Spector 2003; van Rooy and Schulz 2004; Sauerland 2004). These systems compute strengthened meanings of sentences "globally," i.e. at the root $S$ node. As such, the interpretation of sentences is insensitive to the gross syntactic form underlying it. ${ }^{1}$ The truth-conditional output at the root node of sentence $X, M(X)$, is fed to a pragmatic component whose job is to locate some subset of $M(X), S M(X)$, as the final interpretation of $X$. We call $S M(X)$ the "strengthened meaning of $X$." ${ }^{2}$

A second class of theories are what I call "list systems." I take (Simons 2000; Zimmerman 2000) as representative of this class of theories. The list system approach to

[^19]disjunction takes disjunctions to be lists of possibilities. A sentence $\left\ulcorner X_{1}\right.$ or $X_{2}$ or $\ldots$ or $\left.X_{n}\right\urcorner$ will be interpreted as a list $\mathcal{L}=\left\{S M\left(X_{1}\right), S M\left(X_{2}\right), \ldots, S M\left(X_{n}\right)\right\} .{ }^{3}$ The natural way of thinking of this list is as the set of answers to the question under discussion that are compatible with the speaker's epistemic state.

The third class of theories are what I call "syntactic approaches." (Chierchia 2004; Fox 2006) are the state of the art systems implementing this kind of approach. ${ }^{4}$ Syntactic approaches posit the existence of a null morpheme in the syntax, exh, that can decorate the syntactic tree at any $S$ node. This silent operator is used to strengthen meanings: by appending exh to any sentence $X$ in the tree (global or embedded), one generates $S M(X)$ as the interpretation of $X$.

What is crucial to note here is that each of the above systems is fully symmetric with respect to the disjuncts: a sentence $\ulcorner X$ or $Y\urcorner$ is predicted to be felicitous iff $\ulcorner Y$ or $X\urcorner$ is. In a fuller version of this paper, I argue that without radical revisions or ad hoc stipulations, with the exception of the list systems approach the above mentioned frameworks are illsuited for dealing with the asymmetries noted in (3), (4), (6) and (7). Space limitations prevent me from further discussion of this issue here. In this paper, I will simply propose a variant of the list systems approach that can account for the facts, and, unfortunately, leave for another occasion a more complete discussion of the nature of the mismatch between the facts of asymmetry and the theories briefly discussed above.

Before turning to my own proposal, I should like to mention the one theory that I am aware of that does posit an asymmetry between the disjuncts, namely, Lauri Karttunen's original dynamic proposal (Karttunen 1974). The theory developed there offers the following entry for disjunction: $\ulcorner X$ or $Y\urcorner$ means ' $X$ or $(Y$ and $\neg X)$.' Unfortunately, the entry doesn't quite work. For instance, consider the question Who of John and Mary came to the party?, and imagine its answer (John and Mary) or Sue. Letting " $j$, " " $m$ ", " $s$ " be the obvious abbreviations, Karttunen predicts that ( $j$ and $m$ ) or $s$ means '( j and m ) or ( s and $(\neg \mathrm{j}$ or $\neg \mathrm{m}))$. . As such, he predicts that the sentence could be judged true in a situation where John and Sue, but not Mary, came to the party. This is an incorrect prediction, for the only available reading of the sentence is that either only John and Mary came to the party or only Sue did. ${ }^{5}$ Thus, even by explicitly encoding what seems, prima facie, to be the right kind of asymmetry given the above facts, we are still unable to arrive at the correct interpretations of disjunctive sentences.

[^20]
## 3 Analysis: Taking Dynamics Seriously

I believe that, despite the shortcomings of Karttunen's proposal, dynamic systems of interpretation provide the most natural framework for dealing with the facts of asymmetry. In this section, I will state my own dynamic proposal. I will attempt to keep the discussion as theory neutral as possible so as to convey the force of the idea itself. In Section 4, I will provide specific details of my particular way of implementing the ideas discussed here. This will necessarily involve additional assumptions I make about the dynamics of information flow in communication. Here I state, informally, the conceptual foundations of the theory, its content, and its intended range of application.

The main assumptions of the theory I wish to propose are:
Questions in Contexts Following (Collingwood 1940; Groenendijk and Stokhof 1997; Rescher 2000; Spector 2003; van Rooy and Schulz 2004), inter alia, I assume that any actual discourse always contains some (possibly implicit) question $Q$. In context $c$, I assume that $Q$ partitions $c$ into a set $c_{Q}=\left\{c_{1}, \ldots, c_{r}\right\} .{ }^{6}$

Local Strengthening Given the wealth of facts discussed in (Chierchia 2004; Fox 2006), and some of the facts alluded to above, I assume that meanings are strengthened locally at each disjunct.

Disjunctions as Lists I follow (Simons 2000; Zimmerman 2000) in assuming that disjunctions provide lists of (strengthened) answers to questions. More specifically, in answering a question $Q$ in context $c$, the speaker has to provide a list $\mathcal{L}$ of possible answers to $Q$. Each disjunct provides an answer $\mathcal{D}\rangle \in \wp\left(c_{Q}\right)$, the power set of $c_{q}$. The list $\mathcal{L}=\bigcup \mathcal{D}\rangle .{ }^{7}$

Left-Right Asymmetry I assume that interpretation occurs in time. In partiular, I assume that propositions get added to the list following the L-R order of the disjuncts. As such, the construction of $\mathcal{L}$ through time becomes relevant.

Given these assumptions, I formulate a single constraint on the dynamics of list construction:

Constraint Enforcing Informational Distinctness In the L-R interpretation of a disjunction, information that has already been added to the list cannot be brought up as a candidate for list membership at later stages of interpretation. More specifically, if $\mathcal{L}$ is the list that's been constructed up to the current stage of interpretation, and the current disjunct provides the answer $\mathcal{D}, \mathcal{D} \in \wp\left(c_{Q}\right)$, then if $\mathcal{L} \cap \mathcal{D}=\emptyset$, one may continue constructing the list by combining the members of $\mathcal{L}$ and $\mathcal{D}, \mathcal{L} \cup \mathcal{D}$. If

[^21]$\mathcal{L} \cap \mathcal{D} \neq \emptyset$ the sentence will be judged infelicitous. The constraint is checked at each disjunct in the L-R order in which the disjuncts appear.

I will now show that the resulting system derives HC (1),(2), its obviation in (3),(4), the inability to avoid HC in $(6),(7)$, and the only-one reading of n -ary disjunctions, along with other new facts to be discussed in this section. As such, I claim that it is not quite entailing disjuncts that are the problem (cf. (Hurford 1974; Gazdar 1979; Simons 2000)), but rather, informational overlap between earlier and later stages of interpretation. I will make all of these ideas precise shortly, but let us first examine, informally, how the system is supposed to work.

Consider first the sentence in (1): John was born in Paris or in France. Imagine this is a response to the question Where was John born? We can assume without loss of generality that this is a "city-level" question, i.e. that the question is asking for which city it is that John was born in. ${ }^{8}$ Imagine there are only three cities in each world in the common ground, Paris, Nancy, and Montreal. In such a context, then, $c_{Q}=\{[\mathrm{m}],[\mathrm{n}]$, $[\mathrm{p}]\}$, where $[x]=\{\mathrm{w}$ in $c$ : John was born in $x$ in w$\}$. In interpreting this sentence, the list $\mathcal{L}$ is initially empty. The interpreter begins with the leftmost disjunct, and adds the information in each successive disjunct to the list. In this case, it begins by adding the information that John was born in Paris. Thus, at this stage, $\mathcal{L}=\{[\mathrm{p}]\}$. The next disjunct gives the information that John was born in France, i.e. $\mathcal{D}=\{[\mathrm{n}],[\mathrm{p}]\}$. Since $\mathcal{D} \cap \mathcal{L} \neq \emptyset$, the constraint enforcing distinctness of information between earlier and later disjuncts is violated. Hence the infelicity. Note that the informational overlap in this case is invariant under order permutation, and so sentence (2), John was born in France or in Paris, will be ruled out for the same reason as (1).

Let us now consider the contrast between (4) and (7). We abbreviate (4) and (7) as $j$ or ( $j$ and $m$ ) and ( $j$ and $m$ ) or $j$, respectively. Imagine these sentences uttered as answers to the question, Who of John and Mary came to the party? This question in context $c$ results in partition $c_{Q}=\{[\mathrm{j}, \mathrm{m}],[\mathrm{j}],[\mathrm{m}],[]\}$. This is a set of propositions where both John and Mary came to the party, only John came to the party, only Mary came to the party, and neither John nor Mary came to the party. ${ }^{9}$

Interpretation of (4) will proceed as follows. The strengthened meaning of the first disjunct, $j{ }^{10}$ will be $\mathcal{D}=\{[\mathrm{j}]\}$. Since $\mathcal{L}$ is empty at this point, we add $[\mathrm{j}]$ to $\mathcal{L}$, so that $\mathcal{L}=\{[\mathrm{j}]\}$. The information in the second disjunct, $j$ and $m$, is a set of worlds $[\mathrm{j}, \mathrm{m}]$ where both John and Mary came to the party. Since $\mathcal{L} \cap\{[j, m]\}=\emptyset$, the constraint against informational overlap is satisfied, and the sentence is accepted as felicitous.

In the other direction, at the point at which the interpreter is done with the first disjunct $j$ and $m, \mathcal{L}=\{[\mathrm{j}, \mathrm{m}]\}$. Next up is the sentence $j$. Observe that, truth-conditionally, $j$ means $\{[\mathrm{j}],[\mathrm{j}, \mathrm{m}]\}$. The strengthened meaning of $j, S M(\mathrm{j})=\{[\mathrm{j}]\}$. This raises a question that has been lurking in the background of our discussion so far: does the interpreter check the constraint against informational overlap before the meaning of the disjunct is strengthened

[^22]or after strengthening has already taken place? If the former, then then the observed infelicity will be predicted. If the latter, then we will still require some explanation as to why the sentence is judged inappropriate. Given the observed infelicity of this sentence, along with other facts to be discussed shortly, it seems that natural language opts to check the constraint before strengthening has a chance to apply. Note that this is a choice point that could have gone either way. I currently have no explanation for why the choice should have gone this way rather than the other. It is in fact this element of timing that forces us to take the L-R order of the disjuncts seriously. Thus, not only must distinctness be enforced, it seems it must be enforced at a particular point in the temporal evolution of the sentence's interpretation:

Timing Principle Ensure that the truth-conditional meaning (as a set of propositions) of the current disjunct has zero intersection with $\mathcal{L}$.

This suggests a certain sort of "eagerness" on the part of the constraint against overlap, for it seems to apply as soon as it can, not allowing strengthening to have a chance to potentially save the utterance. We may be justified in asking: why so eager? Why not allow for strengthening to take place before checking for distinctness? Whatever the answer to this question, the constraint really is one about local checking of distinctness, and not about global properties of the disjunction:
\#(((John and Mary) or John) or Sue) [came to the party]

Interestingly, no amount of contrastive focus can override the constraint:
(9) \#((John and Mary) or $\left.[J o h n]_{F}\right)$ or ((John and Mary) or [Sue] ${ }_{F}$ ) [came to the party]

Note, additionally, that we predict that adding an only to the second disjunct of (7) should save it from the constraint:
(10) Either John and Mary came to the party or only John did [come to the party]

This is because only $j$ means, truth-conditionally, that John came and no one else did. Since $\{[\mathrm{j}]\}$ has zero overlap with $\mathcal{L}=\{[\mathrm{j}, \mathrm{m}]\}$, the sentence is (correctly) predicted to be felicitous.

Note that we not only derive HC as a subcase of informational overlap, we can actually show that HC is not in and of itself a correct generalization of the facts. Observe that (11) is infelicitous, despite the fact that neither disjunct entails the other:
(11) \#John ate some of the cookies or not all of them
(Hurford 1974) and (Gazdar 1979) both predict that there should be at least one reading of (11) under which it becomes acceptable. Since strengthening is optional, by not strengthening the first disjunct to mean 'some but not all,' we can ensure that there is no entailment between the disjuncts. As such, Hurford and Gazdar both predict that (11)
should be felicitous. However, under our analysis, whether you strengthen or not, there will be informational overlap. For imagine that we have a partition fine-grained enough to distinguish between worlds where John ate some but not all of the cookies and those where he ate all of them, so that $c_{Q}=\{[\mathrm{a}],[$ sbna $],[\mathrm{n}]\} .{ }^{11}$ Now, observe that if you do strengthen the first disjunct, $\mathcal{L}=\{[$ sbna $]\}$. The meaning of the second disjunct yields $\mathcal{D}=\{[$ sbna $]$, $[\mathrm{n}]\}$. Thus, there is overlap between $\mathcal{L}$ and $\mathcal{D}$, and we thereby incur a violation of the constraint on distinctness. If you don't strengthen the first disjunct, then $\mathcal{L}=\{[$ sbna], [a]\}, which again results in overlap with $\mathcal{D}=\{[s b n a],[\mathrm{n}]\}$. Thus, under any way of interpreting the first disjunct, we predict the sentence to be infelicitous. This is the correct prediction. There is no way for this sentence to escape the constraint.

The analysis also connects with the theory of presupposition. Consider, for instance, the contrast between (12) and (13):
(12) Either there is no King of France or the King of France is in Paris
\#Either there is a King of France or the King of France is in Paris
Imagine these sentences as answers to the question where is the King of France? raised in a context in which it is not known whether or not there is a King of France. ${ }^{12}$ Suppose that there are only two cities in France in each world in the common ground, Paris and Nancy. Then, in such a context, this "problematic question" will result in partition $c_{Q}$ $=\{[-\mathrm{kf}],[\mathrm{p}],[\mathrm{n}]\}$, a set of propositions where either there is no King of France, or there is and he's in Paris, or there is and he's in Nancy. In answer (12), interpretation of the first disjunct results in $\mathcal{L}=\{[-\mathrm{kf}]\}$. The content of the second disjunct yields the set $\mathcal{D}=$ $\{[\mathrm{p}]\}$. Since $\mathcal{L} \cap \mathcal{D}=\emptyset$, the constraint is satisfied, and (12) is judged felicitous. In (13), on the other hand, the information conveyed by the first disjunct creates $\mathcal{L}=\{[\mathrm{n}]$, $[\mathrm{p}]\}$. The information in the second disjunct results in $\mathcal{D}=\{[p]\}$. Thus, $\mathcal{L}$ and $\mathcal{D}$ have $\{[p]\}$ as their intersection, and, because of this non-distinctness, the sentence is judged infelicitous.

Finally, we make a rather strong prediction about the interaction between meaning strengthening and presupposition. Imagine a sentence $\ulcorner X$ or $Y\urcorner$ where $X$ has $\neg p$ as a scalar implicature and $Y$ has $p$ as a presupposition. Imagine further that the disjunction is uttered in a context compatible with both $p$ and $\neg p$. Our constraint enforcing distinctness of information predicts that $\ulcorner X$ or $Y\urcorner$ should be felicitous, while $\ulcorner Y$ or $X\urcorner$ should be infelicitous. In interpreting $\ulcorner X$ or $Y\urcorner$, interpretation of $X$ (including strengthening) of $X$

[^23]will result in $\mathcal{L}=\{[X, \neg p]\}{ }^{13}$ With local accommodation of $p$ when interpreting $Y$, the truth-conditional output of the interpretation of $Y$ will be $\mathcal{D}=\{[Y, p]\}$. Thus, distinctness will be met, and $\ulcorner X$ or $Y\urcorner$ will escape unscathed. Flipping the order to $\ulcorner Y$ or $X\urcorner$, however, should result in infelicity. Interpretation of $Y$, with local accommodation of $p$, will result in $\mathcal{L}=\{[Y, p]\}$. The truth-conditional meaning of $X$ will result in $\mathcal{D}=\{[X, p],[X, \neg p]\}$. Then, if for whatever reason, the context licenses a further partitioning of $[X, p]$ into $[Y$, $p]$ and $[\neg Y, p]$, we will have the required infelicity. For in such a context, $\mathcal{L}=\{[Y, p]\}, \mathcal{D}$ $=\{[Y, p],[\neg Y, p],[X, \neg p]\}, \mathcal{L} \cap \mathcal{D}=\{[\mathrm{Y}, \mathrm{p}]\} \neq \emptyset$, and so we will have a violation of the constraint enforcing distincness. This is indeed what we find: ${ }^{14}$
(14) Either John ate some of the cookies or he regrets having eaten all of the cookies \#Either John regrets having eaten all of the cookies or he ate some of the cookies

## 4 The Formal System

I assume familiarity with the context-change theory of (Heim 1983). I will revise this model along two dimensions. First, I will enrich contexts which, in (Heim 1983), are sets of worlds $c$ (information taken for granted by speaker and hearer). In the system to be developed here, $c$ will be partitioned by a question $Q=$ ? $x P x$, resulting in a set of propositions $c_{Q}=\left\{c_{1}, \ldots, c_{r}\right\}$. I will add further structure to $c_{Q}$ by ordering the cells of the partition by a measure on the extension of the question predicate $P$ in the worlds in each cell, an idea inspired by proposals made in (van Benthem 1989; van Rooy and Schulz 2004). Second, I will modify the kinds of operations that can be performed on contexts by using variants of the "transformational" and "minimization" operators developed in (van Benthem 1989).

### 4.1 Enriching Contexts

Definition 1. A question $Q$ partitions $c$ into $c_{Q}=\left\{c_{1}, \ldots, c_{r}\right\}$ by inducing an equivalence relation $\mathcal{R}_{Q}$ on $c: \forall w, w^{\prime} \in c, w \mathcal{R}_{Q} w^{\prime}$ iff $[[\mathrm{P}]]^{w}=[[\mathrm{P}]]^{w^{\prime}}$.

Example 1. If Who of John and Mary came to the party? is raised in context $c$, then $c_{Q}$ $=\{[j, m],[j],[m],[]\}$, as seen earlier.

[^24]Definition 2. Let $c_{Q}=\left\{c_{1}, \ldots, c_{r}\right\}$. Then: $\forall c_{i}, c j \in c_{Q}, c_{i} \geq c_{j}$ iff for any $w_{i} \in c_{i}, w_{j} \in c_{j}$, $[[\mathrm{P}]]_{i}^{w} \cap\left[[\mathrm{P}] j_{j}^{w}=[[\mathrm{P}]]_{j}^{w}\right.$.

Example 2. Let $c_{Q}=\{[j, m],[j],[m],[]\}$ as before. Then $\geq=\{([j, m],[j]),([j, m],[m])$, ([j,m], []), ([j], []), ([m], [])\}. We abbreviate this by writing $[j, m] \geq[j],[m] \geq[]$.

### 4.2 Operations on Contexts

In this section, I will define the basic operations from which we will develop compositional definitions of context change potentials (CCPs) for atomic, conjunctive, and disjunctive sentences. These CCPs are designed to exploit the enriched structure of contexts induced by the ordered partition. Recall that CCPs in this system are (partial) functions from $c^{*}$ to a list $\mathcal{L} \subseteq c_{Q}$. They are constructed as combinations of two primitive operations. There is a transormational operator, $\tau$, indexed by sentences of natural language. This operator, $\tau_{\phi}$, takes $c^{*}$ as input and returns a set $\mathcal{D} \in \wp\left(c_{Q}\right)$. $\mathcal{D}$ is composed of those cells $c_{i}$ of $c_{Q}$ in which $\phi$ is true in all worlds in $c_{i}$. As such, $\tau_{\phi}$ recaptures the effect of the $+\phi$ function of (Heim 1983). One can think of $\mathcal{D}$ as a set of candidates for the list $\mathcal{L}$. Given this set of candidates, a conversational minimization operator, $\mu$, will select the minimal element (with respect to the order defined above) from this candidate set if such and such contextual conditions will be found to hold, otherwise it will select the entire candidate set. The output of minimization is then added to $\mathcal{L}$.

Definition 3. Let $k$ be an arbitrary set of worlds, $\phi$ an atomic sentence, and $p$ 's presupposition. ${ }^{15}$ Then $k$ admits $\phi$ just in case $p(w)=1$ for all $w$ in $k$.

Example 3. Suppose that all $w$ in $k$ are such that there is a King of France in $w$. Then $k$ admits The King of France is in Paris.

Now, let $c^{*}$ be a structured context, $\phi$ an atomic sentence, and $\tau_{\phi}$ the transformational operator $\tau$ indexed by $\phi$. Then:

Definition 4. Structured context $c^{*} \in \operatorname{dom}\left(\tau_{\phi}\right)$ iff there is a $c_{i}$ in $c^{*}$ such that $c_{i}$ admits $\phi$. In such a case, we say that $\tau_{\phi}$ is well-defined on $c^{*}$.

Example 4. Let $c^{*}=\left(c_{Q}, \geq\right)$, where $c_{Q}=\{[-k f],[n],[p]\}$ as in Section 3. Then $c^{*}$ admits The King of France is in Paris.

Definition 5. If $\tau_{\phi}$ is well-defined on $c^{*}$, then $\tau_{\phi}\left(c^{*}\right)=\left\{c_{i} \in c^{*}\right.$ : $c_{i}$ admits $\phi$, and $\ulcorner\phi\urcorner$ in $w$ for all $w$ in $\left.c_{i}\right\}$.

Example 5. Let $\phi=$ The King of France is in Paris, and let $c^{*}$ be defined as above. Then $\tau_{\phi}\left(c^{*}\right)=\{[p]\}$.

[^25]Example 6. Let $c^{*}=\left(c_{Q}, \geq\right)$, where $c_{Q}=\{[j, m],[j],[m],[]\}$, and consider $\phi=$ John came to the party. Then $\tau_{\phi}\left(c^{*}\right)=\tau_{j}\left(c^{*}\right)=\{[j, m],[j]\}$.

With the transformational system now set up, we can define our conversational minimization operator. We first give a general definition of minimization on information states, making use of technical developments found in (van Benthem 1989).

Definition 6. Let $K=\left\{k_{1}, \ldots, k_{r}\right\}$, where $k_{i}$ is a set of worlds, and let $\geq$ be a partial order on $K$. Then $\mu^{*}(K)=\left\{k_{i} \in K\right.$ : there is no $k_{j}$ in $K$ such that $\left.k_{i} \geq k_{j}\right\}$. Call $\mu^{*} a$ minimization operator.

Definition 7. Let $K$ be as above. Then the speaker is opinionated about $K$ if the speaker's believing the union of the propositions in $K$ implies that she believes only one proposition in $K$.

Definition 8. Let $K$ and $\mu^{*}$ be defined as above. Then our conversational minimization operator, $\mu$, operates on $K$ as follows:

$$
\mu(K)= \begin{cases}\mu^{*}(K) & \text { if it is common ground that the speaker is opinionated about } K \\ K & \text { otherwise }\end{cases}
$$

Example 7. Recall $c_{Q}=\{[j, m],[j],[m]$, []\}, with order $[j, m] \geq[j],[m] \geq[]$. Let $K=$ $\tau_{j}\left(c^{*}\right)=\{[j, m],[j]\}$. Then, if it is common ground that the speaker is opinionated about $K$, then $\mu(K)=\{[j]\}$. Otherwise, $\mu(K)=K=\{[j, m],[j]\}$.

For the remainder of this paper, we will assume that it is common ground that the speaker is opinionated at each point at which the choice comes up.

### 4.3 Revised Context Change Potentials

We associate with each sentence of the language $\phi$ a CCP $\pi_{\phi}$. CCPs are partial functions from structured contexts to lists.

Definition 9. Let $\phi$ be an atomic sentence uttered in structured context $c^{*}$. Then the execution of $\pi_{\phi}$ on $c^{*}$ is: $\pi_{\phi}\left(c^{*}\right):=\mu\left(\tau_{\phi}\left(c^{*}\right)\right)$.

Example 8. Imagine the question Who of John and Mary came to the party? is asked in context $c$. Then $c^{*}=\left(c_{Q}, \geq\right)$, where $c_{Q}=\{[j, m],[j],[m],[]\}$. and $[j, m] \geq[j]$, $[m] \geq[]$. Recall from just above that $\tau_{j}\left(c^{*}\right)=\{[j, m],[j]\}$. Then $\pi_{j}(c *)=\mu\left(\tau_{j}(c *)\right)=$ $\mu(\{[j, m],[j]\})=\{[j]\}$. Before any conversational reasoning begins, our list $\mathcal{L}$ will be empty. Thus, in the atomic case, we simply identify $\mathcal{L}$ with the output of $\pi_{\phi}\left(c^{*}\right)$, so that, in this example, $\mathcal{L}=\{[j]\}$. Once the list has been completely specified, we can simply take the union of all the propositions in $\mathcal{L}$ to return a new unstructured context, $c^{\prime} \subseteq c$, which will now be ready for a new question to come and partition it for another round of conversational reasoning. In this case, $c^{\prime}=\bigcup \mathcal{L}=[j]$.

Definition 10. Let $\phi=\left\ulcorner\alpha_{1}\right.$ and $\alpha_{2}$ and $\ldots$ and $\left.\alpha_{n}\right\urcorner$.
Then $\pi_{\phi}\left(c^{*}\right)=\mu\left(\tau_{\phi}\left(c^{*}\right)\right):=\mu\left(\left(\tau_{\alpha_{1}} ; \tau_{\alpha_{2}} ; \ldots ; \tau_{\alpha_{n}}\right)\left(c^{*}\right)\right)^{16}=\mu\left(\tau_{\alpha_{n}}\left(\tau_{\alpha_{n-1}}\left(\ldots\left(\tau_{\alpha_{1}}\left(c^{*}\right)\right)\right)\right)\right)$.
Example 9. Suppose the answer to the question were John and Mary came to the party, which we abbreviate as j and m . Then $\pi_{j \text { and } m}(c *)=\mu\left(\tau_{j \text { and } m}(c *)\right)=\mu\left(\left(\tau_{j} ; \tau_{m}\right)(c *)\right)=$ $\mu\left(\tau_{m}\left(\tau_{j}(c *)\right)\right)=\mu\left(\tau_{m}(\{[j, m],[j]\})\right)=\mu(\{[j, m]\})=\{[j, m]\}$.
Minimization occurs after all the transformational operations have applied. Since it is minimization that determines which propositions go into $\mathcal{L}$, we can, as in the atomic case, simply identify $\mathcal{L}$ with $\pi_{\phi}\left(c^{*}\right)$ when $\phi$ is a conjunctive sentence.

Recall from our informal discussion in Section 3 that in disjunctive sentences, strengthening occurs locally at each disjunct. Thus, the decision about what goes into $\mathcal{L}$ is made at each disjunct, and the list is expanded at each disjunct. The list created by a disjunctive sentence in structured context c* will, in general, be the union of the output of the execution of the CCPs of each disjunct on $c^{*}$. Thus, the recursion is on the CCPs of each disjunct. However, as opposed to the atomic and conjunctive cases, the list $\mathcal{L}$ changes as the interpretation of the disjunction proceeds. As such, the clearest way of representing the flow of information here is by writing out a high-level program incorporating both the CCP of disjunction and the constraint enforcing informational distinctness:

Definition 11. Let $\phi=\left\ulcorner\alpha_{1}\right.$ or $\alpha_{2}$ or $\ldots$ or $\left.\alpha_{n}\right\urcorner$.
Then the execution of $\pi_{\phi}\left(c^{*}\right)$ is represented in the following program:
Initialize: $\mathcal{L} \longleftarrow \emptyset$
for $i=1, \ldots, n$
if $\left(\mathcal{L} \cap \tau_{\alpha_{i}}(c *)\right)=\emptyset$
then $\mathcal{L} \longleftarrow\left(\mathcal{L} \cup \pi_{\alpha_{i}}(c)\right)$ else Output "\#" and Halt
end if
end for
Output $\mathcal{L}$

The choice point in the algorithm is at the minimization step; different selections by $\mu$ will result in different outputs. If all of $\mu$ 's choices result in output " $\#$," the sentence will be judged infelicitous.

### 4.4 Some Example Computations

We run through a few of the key examples discussed informally in Section 3. We focus on our running example, where the question is Who of John and Mary came to the party? All the other examples will fall out in the same way - there is no practical or theoretical difference to speak of. We will proceed by making our way through the for-loop. We will

[^26]call the result of the if-test Step 1. If the condition is satisfied, we will denote this with a YES; otherwise, we will say NO. Depending on the answer, we will either output "\#" or add the result of applying $\mu$ to the current list $\mathcal{L}$. Call the result of this second stage of the computation Step 2.

Consider first the response $j$ or $m$. Initially, $\mathcal{L}=\emptyset$. We begin with the first disjunct, $j$. Recall that $\tau_{j}\left(c^{*}\right)=\{[\mathrm{j}, \mathrm{m}],[\mathrm{j}]\}$. Thus, Step $1=$ YES. Since $\mu\left(\tau_{j}\left(c^{*}\right)\right)=\mu(\{[\mathrm{j}, \mathrm{m}],[\mathrm{j}]\})$ $=\{[\mathrm{j}]\}$, Step $2=\mathcal{L}=\{[\mathrm{j}]\}$. Next, we get to $m . \tau_{m}\left(c^{*}\right)=\{[\mathrm{j}, \mathrm{m}],[\mathrm{m}]\}$, which has zero intersection with $\mathcal{L}=\{[\mathrm{j}]\}$. Thus, Step $1=$ YES. Since $\mu\left(\tau_{m}\left(c^{*}\right)\right)=\mu(\{[\mathrm{j}, \mathrm{m}],[\mathrm{m}]\})=$ $\{[\mathrm{m}]\}$, Step $2=\mathcal{L}=\{[\mathrm{j}],[\mathrm{m}]\}$. We are at the end of the disjunction, and so we output $\mathcal{L}$. This is the correct result.

Now consider response (4), $j$ or ( $j$ and $m$ ). Begin with the first disjunct, and with an initially empty list. The result here will thus be the same as above, with Step $2=\mathcal{L}=$ $\{[\mathrm{j}]\}$. At the second disjunct, $j$ and $m, \tau_{j}$ and $m\left(c^{*}\right)=\{[\mathrm{j}, \mathrm{m}]\}$. Thus, Step $1=$ YES, and so Step $2=\mathcal{L}=\{[\mathrm{j}],[\mathrm{j}, \mathrm{m}]\}$. This will be the output of the algorithm, which is the correct result.

We now consider response (7), which is (4) in reverse order. At the first disjunct, $j$ and $m$, we have $\mathcal{L}=\{[\mathrm{j}, \mathrm{m}]\}$. At the second disjunct, $j, \tau_{j}(c *)=\{[\mathrm{j}, \mathrm{m}],[\mathrm{j}]\}$. Thus, Step $1=$ NO, and so Step $2=\#$. There is no other way to interpret this sentence, and so we have captured the observed infelicity.

## 5 Final Remarks

I would like to end with two short notes. First, the only-one reading follows immediately from the above system. The proof is by induction on the complexity of the disjunction. Second, if the approach developed here is on the right track, we have evidence that numerals should be given an exactly interpretation, for the following would be infelicitous if numerals came with an at-least interpretation:
(16) John has three sons or two. I forget which.

## Acknowledgements

For much helpful discussion on the thoughts presented here, I would like to thank Asaf Bachrach, Johan van Benthem, Hiba Bou Akar, Gennaro Chierchia, Noam Chomsky, Mike Collins, Cleo Condoravdi, Kai von Fintel, Danny Fox, Irene Heim, Sabine Iatridou, Roni Katzir, Angelika Kratzer, Ivona Kucerova, Alec Marantz, Paula Menendez-Benito, Bob Stalnaker, Tamina Stephenson, and Donca Steriade. Special thanks go to Kai von Fintel, Danny Fox, Irene Heim, Roni Katzir, and Bob Stalnaker. For help with $\mathrm{ET}_{\mathrm{E}} \mathrm{X}$, much thanks to Roni Katzir, Ezra Keshet, Ali Mohammad, Federico Mora, and Gabriel Zaccak. In a different sort of way, I should express my gratitude to the Social Sciences and Humanities Research Council of Canada, for their generous support through Doctoral Fellowship Award NO. 752-2004-0542.

## Bibliography

van Benthem, J. (1989). Semantics Parallels in Natural Language and Computation. In H.D. Ebbinghaus et al. (eds.), Logic Colloquium '87, Amsterdam: Elsevier, 331-375.
Chierchia, G. (2004). Scalar Implicatures, Polarity Phenomena, and the Syntax/Pragmatics Interface. In A. Belletti (ed.), Structures and Beyond, Oxford: Oxford University Press.
Collingwood, R.G. (1940). An Essay on Metaphysics. Oxford: Clarendon Press.
von Fintel, Kai. (2004). Lecture Notes, M.I.T.
Fox, D. (2006). Free Choice and the Theory of Scalar Implicature. MS, M.I.T.
Gamut, L.T.F. (1991). Logic, Language, and Meaning, Vol.1. Chicago: University of Chicago Press.
Gazdar, G. (1979). Pragmatics. New York, NY: Academic Press.
Groenendijk, J. and Stokhof, M. (1997). Questions. In J. van Benthem and A. ter Meulen (eds.), Handbook of Logic and Language, Cambrdige, MA: MIT Press, 1055-1124.

Heim, I. (1983). On the Projection Problem for Presuppositions. WCCFL 2: 114-125.
Hurford, J. (1974). Exclusive or Inclusive Disjunction. Foundations of Language 11: 409-411.

Karttunen, L. (1974). Presupposition and Linguistic Context. Theoretical Linguistics 1: 181-193.

Potts, C. (2006). Integrated Pragmatic Values. MS, UMass Amherst.
Quine, W.V.O. (1951). Mathematical Logic, Revised Edition. Cambridge, MA: Harvard University Press.
Rescher, N. (2000). Inquiry Dynamics. New Brunswick, NJ: Transaction Publishers.
van Rooy, R. and Schulz, K. (2004). Exhaustive Interpretation of Complex Sentences. Journal of Logic, Language, and Information, 13: 491-519.
Sauerland, U. (2004). Scalar Implicatures in Complex Sentences. Linguistics and Philosophy 27: 367-391.
Simons, M. (2000). Issues in the Semantics and Pragmatics of Disjunction. New York, NY: Garland.

Spector, B. (2003). Scalar Implicatures: Exhaustivity and Gricean Reasoning. Paper Presented at the ESSLLI'03 Student Session, Vienna.

Stalnaker, R. (1974). Pragmatic Presuppositions. In M. Munitz and P. Unger (eds.), Semantics and Philosophy, New York, NY: NYU Press, 197-213.

Zimmerman, T.E. (2000). Free Choice Disjunction and Epistemic Possibility. Natural Language Semantics 8: 255-290.

## Part II

Logic \& Computation

# An Improved Algorithm for Discounted Payoff Games 

Daniel Andersson<br>Uppsala University<br>daniel@math.uu.se


#### Abstract

We show that an optimal counterstrategy against a fixed positional strategy in a generalized discounted payoff game, where edges have individual discounts, can be computed in $O\left(m n^{2} \log m\right)$ strongly polynomial time, where $n$ and $m$ are the number of vertices and edges in the game graph. This results in the best known strongly subexponential time bound for solving two-player generalized discounted payoff games.


## 1 Introduction

Parity games, mean payoff games, and discounted payoff games constitute a chain of increasingly complex infinite two-player perfect information non-cooperative games played on finite graphs. They are closely related to $\mu$-calculus model checking, popular in computeraided program verification, and their associated decision problems all share the rare property of being interesting problems in the complexity class NP $\cap$ coNP with widely conjectured, but yet unproved, P-membership.

In this paper we investigate the problem of finding an optimal counterstrategy against a fixed positional strategy in a generalized discounted payoff game (DPG), where edges have individual discounting factors. We present a strongly polynomial algorithm which, when used as a component in the randomized combinatorial optimization schemes of (Björklund and Vorobyov 2005), gives the best strongly subexponential algorithm for two-player DPGs currently known. The improvement from the previously best known one is roughly $\sqrt{T(n, m)}$ compared with $T(n, m)$.

The problem of solving DPGs is related to the following well-known combinatorial optimization problems.

Stopping simple stochastic games (Condon 1992). This is a more general class of twoplayer games allowing random choice vertices with probability distributions on their outgoing edges. Their associated decision problem also belongs to NP $\cap$ coNP, and even the existence of a strongly polynomial algorithm for the one-player version is an open problem.

Generalized network flow problems (Cohen and Megiddo 1991). It turns out that the linear programs generated by one-player DPGs are dual to those generated by a particular case of the generalized transshipment problem.

The linear complementarity problem (Murty and Yu 1988; Cottle, Pang, and Stone 1992). Given a square matrix $M$ and a vector $q$, find some vector $z \geq 0$ such that $M z+q \geq 0$ and $z^{T}(M z+q)=0$. If $M$ is a $Z$ - and $P$-matrix, this can be done by solving a linear program whose feasible region contains a unique minimal element, and there is a strongly polynomial algorithm due to Chandrasekaran; see (Cottle et al. 1992). When the problem is generalized to allow several constraints for each variable (Cottle and Dantzig 1970), the minimal element property is preserved, but the existence of a strongly polynomial algorithm is an open problem.

## 2 Preliminaries

A generalized discounted payoff game (DPG) is a 5 -tuple $\left(V_{\mathrm{Max}}, V_{\mathrm{MIN}}, E, w, \lambda\right)$, where:

- $V_{\text {Max }}$ and $V_{\text {Min }}$ are disjoint sets of vertices belonging to the players Max and Min, respectively; $V=\left\{v_{1}, \ldots, v_{n}\right\}$ denotes $V_{\mathrm{Max}} \cup V_{\mathrm{MIN}}$;
- $E=\left\{e_{1}, \ldots, e_{m}\right\}$ is a set of directed edges between vertices in $V$, such that each vertex has at least one outgoing edge; we allow multiple edges between the same ordered pair of vertices and denote the set of edges from $u$ to $v$ by $E(u, v)$; we denote the set of outgoing edges from $v$ by $E(v)$; we define $E_{p}=\bigcup_{v \in V_{p}} E(v)$ for $p \in\{$ Max, $\operatorname{Min}\} ;$
- $w: E \rightarrow \mathbb{Q}$ is a weight function;
- $\lambda: E \rightarrow\{x \in \mathbb{Q}: 0<x<1\}$ is a discount function.

The game is played as follows. First, a token is placed at some initial vertex. Then, the following step is repeated ad infinitum: the owner of the vertex where the token is currently placed chooses an outgoing edge from this vertex and then moves the token to the head of the chosen edge.

This results in an infinite play $\pi=e_{i_{0}} e_{i_{1}} \ldots$, and the objective of MAX and Min is to maximize and minimize, respectively, its value $\mu(\pi)$, defined by

$$
\begin{equation*}
\mu(\pi)=w\left(e_{i_{0}}\right)+\lambda\left(e_{i_{0}}\right)\left(w\left(e_{i_{1}}\right)+\lambda\left(e_{i_{1}}\right)(\cdots)\right)=\sum_{j=0}^{\infty}\left(w\left(e_{i_{j}}\right) \prod_{0 \leq k<j} \lambda\left(e_{i_{k}}\right)\right) . \tag{1.1}
\end{equation*}
$$

A pure positional strategy $\sigma$ for player $p \in\{\operatorname{Max}, \operatorname{Min}\}$ is a selection of exactly one outgoing edge from each vertex owned by $p$, i.e., an element of the set $\prod_{v \in V_{p}} E(v)$ which we denote by $\mathcal{P}_{p}$. A play where $p$ only uses edges in $\sigma$ is said to be consistent with $\sigma$.

It follows from (Shapley 1953; Zwick and Paterson 1996) that there exist $\nu: V \rightarrow \mathbb{Q}$ and pure positional optimal strategies for Max and Min ensuring $\mu(\pi) \geq \nu(v)$ and $\mu(\pi) \leq \nu(v)$, respectively, for any play $\pi$ starting from $v$ that is consistent with the respective strategy. Henceforth, all strategies will be assumed to be pure positional, unless otherwise stated.

To solve a DPG is to compute the values $\nu(v)$ for all $v \in V$. From these values an optimal strategy for any player $p \in\{$ Max, Min $\}$ can be constructed by, for each vertex $u \in V_{p}$, selecting an edge $e \in E(u)$ such that $\nu(u)=w(e)+\lambda(e) \nu(v)$, where $v$ is the head of $e$. Conversely, given an optimal strategy for each player, we can easily compute $\nu(v)$ for any $v \in V$ by noting that $\nu(v)=\mu(\pi)$, where $\pi$ is the unique play starting from $v$ that is consistent with the given strategies.

A one-player DPG is a DPG with $V=V_{\mathrm{Max}}$ or $V=V_{\mathrm{Min}}$. Given a strategy for one of the players, a corresponding optimal counterstrategy is an optimal strategy for the opponent in the one-player game obtained by removing all edges not used by the given strategy and assigning all vertices to the opponent. Below, we will present an algorithm for the problem of finding an optimal counterstrategy, and then show how it can be used as a component in an algorithm for solving general (two-player) DPGs.

In any DPG, the roles of MAX and Min can be interchanged by, before and after the game is solved, changing the sign of each edge weight and computed vertex value, respectively. Thus, it suffices to consider the problem of finding an optimal counterstrategy for Min against a given strategy for MAx, i.e., solving a one-player DPG with $V=V_{\text {MIN }}$, as will be done below.

## 3 Solving One-Player DPGs

### 3.1 Linear Programming Formulation

We consider the problem of solving a one-player DPG with $V=V_{\text {Min }}$, i.e., for each vertex $v \in V$ finding the minimum weight $\nu(v)$ of any infinite "discounted path" from $v$. The vector $\left\langle\nu\left(v_{1}\right), \ldots, \nu\left(v_{n}\right)\right\rangle$ must be a feasible solution to the following system of inequalities:

$$
\begin{equation*}
x_{i} \leq w(e)+\lambda(e) x_{j} \quad \text { for all } v_{i}, v_{j} \in V \text { and } e \in E\left(v_{i}, v_{j}\right) \tag{1.2}
\end{equation*}
$$

This system has many special properties. We can easily find some feasible solution in $O(m)$ time, by noting that $\langle\xi, \ldots, \xi\rangle$ is feasible iff $\xi \leq \frac{w(e)}{1-\lambda(e)}$ for all $e \in E$. Furthermore, since $a_{i} \leq w(e)+\lambda(e) a_{j}$ and $b_{i} \leq w(e)+\lambda(e) b_{j}$ implies

$$
\begin{equation*}
\max \left\{a_{i}, b_{i}\right\} \leq \max \left\{w(e)+\lambda(e) a_{j}, w(e)+\lambda(e) b_{j}\right\}=w(e)+\lambda(e) \max \left\{a_{j}, b_{j}\right\} \tag{1.3}
\end{equation*}
$$

and similarly for the minima, the set of feasible solutions equipped with the binary operations of componentwise maximum and minimum forms a lattice. In particular, there can be at most one maximal solution $x^{*}$ such that $x^{*} \geq a$ (i.e., $x_{i}^{*} \geq a_{i}$ for $i=1, \ldots, n$ ) for any feasible solution $a$. To see that $\left\langle\nu\left(v_{1}\right), \ldots, \nu\left(v_{n}\right)\right\rangle$ is in fact the unique maximal solution, consider a play $e_{k_{0}} e_{k_{1}}, \ldots$ from $v_{i} \in V$ consistent with an optimal strategy for Min and note that the corresponding chain of inequalities implies $x_{i} \leq \nu\left(v_{i}\right)$.

Thus, the problem can be stated as the following linear program:

$$
\begin{align*}
\operatorname{maximize} & \sum_{i=1}^{n} x_{i}  \tag{1.4}\\
\text { subject to } & x_{i} \leq w(e)+\lambda(e) x_{j} \quad \text { for all } v_{i}, v_{j} \in V \text { and } e \in E\left(v_{i}, v_{j}\right) .
\end{align*}
$$

### 3.2 General Feasibility Algorithms

Megiddo (1983) gave an $O\left(m n^{3} \log m\right)$ strongly polynomial algorithm for finding a feasible solution to any linear program with $n$ variables, $m$ inequalities, and at most two variables per inequality. Since the algorithm also computes the feasible range for each variable, it can be used to solve our optimization problem (1.4). We could also use the deterministic $O\left(m n^{2}\left(\log m+\log ^{2} n\right)\right)$ algorithm or the expected $O\left(n^{3} \log n+m n \log ^{3} m \log n+m n \log ^{5} n\right)$ randomized algorithm of Cohen and Megiddo (1991).

Hochbaum and Naor (1994) suggested a simpler and faster deterministic $O\left(m n^{2} \log m\right)$ algorithm for finding a feasible solution. However, their algorithm does not compute the feasible ranges explicitly. We will show how to modify it so that it can be used to solve our optimization problem (1.4).

The approach of (Hochbaum and Naor 1994) is to use the Fourier-Motzkin elimination method (Schrijver 1986). To eliminate a variable $x_{i}$, all inequalities containing $x_{i}$ are replaced with inequalities $L \leq U$ for each pair $L \leq x_{i}$ and $x_{i} \leq U$ in the original system. Feasibility is preserved, and the method can be applied recursively to compute a feasible solution, or determine that no one exists. However, the number of inequalities created during a straightforward application of such repeated elimination may be exponential.

The algorithm in (Hochbaum and Naor 1994) limits the growth of the number of inequalities during the repeated Fourier-Motzkin elimination by simplifying the system before each elimination. Using a decision procedure by Aspvall and Shiloach (1980), the algorithm attempts to locate a small interval containing a feasible value for the variable to be eliminated. When the variable is restricted to this interval, all but $O(n)$ of the inequalities containing it can be identified as redundant, provided that the interval is sufficiently small.

In order to find such an interval in strongly polynomial time, the search is confined to certain "interesting" values. For any two distinct variables $x_{i}$ and $x_{j}$, the feasible region of the subsystem of inequalities not containing variables other than $x_{i}$ and $x_{j}$ lies between an upper and lower envelope, which are piecewise linear functions in the $x_{i} x_{j}$-plane; we denote the set of breakpoints of these functions by $B\left(x_{i}, x_{j}\right)$. The interesting values will be projections of such breakpoints.

The original algorithm is focused on finding any feasible solution. Feasibility is trivial for our system (1.2), and the algorithm must be modified. We now state the modified version, which solves our optimization problem (1.4), and refer the reader to (Hochbaum and Naor 1994) for a detailed description of the original algorithm.

### 3.3 Modified Algorithm

First, for each variable $x_{i}$, we compute values $l_{i}$ and $u_{i}$ such that $l_{i} \leq x_{i}^{*}=\nu\left(v_{i}\right) \leq u_{i}$. As $l_{i}$ we may take the $i$-th component of some feasible solution, which, as noted above, can be computed in $O(m)$ time. As $u_{i}$ we may use the value $\mu(\pi)$ of any play $\pi$ starting from $v_{i}$. We add to the original system (1.2) the inequalities $l_{i} \leq x_{i}$ and $x_{i} \leq u_{i}$ for $i=1, \ldots, n$.

Then we perform the steps $1-5$ below for $i=1, \ldots, n-1$ and maintain the following invariant:

$$
\text { before the } i \text {-th iteration, }\left\langle x_{i}^{*}, \ldots, x_{n}^{*}\right\rangle \text { is a feasible solution to the current system. }
$$

1. Let $B=\left\langle b_{1}, \ldots, b_{k}\right\rangle$ be the sorted sequence of $x_{i}$-coordinates of the breakpoints $\bigcup_{i<j \leq n} B\left(x_{i}, x_{j}\right)$ of the current system.

To maintain the invariant, we must make sure that $x_{i}^{*}$ remains a feasible value for $x_{i}$ after step 3.
2. Using the procedure of Aspvall and Shiloach (1980) (which, given any value $\xi$, decides whether $x_{i}^{*}<\xi$ in $O(m n)$ time), perform a binary search in $B$ to find $b_{l}$ and $b_{l+1}$ such that $b_{l} \leq x_{i}^{*} \leq b_{l+1}$ (if there is no $b_{l+1}$ such that $x_{i}^{*}<b_{l+1}$, then $x_{i}^{*}=b_{k}$ ).
3. Add the inequalities $b_{l} \leq x_{i}$ and $x_{i} \leq b_{l+1}$ to the current system.
4. For $j=i, \ldots, n$, discard all inequalities that are redundant with respect to all other inequalities that do not contain variables other than $x_{i}$ and $x_{j}$.

For any $x_{j}$, there will now be at most two inequalities containing both $x_{i}$ and $x_{j}$.
5. Apply the Fourier-Motzkin elimination method to $x_{i}$.

Since Fourier-Motzkin elimination preserves feasible ranges for the remaining variables, the invariant is preserved.

After $n-1$ iterations of steps $1-5$, what remains is $x_{n}$ and two inequalities $\alpha \leq x_{n}$ and $x_{n} \leq \beta$, where $\alpha$ and $\beta$ are constants. By the invariant, we have $\beta=x_{n}^{*}$, and thus we assign $\beta$ to $x_{n}$. Backtracking, i.e., restoring previously discarded inequalities containing $x_{n-1}$ and $x_{n}$, we assign to $x_{n-1}$ the maximum feasible value with respect to these inequalities and the value assigned to $x_{n}$. By the invariant and the lattice structure of the feasible region, continuing in this fashion for $x_{n-2}, \ldots, x_{1}$ gives us the optimal solution.

Our modifications do not significantly affect the worst case analysis in (Hochbaum and Naor 1994), and thus the running time is $O\left(m n^{2} \log m\right)$.

### 3.4 Equal Discounts

In (Andersson and Vorobyov 2006), a different approach to the problem is presented. For the particular case when all edges have the same discount, the resulting algorithm has a running time of $O\left(m n^{2}\right)$, which is a slightly better bound than the above.

## 4 Solving Two-Player DPGs

We now consider the problem of solving a two-player DPG. Björklund and Vorobyov (2005) give a general scheme for optimizing a wide class of functions, which contains strategy evaluation functions for many infinite games - the so-called recursively local-global (RLG) functions. We present the scheme applied to the case of DPGs, and refer the reader to (Björklund and Vorobyov 2005) for a detailed description of the more general approach.

Let $\mathcal{P}_{\mathrm{Max}}$ be the set of all (pure positional) strategies for MAX in $\left(V_{\mathrm{Max}}, V_{\mathrm{MiN}}, E, w, \lambda\right)$. A face $\mathcal{P}_{\text {Max }}^{\prime}$ of $\mathcal{P}_{\text {Max }}$ is the set of strategies for Max in a game ( $V_{\mathrm{MAX}}, V_{\mathrm{MIN}}, E^{\prime}, w, \lambda$ ) with $E^{\prime} \subseteq E$. Furthermore, if $E^{\prime}$ is obtained from $E$ by removing all but one of the outgoing edges from some vertex $v \in V_{\mathrm{Max}}$, then $\mathcal{P}_{\mathrm{Max}}^{\prime}$ is called a facet of $\mathcal{P}_{\mathrm{Max}}$. Any two strategies that differ only for a single vertex are called neighbors.

Suppose that eval : $\mathcal{P}_{\mathrm{Max}} \rightarrow \mathbb{Q}$ computes the optimal value of the linear program (1.4) resulting from fixing Max's strategy. To maximize eval on a face $\mathcal{P}_{\text {Max }}^{\prime} \subseteq \mathcal{P}_{\text {Max }}$, starting from some $\sigma \in \mathcal{P}_{\text {Max }}^{\prime}$, we use the following randomized iterated improvement algorithm from (Björklund and Vorobyov 2005).

1. If $\left|\mathcal{P}_{\text {MAX }}^{\prime}\right|=1$, then return $\sigma$.
2. Otherwise, select uniformly at random a facet $\mathcal{F}$ of $\mathcal{P}_{\text {MAX }}^{\prime}$ such that $\sigma \notin \mathcal{F}$.
3. Recursively find the maximum element $\sigma^{*}$ of $\mathcal{P}_{\operatorname{Max}}^{\prime} \backslash \mathcal{F}$ starting from $\sigma$.
4. Let $\sigma^{\prime}$ be the unique neighbor of $\sigma^{*}$ on $\mathcal{F}$.
5. If $\operatorname{eval}\left(\sigma^{\prime}\right) \leq \operatorname{eval}\left(\sigma^{*}\right)$, then return $\sigma^{*}$.
6. Otherwise, recursively find and return the maximum element of $\mathcal{F}$ starting from $\sigma^{\prime}$.

The correctness follows from the results on simple stochastic games in (Björklund and Vorobyov 2005), which also apply to DPGs. From the analyses done in (Kalai 1992; Matoušek et al. 1996) (described in (Björklund and Vorobyov 2005)), it follows that the expected number of calls to the subroutine for eval is $f\left(\left|V_{\mathrm{Max}}\right|,\left|E_{\mathrm{Max}}\right|\right)$, where

$$
\begin{equation*}
f(n, m)=e^{2 \sqrt{n \ln (m / \sqrt{n})}+O(\sqrt{n}+\ln m)} . \tag{1.5}
\end{equation*}
$$

Using the strongly polynomial algorithm presented in the previous section to compute eval, and recalling that the roles of Max and Min can be interchanged by simple transformations, we thus get a strongly subexponential algorithm with an expected total running time of

$$
\begin{equation*}
\min \left\{f\left(\left|V_{\mathrm{MAX}}\right|,\left|E_{\mathrm{MAX}}\right|\right), f\left(\left|V_{\mathrm{MIN}}\right|,\left|E_{\mathrm{MIN}}\right|\right)\right\} \cdot m n^{2} \log m . \tag{1.6}
\end{equation*}
$$

This is an improvement compared to previously available algorithms for eval, which either resorted to non-strongly polynomial LP-solvers, or to once again applying subexponential iterated improvement algorithms similar to the one above, resulting in a total expected running time of, roughly, $f\left(\left|V_{\mathrm{Max}}\right|,\left|E_{\mathrm{Max}}\right|\right) \cdot f\left(\left|V_{\mathrm{MiN}}\right|,\left|E_{\mathrm{MiN}}\right|\right)$.

## 5 Conclusions

We have described a new, and currently the best available, strongly polynomial algorithm for solving one-player generalized discounted payoff games, and shown how it can be incorporated into an algorithm for the two-player version, reducing the running time to roughly $\sqrt{T(n, m)}$ from $T(n, m)$.

It is likely that these results can be further improved, by exploiting more of the special properties of the linear programs arising from DPGs. The natural next step is to investigate more general classes of one-player games, such as one-player simple stochastic games with arbitrary probability distributions, for which the existence of a strongly polynomial algorithm is still an open problem.

## Acknowledgements

The author is grateful to Sergei Vorobyov for many useful comments and ideas.

## References

Andersson, D. and S. Vorobyov (2006, May). Fast algorithms for monotonic discounted linear programs with two variables per inequality. Preprint NI06019-LAA, Isaac Newton Institute for Mathematical Sciences, Cambridge, UK.
Aspvall, B. and Y. Shiloach (1980). Polynomial time algorithm for solving systems of linear inequalities with two variables per inequality. SIAM J. Comput. 9, 827-845.

Björklund, H. and S. Vorobyov (2005). Combinatorial structure and randomized subexponential algorithms for infinite games. Theoretical Computer Science 349(3), 347360.

Cohen, E. and N. Megiddo (1991). Improved algorithms for linear inequalities with two variables per inequality. In ACM Annual Symposium on Theory of Computing, pp. 145-154.
Condon, A. (1992). The complexity of stochastic games. Information and Computation 96, 203-224.
Cottle, R. W. and G. B. Dantzig (1970). A generalization of the linear complementarity problem. Journal of Combinatorial Theory 8, 79-90.
Cottle, R. W., J.-S. Pang, and R. E. Stone (1992). The Linear Complementarity Problem. Academic Press.

Hochbaum, D. S. and J. Naor (1994). Simple and fast algorithms for linear and integer programs with two variables per inequality. SIAM J. Comput. 23(6), 1179-1192.

Kalai, G. (1992). A subexponential randomized simplex algorithm. In 24th ACM STOC, pp. 475-482.

Matoušek, J., M. Sharir, and M. Welzl (1996). A subexponential bound for linear programming. Algorithmica 16, 498-516.

Megiddo, N. (1983). Towards a genuinely polynomial algorithm for linear programming. SIAM J. Comput. 12(2), 347-353.

Murty, K. G. and F.-T. Yu (1988). Linear Complementarity, Linear and Nonlinear Programming. Heldermann Verlag. http://ioe.engin.umich.edu/people/fac/books/.

Schrijver, A. (1986). Theory of Linear and Integer Programming. John Wiley and Sons.
Shapley, L. S. (1953). Stochastic games. Proc. Nat. Acad. Sci. USA 39, 1095-110.
Zwick, U. and M. Paterson (1996). The complexity of mean payoff games on graphs. Theor. Comput. Sci. 158, 343-359.

# Hybrid Layering 

Daniel Gorín<br>Universidad de Buenos Aires<br>dgorin@dc.uba.ar


#### Abstract

Many modal-like logics (including some temporal logics, description logics, hybrid logics, etc.) can be seen as fragments of first order logic. As such, a possible approach to automated deduction in these languages would be to implement satisfaction preserving translations and employ state-of-the-art first order theorem provers. As discussed in (Hustadt, Schmidt, and Weidenbach 1998; Areces, Gennari, Heguiabere, and de Rijke 2000) such approaches do not work if naive translations are used. We propose optimized translations for hybrid languages and report on some interesting behavior we encountered during our empirical testing.


## 1 Introduction

Many propositional modal logics can be seen as fragments of first order logic (FO), via simple satisfiability preserving translations. On the one hand, these translations are a tool of theoretical interest for certain results can be transferred this way from one domain to the other. On the other hand, since automated theorem proving for FO is an active and mature field, it would be reasonable to expect these translations to also be of practical relevance: given a modal formula, translate it into FO and feed it into a FO theorem prover. In this way, once you have a satisfiability preserving translation for a modal logic, you get a sound and complete theorem prover for free.

However, though correct, this process is rather unpractical. Even for formulas of the most basic modal logic $(K)$ the outcome of a naive translation is just too hard for FO theorem provers. In order to make this translation based procedure feasible, more complex translations have to be developed, like the functional (Hustadt, Schmidt, and Weidenbach 1998) and layered translations (Areces, Gennari, Heguiabere, and de Rijke 2000). Although they are quite different in nature, both were designed bearing in mind the way modern theorem provers work, and tailored to guide them in their search for a proof. Even then, this translation based scheme is usually outperformed by theorem provers built specifically for modal logics, like description logic reasoners.

In this paper, we focus on a family of modal logics called hybrid logics. These logics extend classical modal logics with machinery to name and refer to specific elements of the domain (as we'll see, this entails a weak notion of equality that make them more expressive than their modal counterparts). Naive translations from hybrid logics into FO are already known but translations specially designed for theorem proving had not yet been
investigated. Furthermore, although there exists much interest in extending description logic reasoners with hybrid operators, it is still not clear how this can be achieved in an efficient way, hence efficient translations into FO are an interesting alternative. Moreover, as we'll see in Section 3 of the present paper where we discuss empirical testing, we'll hit into an unexpected result: despite the gained expressive power, reasoning with hybrid logics using translations (even with plain standard translations) seems to be not only simpler than in the classical setup, but in fact, a feasible approach.
The basic modal language $\mathcal{M L}$ is built over a vocabulary of propositional symbols $\mathbf{P}=$ $\{p, q, r, \ldots\}$, and relational symbols $\mathbf{R}=\{m, n, \ldots\}$. Its formal syntax is given by $\mathcal{M} \mathcal{L}::=$ $p|\neg \varphi| \varphi \wedge \psi \mid\langle m\rangle \varphi$ where $p \in \mathbf{P}, m \in \mathbf{R}$ and $\varphi, \psi \in \mathcal{M L}$. Other boolean connectives are defined as usual, while $[m] \varphi$ is a shortcut for $\neg\langle m\rangle \neg \varphi$.

Modal formulas are interpreted over relational structures called Kripke models. These are FO models $\mathcal{M}=\langle W, \cdot \mathcal{M}\rangle$ with a domain $W$ (the set of worlds or states) and an interpretation function $\cdot \mathcal{M}$ that associates a two-place relation $R_{i}^{\mathcal{M}}$ with each $m_{i} \in \mathbf{R}$ and a one-place relation $P_{i}^{\mathcal{M}}$ with each $p_{i} \in \mathbf{P}$. For the sake of brevity, we will give the semantics of $\mathcal{M} \mathcal{L}$ formulas directly via their standard translation into FO. ${ }^{1}$ The vocabulary of this target logic consists of a two-place relation symbol $R_{i}$ for each $m_{i} \in \mathbf{R}$, a one-place relation symbol $P_{i}$ for each $p_{i} \in \mathbf{P}$ and a set of first-order variables $\{x, y, z, \ldots\}$. The standard translation $\mathrm{ST}_{x}$, takes formulas of $\mathcal{M} \mathcal{L}$ to formulas of FO with one free variable $x$ in the following way:

$$
\begin{array}{ll}
\mathrm{ST}_{x}\left(p_{i}\right) & \equiv P_{i} x, \text { for } p_{i} \in \mathbf{P} \\
\mathrm{ST}_{x}(\neg \varphi) & \equiv \neg \mathrm{ST}_{x}(\varphi) \\
\mathrm{ST}_{x}(\varphi \wedge \psi) & \equiv \mathrm{ST}_{x}(\varphi) \wedge \mathrm{ST}_{x}(\psi) \\
\mathrm{ST}_{x}(\langle m\rangle \varphi) & \equiv(\exists y)\left(R_{m} x y \wedge \mathrm{ST}_{y}(\varphi)\right), y \text { is fresh. }
\end{array}
$$

It follows that $\mathrm{ST}_{x}([m] \psi) \equiv(\forall y)\left(R_{m} x y \rightarrow \mathrm{ST}_{y}(\psi)\right)$. Given a Kripke model $\mathcal{M}=\langle W, \cdot \mathcal{M}\rangle$ and $w \in W$, the satisfaction relation $\vDash$ between $\mathcal{M}, w$ and formulas of $\mathcal{M} \mathcal{L}$ is defined as: $\mathcal{M}, w \models \varphi$ iff $\mathcal{M}, g[x \mapsto w] \models \mathrm{ST}_{x}(\varphi)$, for any first order valuation $g$. We call $w$ in this definition, the point of evaluation.

We now turn to hybrid logics. What distinguishes these logics from classical modal logics is their means to identify elements of the model and move the point of evaluation arbitrarily between different worlds. This identification of elements is usually done either by introducing name constants, known as nominals or by way of a special binder together with state variables which store the current point of evaluation. We briefly give the syntax and semantics of the very expressive hybrid $\operatorname{logic} \mathcal{H}(@, \downarrow)(\mathcal{M} \mathcal{L}$ extended with nominals, the satisfaction operator @, state variables and the $\downarrow$ binder) and regard the basic hybrid logic $\mathcal{H}(@)$ as a fragment of the former.

The language of the logic $\mathcal{H}(@, \downarrow)$ is built from a vocabulary consisting of proposition symbols $\mathbf{P}$ and relation symbols $\mathbf{R}$ as before, but also of nominal symbols $\mathbf{N}=\{i, j, k, \ldots\}$, and state variables $\mathbf{V}=\{\dot{x}, \dot{y}, \dot{z}, \ldots\}$. The set $\mathbf{S S}=\mathbf{N} \cup \mathbf{V}$ is called the set of state

[^27]symbols. The syntax is given by $\mathcal{H}(@, \downarrow)::=s|p| \neg \varphi|\varphi \wedge \psi|\langle m\rangle \varphi|\downarrow \dot{x} . \varphi| @_{s} \varphi$, where $\dot{x} \in \mathbf{V}, s \in \mathbf{S S}, p \in \mathbf{P}, m \in \mathbf{R}$ and $\varphi, \psi \in \mathcal{H}(@, \downarrow)$.

A hybrid model $\mathcal{M}$ is a Kripke model where $\cdot \mathcal{M}$ assigns a domain element $i^{\mathcal{M}}$ to each $i \in \mathbf{N}$. The translation into FO is as follows (the target language is extended with a constant $i$ for each $i \in \mathbf{N}$, and elements of $\mathbf{V}$ are considered as FO variables):

$$
\begin{aligned}
& \mathrm{HT}_{x}(\dot{x}) \equiv x=\dot{x}, \text { for } \dot{x} \in \mathbf{V} \mid \mathrm{HT}_{x}(\varphi \wedge \psi) \equiv \mathrm{HT}_{x}(\varphi) \wedge \mathrm{HT}_{x}(\psi) \\
& \mathrm{HT}_{x}(i) \equiv x=i \text {, for } i \in \mathbf{N} \quad \mathrm{HT}_{x}(\langle m\rangle \varphi) \equiv(\exists y)\left(R_{m} x y \wedge \mathrm{HT}_{y}(\varphi)\right), y \text { is fresh } \\
& \left.\mathrm{HT}_{x}\left(p_{i}\right) \equiv P_{i} x \text {, for } p_{i} \in \mathbf{P} \quad \mathrm{HT}_{x}(\downarrow \dot{x} \cdot \varphi) \equiv \mathrm{HT}_{x}(\varphi){ }^{\dot{x}} / x\right] \\
& \mathrm{HT}_{x}(\neg \varphi) \equiv \neg \mathrm{HT}_{x}(\varphi) \quad \mathrm{HT}_{x}\left(@_{s} \varphi\right) \equiv \mathrm{HT}_{y}(\varphi)\left[{ }^{[ } / s\right], y \text { is fresh. }
\end{aligned}
$$

Here, $\varphi[a / b]$ means "replace all the free occurrences of $a$ in $\varphi$ by $b$ ". The basic hybrid logic $\mathcal{H}(@)$ is the fragment of $\mathcal{H}(@, \downarrow)$ without $\downarrow$ and state variables. It is more expressive than $\mathcal{M L}$ but not more complex: the satisfiability problem for both logics is PSPACEcomplete. On the other hand, satisfiability for $\mathcal{H}(@, \downarrow)$ is undecidable. For an introduction to hybrid logics, see, for example, Chapter 7 of (Blackburn, de Rijke, and Venema 2002); their computational complexity is explored in (Areces, Blackburn, and Marx 1999).

## 2 Layered Translations for Hybrid Logics

The layered translation for the basic modal logic proposed in (Areces, Gennari, Heguiabere, and de Rijke 2000) was motivated by an empirical fact: FO theorem provers exhibit a poor behavior when fed with the standard translation of modal formulas. The following example will motivate a layered translation for $\mathcal{H}(@)$.

Example 1. Consider the trivially satisfiable formula $\varphi \equiv_{i}[m] \neg p \wedge @_{j}[m](p \rightarrow\langle m\rangle p)$. An FO prover based in ordered resolution with selection should be able to prove satisfiability by saturating the following set of clauses, obtained by turning the translation of $\varphi$ into clausal form (Bachmair and Ganzinger 2001). Here $k$ is a skolem function.

$$
C l=\left\{1:\left\{\neg R i x_{1}, \neg P x_{1}\right\}, 2:\left\{\neg R j x_{2}, \neg P x_{2}, R x_{2} k\left(x_{2}\right)\right\}, 3:\left\{\neg R j x_{3}, \neg P x_{3}, P k\left(x_{3}\right)\right\}\right\}
$$

Now, assume $P x_{1}$ is a selected atom in clause 1 and $P k\left(x_{3}\right)$ is maximal in clause 3. By taking $\left\{x_{1} \mapsto k\left(x_{3}\right)\right\}$ as most general unifier (mgu) we can derive 4: $\left\{\neg \operatorname{Rik}\left(x_{4}\right), \neg \operatorname{Rj} x_{4}\right.$, $\left.\neg P x_{4}\right\}$. And again, if $P x_{4}$ is the selected atom in clause 4, with $\left\{x_{4} \mapsto k\left(x_{3}\right)\right\}$ as mgu, from clauses 3 and 4 we derive $5:\left\{\neg \operatorname{Rik}\left(k\left(x_{5}\right)\right), \neg R j k\left(x_{5}\right), \neg R j x_{5}, \neg P x_{5}\right\}$.

If the $P$-atom is consistently selected, this procedure leads to the generation of terms of unbounded depth and, thus, to non-termination.

Arguably, other admissible orderings and/or selection functions could render this particular example terminating. However, the point we want to establish is the following. Clauses 1 and 3 were resolved upon $\neg P x_{1}$ and $P f\left(x_{3}\right)$, and these literals correspond to the translation of two totally unrelated occurrences of $p$. The former is predicating over worlds that are at distance 1 from $i$, while the latter must hold at a world at distance 2 from $j$. Since nothing
in the formula says that there must exist some world located at both distance 1 from $i$ and distance 2 from $j$, this inference is irrelevant, and we should prevent the unification of $\neg P x_{1}$ and $P f\left(x_{3}\right)$ in Example 1. The layered translation of (Areces, Gennari, Heguiabere, and de Rijke 2000) achieves separation for the basic modal language by marking each symbol with its modal depth; however this is easily shown to be unsound in the presence of nominals.

Instead of relying on the notion of modal depth, our layered translation for hybrid logics is based in the more general idea of "levels". Intuitively, a "level" is a description of a path from a nominal, that can be followed in an arbitrary model. These levels can be encoded using ground first order terms and, if used as an additional parameter of every relation symbol, can prevent irrelevant unifications. We now give the definition of the layered translation for $\mathcal{H}(@)$, which will make clear the exact purpose of levels.

The target vocabulary of the translation is similar to that of HT, but extended with a new constant $c_{i}$ for each nominal $i$ and an additional extra constant $c$, together with a unary function symbol $f_{m}$ for each $m \in \mathbf{R}$. Ground terms built exclusively from these new symbols will be called l-terms (for "level terms"). Additionally, the arity of all relational symbols is increased by one to accommodate the new l-term parameter used for layering. For the sake of simplicity (but without loss of generality), we will define the translation for $\mathcal{H}(@)$-formulas in negation normal form (i.e., negations only occur in front of propositions or nominals, and $\vee$ and [•] become primitive symbols).
Definition 1. For any $\mathcal{H}(@)$-formula $\varphi$ in negation normal form and any l-term $t$, the layered translation $\mathrm{LHT}_{x}(\varphi, t)$ is defined as:

$$
\begin{aligned}
& \operatorname{LHT}_{x}(i, t) \equiv x=i \wedge c_{i}=t \mid \operatorname{LHT}_{x}(\varphi \wedge \psi, t) \equiv \operatorname{LHT}_{x}(\varphi, t) \wedge \operatorname{LHT}_{x}(\psi, t) \\
& \mathrm{LHT}_{x}(\neg i, t) \equiv x \neq i \quad \operatorname{LHT}_{x}(\varphi \vee \psi, t) \equiv \operatorname{LHT}_{x}(\varphi, t) \vee \mathrm{LHT}_{x}(\psi, t) \\
& \operatorname{LHT}_{x}\left(p_{i}, t\right) \equiv P_{i} x t \quad \operatorname{LHT}_{x}(\langle m\rangle \varphi, t) \equiv(\exists y)\left(R_{m} x y t \wedge \operatorname{LHT}_{y}\left(\varphi, f_{m}(t)\right)\right) \\
& \mathrm{LHT}_{x}\left(\neg p_{i}, t\right) \equiv \neg P_{i} x t \quad \begin{array}{ll}
\mathrm{LHT}_{x}([m] \varphi, t) & \equiv(\forall y)\left(R_{m} x y t \rightarrow \mathrm{LHT}_{y}\left(\varphi, f_{m}(t)\right)\right) \\
\mathrm{LHT}_{x}\left(@_{i} \varphi, t\right) & \equiv \mathrm{LHT}_{x}\left(\varphi, c_{i}\right)[x /]
\end{array}
\end{aligned}
$$

as before, the variable $y$ in the clauses for the modalities is fresh.
The encoding of levels using l-terms is the key to understanding the layered translation for $\mathcal{H}(@)$. For each nominal $i$, the constant $c_{i}$ should be thought of as "level 0 with respect to $i$." The world designated by $i$ is clearly at level 0 with respect to $i$ and we use the term $c_{i}$ to represent this. Now, every world accessible from $i$ via $m \in \mathbf{R}$ is at level $f_{m}\left(c_{i}\right)$, and so on. But levels are not necessarily distinct (e.g., in any model for $@_{i}\langle m\rangle j$, levels $f_{m}\left(c_{i}\right)$ and $c_{j}$ coincide). Taking a look at the clause for $\mathrm{LHT}_{x}(i, t)$ we see that if $x=i$ and $x$ was taken to be at level $t$, then it must also be the case that $c_{i}=t$. In the same way, in the clause for $\mathrm{LHT}_{x}(\langle m\rangle \varphi, t)$, if $x$ was taken to be at level $t, y$ must be at level $f_{m}(t)$. Finally observe that in $\mathrm{LHT}_{x}\left(@_{i} \varphi, t\right)$, since @ moves the point of evaluation to a new state regardless of $x$ (and its level $t$ ), we have to reset the level to $c_{i}$.

Theorem 1. For every $\mathcal{H}(@)$-formula $\varphi$ in negation normal form, and all l-terms $t$, $\mathrm{LHT}_{x}(\varphi, t)$ is satisfiable iff $\varphi$ is satisfiable.

Proof. For the left to right implication, observe that equality of l-terms occurs only positively in $\operatorname{LHT}_{x}(\varphi, t)$ and, thus, any model for $\mathrm{HT}_{x}(\varphi)$ is trivially turned into a model for $\mathrm{LHT}_{x}(\varphi, t)$ by making all the l-terms denote the same domain element.

For the other direction, let $\mathcal{M}=\langle W, \cdot \mathcal{M}\rangle$ be a model for the vocabulary of the layered translation of $\varphi$. split $(\mathcal{M})$ is a model for the vocabulary of $\varphi$, its domain is $W \times W$, and

$$
\begin{aligned}
R_{i}^{\text {split }(\mathcal{M})} & =\left\{\left(\left(x, t^{\mathcal{M}}\right),\left(y, f_{m_{i}}(t)^{\mathcal{M}}\right)\right) \mid \text { for } t \text { an l-term and }\left(x, y, t^{\mathcal{M}}\right) \in R_{i}^{\mathcal{M}}\right\} \\
P_{i}^{\text {split }(\mathcal{M})} & =\left\{\left(x, t^{\mathcal{M}}\right) \mid \text { for } t \text { an l-term and }\left(x, t^{\mathcal{M}}\right) \in P_{i}^{\mathcal{M}}\right\} \\
i^{\text {split }(\mathcal{M})} & =\left(i^{\mathcal{M}}, c_{i}^{\mathcal{M}}\right)
\end{aligned}
$$

We show by induction on $\varphi$ that for every valuation $v$ and any l-term $t$, if $\mathcal{M}, v \models$ $\operatorname{LHT}_{x}(\varphi, t)$ then $\operatorname{split}(\mathcal{M}),\left(v(x), t^{\text {split }(\mathcal{M})}\right) \models \varphi$. We prove only the relevant cases:

Case $\varphi \equiv i$. If $\mathcal{M}, v \models \operatorname{LHT}_{x}(i, t)$ then $\mathcal{M}, v \models\left(x=i \wedge t=c_{i}\right)$ and, thus, $i^{\mathcal{M}}=v(x)$ and $t^{\mathcal{M}}=c_{i}^{\mathcal{M}}$. But this means that $\left(v(x), t^{\mathcal{M}}\right)=\left(i^{\mathcal{M}}, c_{i}^{\mathcal{M}}\right)=i^{\text {split }(\mathcal{M})}$ and, therefore , $\operatorname{split}(\mathcal{M}),\left(v(x), t^{\mathcal{M}}\right) \models i$.

Case $\varphi \equiv p$. Suppose $\mathcal{M}, v \vDash \operatorname{LHT}_{x}(p, t)$, that is, $\mathcal{M}, v \models P x t$; it follows that $\left(v(x), t^{\mathcal{M}}\right) \in$ $P^{\mathcal{M}}$ and, consequently, $\operatorname{split}(\mathcal{M}),\left(v(x), t^{\mathcal{M}}\right) \models p$.

Case $\varphi \equiv\langle m\rangle \psi$. Assume $\mathcal{M}, v \models(\exists y)\left(R_{m} x t y \wedge \operatorname{LHT}_{y}\left(\psi, f_{m}(t)\right)\right)$. There must exist some $w \in W$ such that $\left(v(x), t^{\mathcal{M}}, w\right) \in R_{m}^{\mathcal{M}}$ and $\mathcal{M}, v[y \mapsto w] \vDash \operatorname{LHT}_{y}\left(\psi, f_{m}(t)\right)$. Now, from the definition of $\operatorname{split}(\mathcal{M}),\left(\left(v(x), t^{\mathcal{M}}\right),\left(w, f_{m}(w)^{\mathcal{M}}\right)\right) \in R_{m}^{\mathcal{M}}$ and, by inductive hypothesis, $\operatorname{split}(\mathcal{M}),\left(w, f_{m}(w)^{\mathcal{M}}\right) \models \psi$. Hence, $\operatorname{split}(\mathcal{M}),\left(v(x), t^{\mathcal{M}}\right) \models\langle m\rangle \psi$.

Case $\varphi \equiv @_{i} \psi$. If $\mathcal{M}, v \models \operatorname{LHT}_{x}\left(@_{i} \psi, t\right)$, then $\mathcal{M}, v \models \operatorname{LHT}_{x}\left(\psi, c_{i}\right)\left[{ }^{x} / i\right]$ and this is equivalent to $\mathcal{M}, v\left[x \mapsto i^{\mathcal{M}}\right] \models \mathrm{LHT}_{x}\left(\psi, c_{i}\right)$. By inductive hypothesis, $\operatorname{split}(\mathcal{M}),\left(i^{\mathcal{M}}, c_{i}^{\mathcal{M}}\right) \models$ $\psi$, but since $i^{\text {split }(\mathcal{M})}=\left(i^{\mathcal{M}}, c_{i}^{\mathcal{M}}\right)$, it follows that $\operatorname{split}(\mathcal{M}),\left(v(x), t^{\mathcal{M}}\right) \models @_{i} \psi$.

Observe that we have reserved an additional constant $c$, that will be used as the "level 0 with respect to the point of evaluation for the whole formula". The following example shows the layered translation in action.

Example 2. Let's consider once again the formula $\varphi$ of Example 1. An FO prover fed with $(\exists x) \mathrm{LHT}_{x}(\varphi, c)$ as input would have to saturate the following set of clauses (where $k$ is, once more, a skolem function and $f$ is a shortcut for $f_{m}$ ):

$$
C l=\left\{\begin{array}{l}
1:\left\{\neg R i x_{1} c_{i}, \neg P x_{1} f\left(c_{i}\right)\right\}, \\
2:\left\{\neg R j x_{2} c_{j}, \neg P x_{2} f\left(c_{j}\right), R x_{2} k\left(x_{2}\right) f\left(c_{j}\right)\right\}, \\
3:\left\{\neg R j x_{3} c_{j}, \neg P x_{3} f\left(c_{j}\right), \operatorname{Pk}\left(x_{3}\right) f\left(f\left(c_{j}\right)\right)\right\}
\end{array}\right\}
$$

Observe that, unlike Example 1, no inference can be done using clauses 1 and 3, for $P x_{1} f\left(c_{i}\right)$ and $P k\left(x_{3}\right) f\left(f\left(c_{j}\right)\right)$ don't unify.

Incidentally, LHT made Example 2 terminating regardless of the strategy used by the theorem prover. However, this is not always the case: the layered translation by itself does not ensure termination.

The idea of using ground terms to code levels can be extended to $\mathcal{H}(@, \downarrow)$. Here we only discuss the main intuition: when translating a formula of the form $\downarrow \dot{x} . \varphi$ using a free variable $x$ taken to be at level $t$, record the fact that $\dot{x}$ denotes the variable $x$ at level $t$. This can be done, for example, by keeping a map $v$ from state variables to first order variables and a map $l$ from state variables to l-terms. The most relevant rules of this translation are shown below.

$$
\begin{aligned}
\operatorname{LHT}_{x}(\downarrow \dot{x} \cdot \varphi, v, l, t) & \equiv \operatorname{LHT}_{x}(\varphi, v[\dot{x} \mapsto x], l[\dot{x} \mapsto t], t) \\
\operatorname{LHT}_{x}(\dot{x}, v, l, t) & \equiv v(\dot{x})=x \wedge l(\dot{x})=t \\
\operatorname{LHT}_{x}\left(@_{x} \varphi, v, l, t\right) & \equiv \operatorname{LHT}_{v(\dot{x})}(\varphi, v, l, l(\dot{x})) \\
\operatorname{LHT}_{x}\left(@_{i} \varphi, v, l, t\right) & \equiv \operatorname{LHT}_{y}\left(\varphi, v, l, c_{i}\right)\left[^{y} / i\right], y \text { is fresh }
\end{aligned}
$$

Theorem 2. For all $\mathcal{H}(@, \downarrow)$-formulas $\varphi$ in negation normal form, and all l-terms $t$, $\mathrm{LHT}_{x}(\varphi, t)$ is satisfiable iff $\varphi$ is satisfiable.

## 3 Testing

The layered translation is designed to insert ground terms that block irrelevant inferences. In many formulas (for instance, the one in Example 1) this is enough to improve the behavior of resolution. However, this is not necessarily always the case: the layered translation produces longer and structurally more complex formulas and, moreover, they usually contain additional equality atoms (and equality reasoning is known to be hard). Hence, it is a relevant question whether the beneficial effect of layering is not superseded by the overhead. In this section we present the results of the tests we conducted to investigate this issue.

Test Setup: We used the hgen random generator (Areces and Heguiabehere 2003) to obtain our test sets. hgen is highly configurable, but the relevant parameters we used were the number of proposition symbols $\left(\#_{p}\right)$, nominals $\left(\#_{n}\right)$ and state variables $\left(\#_{v}\right)$; the maximum nesting of classical modalities $\left(d_{\diamond}\right)$ and hybrid operators $\left(d_{@}, d_{\downarrow}\right)$ and the relative frequencies of propositions $\left(f_{p}\right)$, nominals $\left(f_{n}\right)$ and state variables $\left(f_{v}\right)$, and of modal and hybrid operators $\left(f_{\diamond}, f_{@}, f_{\downarrow}\right)$. Only one relation symbol was used.

The normal form used by hgen ensures that once parameters are fixed, the probability of satisfiability is in inverse proportion to the number of clauses. For each test, we fixed adequate parameters and generated batches of 60 formulas, each batch with an increasing number of clauses. These formulas were then translated using both the standard and layered translations and fed to SPASS version 2.2 (SPASS 2006), a first order theorem prover which extends superposition (a generalization of resolution for languages with equality) by sorts and a splitting rule for case analysis. SPASS was invoked with the auto mode
switched on, and no sort constraints were built. ${ }^{2}$ Tests where performed on a Pentium 4 2.6 GHz with 512 Mb RAM under Linux (kernel version 2.6.8.1).
$\mathcal{M} \mathcal{L}$ : When applied to formulas in $\mathcal{M} \mathcal{L}$, the output of LHT can be seen just as a syntactic variation of LT, the layered translation described in (Areces, Gennari, Heguiabere, and de Rijke 2000). However, it is still important to verify wether the use of functional terms instead of additional relation symbols alters in any way the difficulty of the problem. Thus, in our first test, we set hgen's parameters to get purely modal formulas of medium to high difficulty, containing three different propositional symbols and a maximum modal depth of three.


Figure 1.1: Test results for $\mathcal{M} \mathcal{L}$ input, $\#_{p}=3, d_{\diamond}=3$.
Results are shown in Figure 1.1. As expected, layering leads to a much improved performance of the prover. Nevertheless, we can see that these formulas, generated with a low modal depth, are still quite difficult for SPASS. Even with LHT, the prover times out $90 \%$ of the time for formulas with more than 125 clauses.

About the comparison between LT and LHT, the first thing we can observe is that the former consistently shows better median execution times than the latter. However, these differences are almost negligible and are probably due to additional overhead introduced by LHT (e.g., more term indexing is required). On the other hand, there is no clear winner in terms of timeout rate; this suggests that what is prevailing is the heuristic used by the prover to explore the solution space in each case.
$\mathcal{H}(@)$ : We now move to testing over hybrid formulas, comparing the behavior between the standard translation HT and the layered version LHT. In our first test, we run a similar configuration as the test described above: three propositional symbols and up to three nested (classical) modalities, but we also allow for three nominals and at most two nested occurrences of @ (hence, the overall modal depth ${ }^{3}$ of this configuration is five).

The results we obtained were totally unexpected: the test turned to be much simpler for SPASS than the previous one. We'll try to explain this behavior below, but to complete the

[^28]picture, and given that we could now explore much more complex formulas, we performed tests with formulas where the maximum level of nested (classical) modalities was 5, 7 and 10; i.e., overall modal depth of 7,9 and 12 , respectively.


Figure 1.2: Tests results for $\mathcal{H}(@)$ input, $\#_{p}=3, \#_{n}=3, f_{p}=f_{n}=f_{\diamond}=f_{@}=\frac{1}{2}, d_{@}=2$.
Figure 1.2 shows the results. In general, using LHT results in shorter times and it scales better under increasing modal depth and/or number of clauses, even though the cpu-time difference between HT and LHT is not as striking in this case (w.r.t. the difference between LT and ST). But, and this is important, applying LHT results in significantly less timeouts. Figure 1.2(c) shows the percentage of timeouts for the harder test $\left(d_{\diamond}=10\right)$. Notice that in the region of 20 to 40 clauses, even though the median cpu-time is 10 seconds (quite far form the 100 seconds timeout limit), HT reaches $30 \%$ timeouts while for LHT timeouts are below $10 \%{ }^{4}$
$\mathcal{H}(@, \downarrow)$ : For our final test, we investigate formulas containing the $\downarrow$ operator. As the graphs in Figure 1.3 show, we obtain similar results as those found for the $\mathcal{H}(@)$ language.

[^29]Test sets are even easier in this case, and we are able to run cases of up to overall modal depth 14 without serious timeout problems. In the tests, the running times of HT and LHT are comparable (with LHT times slightly higher in the median: formulas are too easy for the layering effect to be noticeably, but the optimized translation has to pay still the price of more complex formulas). But notice again, in Figure 1.3(c), the improvement on the timeout percentage between HT and LHT.


Figure 1.3: Tests results for $\mathcal{H}(@, \downarrow)$ input, $\#_{p}=3, \#_{n}=0, \# v=2, f_{p}=f_{v}=\frac{1}{2}, d_{@}=2, d_{\downarrow}=2$, $f_{\diamond}=f_{@}=f_{\downarrow}=\frac{1}{3}$.

Why are hybrid formulas so much simpler for SPASS than modal ones? This is, in itself, quite an interesting issue (which is independent from the HT vs. LHT comparison that is the main topic of this paper). It deserves careful attention and much more extensive testing. For the moment, we can only advance preliminary conjectures. The observed behavior can be the result of a combination of reasons:
Reason 1. Let $\psi$ be a pure modal formula, and let $\psi\left[{ }^{p} / i\right]$ be the result of replacing all the occurrences of the proposition symbol $p$ by some nominal $i$. If $\psi$ is unsatisfiable, $\psi\left[{ }^{p} /{ }_{i}\right]$ must be unsatisfiable too (any valuation for $i$ is an acceptable valuation for $p$ ), however, it can still be the case that $\psi$ is satisfiable while $\psi\left[{ }^{p} / i\right]$ is not. So, if we take a batch of pure modal formulas and replace all the occurrences of some proposition symbol by a fixed nominal, then the number of unsatisfiable hybrid formulas in the resulting batch may increase but never decrease.

We can regard a batch of randomly generated formulas of the basic hybrid logic without @ as a batch of randomly generated pure modal formulas where several replacements like the abovementioned were performed. Thus, we should expect the probability of finding an unsatisfiable formula in the hybrid batch to be higher than in the corresponding modal one. It is known that resolution provers perform better on unsatisfiable input and probably this higher ratio of unsatisfiable formulas is sufficient to make a difference.

To sustain this hypothesis, we show in Figure 1.4(a) the results of an experiment where we run three tests that only differed in the relative frequency of nominals and propositional symbols. It is clearly observable that the higher the probability of nominals occurring in a formula, the quicker we reach the unsatisfiable region.

A similar reasoning can be done with respect the @ operator: take a random generated pure (multi-) modal formula $\psi$, and let $\psi\left[{ }^{r} / @_{i}\right]$ be the result of replacing all the occurrences of $\langle r\rangle$ and $[r]$ in $\psi$ by $@_{i}$ for some nominal $i$. It is straightforward to see that $\psi$ must be satisfiable whenever $\psi\left[{ }^{r} / @_{i}\right]$ is too, although the converse is not true.

Summing up, when considering FO resolution over randomly generated $\mathcal{M L}$ formulas, the hardest instances will be found among large, structurally complex, satisfiable formulas. Beyond a certain number of clauses, current resolution based theorem provers are just not up to the task of efficiently computing a saturated set. But the presence of hybrid operators increases the chance of turning the formula unsatisfiable making saturation unnecessary.


Figure 1.4: Satisfiability portions for varying frequencies of symbols

Reason 2. The main goal of layering is to avoid redundant unifications. But, as we will show in a moment, the presence of the @ operator can make redundant unifications less frequent without any need of a special encoding. Taking pains to enforce layering is, then, not as crucial as in the purely modal case. We illustrate this idea with a very simple example. Consider the formula $\left[m_{2}\right][m] p \wedge\left[m_{3}\right]\langle m\rangle \top$. To prove its satisfiability using HT we ought to saturate the following set of clauses (where $k$ is a skolem function):

$$
C l=\left\{1:\left\{\neg R_{2} x_{1} x_{2}, \neg R x_{2} x_{3}, P x_{3}\right\}, 2:\left\{\neg R_{3} x_{4} x_{5}, R x_{5} k\left(x_{5}\right)\right\}\right\}
$$

and from this set we may infer $3:\left\{\neg R_{2} x_{6} x_{7}, \operatorname{Pk}\left(x_{7}\right), \neg R_{3} x_{8} x_{7}\right\}$ by way of an unnecessary
unification. On the other hand, by replacing the modalities $m_{2}$ and $m_{3}$ by $@_{i}$ and $@_{j}$ we obtain the hybrid formula $@_{i}[m] p \wedge @_{j}\langle m\rangle \top$ which gets translated into:

$$
C l^{\prime}=\left\{1:\left\{\neg R i x_{1}, P x_{1}\right\}, 2:\{R j k(j)\}\right\}
$$

from where no new clauses may be derived. Observe that since nominals $i$ and $j$ don't unify, they are in a way already introducing a form of layering.
Reason 3. By way of a similar analysis, we may compare formulas in $\mathcal{H}(@, \downarrow)$ against pure modal ones and conclude that the former are more likely to become unsatisfiable with a lesser number of clauses. In fact, in Figure 1.4(b) we run tests where the relative frequency of $\downarrow$ gradually increases (and the frequency of modal operators and @ decreases accordingly), and we notice that we reach unsatisfiability very fast indeed. However we are rather cautious about these results: due to the binding of variables, it is far more difficult to generate non-trivial random $\mathcal{H}(@, \downarrow)$ formulas than it is for $\mathcal{H}(@)$ and we don't rule out the possibility of some form of bias being introduced by the random generation algorithm. We are in contact with the developers of hgen and working on this matter.
Reason 4. Modal depth is a well-accepted measure of formula complexity for modal languages and, as Figure 1.1 shows, modern theorem provers get in trouble even for very low values of it. However, in the hybrid case, modal depth may be a less reliable indicator. For instance, it is simple to construct hybrid formulas that are semantically equivalent but whose modal depths differ as much as wanted: take, for example, $\langle m\rangle @_{i_{1}}\langle m\rangle @_{i_{2}} \ldots\langle m\rangle @_{i_{n}}\langle m\rangle \top$ and $\langle m\rangle \top \wedge @_{i_{1}}\langle m\rangle \top \wedge @_{i_{2}}\langle m\rangle \top \wedge \ldots \wedge @_{i_{n}}\langle m\rangle T$. Furthermore, although Figures 1.2 and 1.3 show a correlation between overall modal depth and complexity in the hybrid setting, the differences in execution times don't look large enough as to discard other sources of complexity (e.g., formula size in symbols). All this would explain why formulas that we a priori believed to be complex turned out to be much simpler than expected. It would also suggest that hybrid languages may need distinct testing methodologies, which still would have to be developed.

## 4 Conclusion

We have described a new satisfiability preserving translation from hybrid logics into FO, especially crafted to be used with resolution based first order theorem provers. The key idea is the encoding via ground terms of a notion of levels inspired by that of (Areces, Gennari, Heguiabere, and de Rijke 2000). We have shown that an improvement both in execution speed and response rate may be expected from using this translation. Additionally, we reported an unexpected behavior: hybrid formulas seem to be better suited for resolution based first order theorem proving than pure modal ones.

We will continue exploring the behavior of this heuristic in larger parts of the problem space, and extending this notion of layering to other hybrid logics (e.g., including inverse operators). Furthermore, it may be interesting to see if a decision procedure for $\mathcal{H}(@)$ can be obtained from LHT. We are also trying to get a better understanding on the reasons behind the difference in performance between modal and hybrid formulas.

## References

Areces, C., P. Blackburn, and M. Marx (1999). A road-map on complexity for hybrid logics. In J. Flum and M. Rodríguez-Artalejo (Eds.), Computer Science Logic, Number 1683 in LNCS, pp. 307-321. Springer. Proceedings of the 8th Annual Conference of the EACSL, Madrid, September 1999.
Areces, C., R. Gennari, J. Heguiabere, and M. de Rijke (2000). Tree-based heuristics in modal theorem proving. In W. Horn (Ed.), Proc. of ECAI'2000, Berlin, Germany, pp. 199-203.
Areces, C. and J. Heguiabehere (2003). hGen: A random CNF formula generator for Hybrid Languages. In Proceedings of Methods for Modalities 3, Nancy, France.

Bachmair, L. and H. Ganzinger (2001). Resolution theorem proving. In J. Robinson and A. Voronkov (Eds.), Handbook of Automated Reasoning, Volume 1, Chapter 2, pp. 19-99. Amsterdam, the Netherlands: Elsevier.

Blackburn, P., M. de Rijke, and Y. Venema (2002). Modal Logic. Cambridge University Press.

Hustadt, U., R. Schmidt, and C. Weidenbach (1998, May 5-8). Optimised functional translation and resolution. In H. de Swart (Ed.), Automated Reasoning with Analytic Tableaux and Related Methods, International Conference, TABLEAUX'98, Oisterwijk, The Netherlands, Proceedings, Volume 1397 of Lecture Notes in Computer Science, pp. 36-37. Springer.

SPASS (2006). SPASS Home Page. http://spass.mpi-sb.mpg.de/. Last access: January 2006.

# Zero-One Law and Rational Quantifiers 

Jarmo Kontinen<br>University of Helsinki<br>jarmo.kontinen@helsinki.fi


#### Abstract

We study extensions of finite variable logic $\mathcal{L}_{\infty}^{\omega} \omega$ by generalized quantifiers. We use strong version of so-called extension axioms and pebble games to show the zero-one law for the obtained logic. In some cases we show that the zero-one law does not hold by constructing a sentence with no limit probability. We construct these sentences with as few variables as possible and thus find the exact number of variables for which the zero-one law breaks.


## 1 Introduction

Many of the problems in computer science can be analysed in mathematical logic when restricting to finite models. This is one of the main reasons that led finite model theory to develop as an independent field of mathematical logic. Although many of the questions can be answered using first order logic, many still require more stronger logics. Sometimes it just comes down to finding the right one. For example, the properties definable in fixed point logic over ordered structures are exactly the properties that can be computed in polynomial time.

There are different ways to increase the expressive power of first order logic. Extending it by generalized quantifiers is a way to increase it in a controlled way. The universal and the existential quantifier known from the context of first order logic can also be seen as generalized quantifiers. Mostowski (1957) initiated the study of new quantifiers with his work on cardinality quantifiers. Since then, there has been a growing interest in generalized quantifiers. A decade after Mostowski, Lindström (1966) formalized the concept of generalized quantifier.

A $\operatorname{logic} \mathcal{L}$ is said to have the zero-one law if every sentence of $\mathcal{L}$ is either true in almost all finite models, or false in almost all finite models. Zero-one law is a method for proving non-definability, which can be difficult to show otherwise. Zero-one law for first order logic was shown by Glebskii, Kogan, Liogon’kii and Talanov (1969) and later independently by Fagin (1976). Fagin's proof for first order logic relies on properties called extension axioms. This is also the method of showing the zero-one law in this paper, although we need a stronger version of the extension axioms.

We study the finite variable logic extended by one quantifier. We concentrate on a specific class of quantifiers, the rational quantifiers, and we show that the zero-one law
holds with certain assumptions on the quantifier. When this condition is not met, we construct a sentence which breaks the zero-one law.

## 2 Generalized Quantifiers

Lindström (1966) defined generalized quantifiers as follows.
Definition 1. Let $\left(r_{1}, \ldots, r_{n}\right)$ be a tuple of natural numbers. A Lindström quantifier of type $\left(r_{1}, \ldots, r_{n}\right)$ is a collection $Q$ of structures of relational vocabulary $\tau_{s}=\left(P_{1}, \ldots, P_{n}\right)$ such that $P_{i}$ is of arity $r_{i}$ for $1 \leq i \leq r$, and $Q$ is closed under isomorphisms. The arity of a quantifier $Q$ is $\operatorname{ar}(Q)=\max \left\{\operatorname{ar}\left(P_{1}\right), \ldots, \operatorname{ar}\left(P_{n}\right)\right\} . Q$ is called simple if $n=1$ and unary if $\operatorname{ar}(Q)=1$.

Definition 2. A simple unary generalized quantifier $Q$ is monotone (increasing), if for all structures $\left(M, P^{M}\right) \in Q$ and for all subsets $X \subseteq M$ such that $P^{M} \subseteq X$, then also $(M, X) \in Q$.

All the quantifiers studied here are simple unary monotone quantifiers. Monotone simple unary quantifiers have a characterisation through their threshold functions.

Definition 3. Let $n \in \mathbb{N}$ and $Q$ a monotone simple unary quantifier. Let $M$ be a structure of size n. A threshold function $f_{Q}(n)$ of $Q$ is

$$
f_{Q}(n)=\min \{m \in \mathbb{N} ; A \subseteq M,|A|=m \text { and }(M, A) \in Q\}
$$

In the next definition we consider rational numbers $s / t$ in reduced form.
Definition 4. Rational quantifier is a monotone simple unary quantifier with a threshold function $f_{Q}(n)=\ulcorner s / t \cdot n\urcorner$, where $s$ and $t$ are natural numbers, $0<s / t<1$. Here $\ulcorner s / t \cdot n\urcorner$ means that the product $s / t \cdot n$ is rounded up to least natural number. We denote the rational quantifier of a threshold function $f_{Q}(n)=\ulcorner s / t \cdot n\urcorner$, with $\exists \geq s / t$.
(1) Here are some examples of monotone simple unary quantifiers.

$$
\begin{aligned}
\exists & =\left\{\left(M, P^{M}\right):\left|P^{M}\right| \geq 1\right\} . \\
\forall & =\left\{\left(M, P^{M}\right):\left|P^{M}\right|=|M|\right\} . \\
\exists^{\geq 1 / 2} & =\left\{\left(M, P^{M}\right): \quad\left|P^{M}\right| \geq|M| / 2\right\} .
\end{aligned}
$$

### 2.1 Extending $\mathcal{L}_{\infty \omega}^{k}$ by Generalized Quantifiers

Infinitary logic $\mathcal{L}_{\infty \omega}$ was introduced by Tarski (1961). In first order logic new formulas are obtained by combining formulas with connectives $\neg, \wedge, \vee$ and quantifiers $\exists, \forall . \quad \mathcal{L}_{\infty \omega}$ extends first order logic by allowing arbitrary large conjunctions and disjunctions. It does
not have the zero-one law since it is possible to define every isomorphism-closed class of finite structures in it. We consider the finite variable logic

$$
\mathcal{L}_{\infty \omega}^{\omega}=\bigcup_{k=1}^{\omega} \mathcal{L}_{\infty \omega}^{k}
$$

where $\mathcal{L}_{\infty \omega}^{k} \quad$ is the set of formulas of infinitary logic with at most $k$ variables. Zero-one law for finite variable logic was shown by Kolaitis and Vardi (1990).

When extending the logic $\mathcal{L}_{\infty \omega}^{k}$ by a rational quantifier $\exists \geq s / t$, we denote the new logic by $\mathcal{L}_{\infty \omega}^{k}(\exists \geq s / t)$. The rational quantifiers apply to one formula and bind one free variable of that formula. The semantics of the new quantifier is defined in the following way.

$$
\mathbb{M} \models \exists \exists^{\geq s / t} x \phi(x) \Leftrightarrow(M,\{a \in M \mid(\mathbb{M}, a) \models \phi(x)\}) \in \exists^{\geq s / t} .
$$

We will show that the $\operatorname{logic} \mathcal{L}_{\infty \omega}^{k}\left(\exists^{\geq s / t}\right)$ has the zero-one law if and only if the quantifier $\exists \geq s / t$ is of the following form: $t \neq 2^{m}$ for all $m \in \mathbb{N}$, or $t=2^{m}$ for some $m \in \mathbb{N}$ and $k<m$.

## 3 Random Graphs

In this paper we concentrate on finite graphs. A graph $\mathbb{G}$ is a pair $(V, E)$, where $V$ is the set of vertices of $\mathbb{G}$ and $E$ is the edge relation, symmetric and irreflexive, over $V$. We study the symmetric case, where the atomic probability is $1 / 2$, which means there is an edge between two vertice with probability $1 / 2$. This leads to a uniform distribution of the graphs. Let $\mathcal{G}$ be the collection of all finite graphs. Let $[n]=\{1, \ldots, n\}$ and $\mathcal{G}_{n}$ the class of all graphs with $[n]$ as universe.

$$
\mathcal{G}_{n}=\{\mathbb{G} \in \mathcal{G}: G=[n]\} .
$$

Definition 5. Let probability $\mu_{n}: \mathcal{L} \rightarrow[0,1]$ be defined as follows.

$$
\mu_{n}(\phi)=\frac{\left|\left\{\mathbb{G} \in \mathcal{G}_{n}: \mathbb{G} \models \phi\right\}\right|}{\left|\mathcal{G}_{n}\right|} .
$$

We say that $\phi$ is true in random graph of cardinality $n$ with probability $\mu_{n}(\phi)$. We are interested in the asymptotic behavior of $\mu_{n}(\phi)$ as $n$ grows. Denote

$$
\mu(\phi)=\lim _{n \rightarrow \infty} \mu_{n}(\phi),
$$

if the limit exists. We say that $\phi$ is satisfied by almost all graphs if $\mu(\phi)=1$.

## 4 Monotone ( $k, \mathcal{Q}$ )-Pebble Game

Our result uses the monotone $(k, \mathcal{Q})$-pebble game introduced by Kolaitis and Väänänen (1995). We use it to characterizes the elementary equivalence between two graphs with respect to $\mathcal{L}_{\infty \omega}^{k}(\exists \geq s / t)$.

Definition 6. Assume $\mathbb{G}$ and $\mathbb{G}^{\prime}$ are graphs and $k$ a positive natural number. Let $m \leq k$ and vertices $v_{1}, \ldots, v_{m} \in G$ and $v_{1}^{\prime}, \ldots, v_{m}^{\prime} \in G^{\prime}$. We write

$$
\left(\mathbb{G}, v_{1}, \ldots, v_{m}\right) \equiv_{\mathcal{C}_{\infty}^{k}(\exists \geq s / t)}\left(\mathbb{G}^{\prime}, v_{1}^{\prime}, \ldots, v_{m}^{\prime}\right)
$$

if for every formula $\varphi\left(x_{1}, \ldots, x_{m}\right) \in \mathcal{L}_{\infty \omega}^{k}(\exists \geq s / t)$ the following holds

$$
\left(\mathbb{G}, v_{1}, \ldots, v_{m}\right) \models \varphi\left(x_{1}, \ldots, x_{m}\right) \Leftrightarrow\left(\mathbb{G}^{\prime}, v_{1}^{\prime}, \ldots, v_{m}^{\prime}\right) \models \varphi\left(x_{1}, \ldots, x_{m}\right) .
$$

We say that $\mathbb{G}$ is $\mathcal{L}_{\infty \omega}^{k}(\exists \geq s / t)$-equivalent with $\mathbb{G}^{\prime}$, denoted by

$$
\mathbb{G} \equiv \equiv_{\mathcal{L}_{\infty \omega}^{k}(\exists \geq s / t)} \mathbb{G}^{\prime}
$$

if the graphs $\mathbb{G}$ and $\mathbb{G}^{\prime}$ satisfy the same sentences of $\mathcal{L}_{\infty \omega}^{k}(\exists \geq s / t)$.
The monotone $(k, \mathcal{Q})$-pebble game is defined for a set $\mathcal{Q}$ of monotone simple unary quantifiers.

Definition 7. Let $\mathcal{Q}=\left\{Q_{i}: i \in I\right\}$ set of monotone simple unary generalized quantifiers and $k$ a natural number. The monotone $(k, \mathcal{Q})$-pebble game on structures $\mathbb{G}$ and $\mathbb{G}^{\prime}$ is played between two players, Eloise and Abelard. The game is played in turns. Abelard moves first. There are two possible moves for him.

1) The pebble move: Abelard chooses one of the structures and plays a pebble on an element of that structure. After this Eloise plays a pebble on an element of the other structure.
2) The quantifier move: Abelard chooses one of the structures, say $\mathbb{G}$, and a quantifier $Q_{i} \in \mathcal{Q}$ and a set $A \subseteq V^{G}$ such that $\left(V^{G}, A\right) \in Q_{i}$. Eloise responds by choosing a subset $B$ of vertices of the other structure $\mathbb{G}^{\prime}$, also such that $\left(V^{G^{\prime}}, B\right) \in Q_{i}$. After this Abelard plays a pebble on an element of $B$ and then Eloise plays a pebble on an element of $A$.

Game is played for $k$ rounds. Let $v_{i}$ be the vertice of $\mathbb{G}$ and $v_{i}^{\prime}$ vertice of $\mathbb{G}^{\prime}$ pebbled on the $i$ :th round. If the mapping $v_{i} \rightarrow v_{i}^{\prime}$ is not a partial isomorphism between $\mathbb{G}$ and $\mathbb{G}^{\prime}$ Abelard wins. Otherwise Abelard removes one pair of pebbles and the game resumes. Eloise wins the game if she can go on playing "forever".
The following theorem is also from (Kolaitis and Väänänen 1995).
Theorem 1. Let $\mathcal{Q}=\left\{Q_{i}: i \in I\right\}$ set of monotone simple unary generalized quantifiers, $\mathbb{G}$ and $\mathbb{G}^{\prime}$ graphs and $k \in \mathbb{N}$ constant. Then the following are equivalent:
(i) $\mathbb{G} \equiv \mathcal{L}_{\infty}^{k}(\mathcal{Q}) \mathbb{G}^{\prime}$.
(ii)Eloise has a winning strategy in the monotone $(k, \mathcal{Q})$-pebble game on graphs $\mathbb{G}$ and $\mathbb{G}^{\prime}$.

## 5 Strong Extension Axioms

Models of the $k$-extension axiom are all elementarily equivalent with respect to first order logic up to $k$ variables. In the case of $\mathcal{L}_{\infty \omega \omega}^{k}\left(\exists^{\geq s / t}\right)$ the normal extension axioms do not suffice to characterize one equivalence class of $\equiv_{\mathcal{L}_{\infty}^{k} \omega}(\exists \geq s / t)$. A stronger version of extension axioms is needed. Shelah (2000) introduced so called strong extension axioms, which are suitable for this purpose. We need the following concept of a $k$-type when defining the strong extension axioms.

Definition 8. A $k$-type $t\left(x_{1}, \ldots, x_{k}\right)$ over graphs is a maximal consistent set of formulas $E\left(x_{i}, x_{j}\right), \neg E\left(x_{i}, x_{j}\right)$ and identity and negated identity formulas with variables $\left\{x_{1}, \ldots, x_{k}\right\}$. A $k$-type $t$ is proper if it includes all the negated identity formulas. We denote the conjunction over formulas in $t\left(x_{1}, \ldots, x_{k}\right)$ by $\phi_{t}\left(x_{1}, \ldots, x_{k}\right)$.
Definition 9. A $k+1$-type $s\left(x_{1}, \ldots, x_{k+1}\right)$ extends $k$-type $t\left(x_{1}, \ldots, x_{k}\right)$, if $t \subseteq s$. A $k+1$-type $s\left(x_{1}, \ldots, x_{k+1}\right)$ extends $t\left(x_{1}, \ldots, x_{k}\right)$ properly, if $\left(x_{k+1} \neq x_{i}\right) \in s$ for all $i \leq k$.
Definition 10. Assume $\mathbb{G}$ is a graph and $\left(v_{1}, \ldots, v_{k}\right)$ a sequence of vertices of $\mathbb{G}$. We say that sequence $\left(v_{1}, \ldots, v_{k}\right)$ realizes the $k$-type $t\left(x_{1}, \ldots, x_{k}\right)$ in $\mathbb{G}$, if

$$
\left(\mathbb{G}, v_{1}, \ldots, v_{k}\right) \models \phi_{t}\left(x_{1}, \ldots, x_{k}\right) .
$$

We denote the type realized by $\left(v_{1}, \ldots, v_{k}\right)$ in $\mathbb{G}$ by $t_{\vec{v}}^{\mathbb{G}}$.
Definition 11. Assume $t\left(x_{1}, \ldots, x_{k}\right)$ is a proper $k$-type and $s\left(x_{1}, \ldots, x_{k+1}\right)$ a $k+1$-type properly extending $t\left(x_{1}, \ldots, x_{k}\right)$. For all $\alpha \in \mathbb{R}, 0<\alpha<1$, the strong extension axiom $S E A_{k}^{\alpha}(t, s)$ associated to a pair of types $(t, s)$ is the following sentence.

$$
\forall x_{1} \ldots \forall x_{k}\left(\phi_{t}\left(x_{1}, \ldots, x_{k}\right) \rightarrow \exists^{\geq \alpha / 2^{k}} x_{k+1} \phi_{s}\left(x_{1}, \ldots, x_{k}, x_{k+1}\right)\right) .
$$

The strong extension axiom $S E A_{k}^{\alpha}$ of graphs is a conjunction of $S E A_{k}^{\alpha}(t, s)$ over type pairs $(t, s)$.

When we fix $k$ vertices $\left(v_{1}, \ldots, v_{k}\right)$ of a graph $\mathbb{G}$, there is a natural partition of the remaining $n-k$ vertices into $2^{k}$ disjoint sets by means of which proper extensions of $t_{\bar{v}}^{\mathbb{G}}$ the vertices realize. Now the $k$-extension axiom says is that every set of the partition is nonempty. The strong extension axiom $S E A_{k}^{\alpha}$ says that every set of the partition contains almost the expected number of vertices, that is at least $\left|V^{G}\right| \cdot \alpha / 2^{k}$ vertices.

The following lemma is from (Blass and Gurevich 2000).
Lemma 1. Fix numbers $\beta, r$ in the open interval $(0,1)$. There is a constant $c \in(0,1)$ such that the following is true for every positive integer $m$. Let $X$ be the number of successes in $m$ independent trials, each having probability $r$ of success. Then probability $P(X \leq \beta m r) \leq c^{m}$.

The limit probability of the strong extension axiom $S E A_{k}^{\alpha}$ is calculated in (Blass and Gurevich 2000) for $\alpha=1 / 2$. They also state that limit can be calculated similarly for all $\alpha<1$.

Theorem 2. For all $k \in \mathbb{N}$ and for all $\alpha \in \mathbb{R}, 0<\alpha<1$, the limit probability $\mu\left(S E A_{k}^{\alpha}\right)=$ 1.

Proof. The strong extension axiom $S E A_{k}^{\alpha}$ fails in a random graph $\mathbb{G}$, if at least one of the axioms $S E A_{k}^{\alpha}\left(t, t^{\prime}\right)$ fails in $\mathbb{G}$. Let $n$ be the size of the graph and $\left(v_{1}, \ldots v_{k}\right)$ sequence of vertices of $\mathbb{G}$ such that $t_{\bar{v}}^{\mathbb{G}}=t$. There exists $n-k$ vertices in $\mathbb{G}$ that could extend $t$ to $t^{\prime}$. Let $X$ be a random variable that gives the number of vertices of $\mathbb{G}$ which extend $t_{\bar{v}}$ to $t^{\prime}$. We can apply the Lemma 1 . Let $\beta \in(0,1), \beta>\alpha$, such that for all large $n \in \mathbb{N}$ holds $\alpha n \leq \beta(n-k)$. Lemma 1 implies

$$
\begin{aligned}
P\left(X \leq \alpha n / 2^{k}\right) & \leq P\left(X<\beta(n-k) / 2^{k}\right) \\
& \leq c^{n-k} .
\end{aligned}
$$

There are at most $n^{k} k$-sequences $\left(v_{1}, \ldots, v_{k}\right)$ that could realize type $t$. Thus the probability that $S E A_{k}^{\alpha}\left(t, t^{\prime}\right)$ fails in $\mathbb{G}$ is at most

$$
n^{k} c^{n-k}
$$

It holds that $0<c<1$, so this bound tends to 0 as $n \rightarrow \infty$. Since the number of pairs of $\left(t, t^{\prime}\right)$ is finite for fixed $k$, also the probability that $S E A_{k}^{\alpha}$ fails tends to 0 .

## 6 Zero-One Law

Knyazev (1990) has studied probabilities of formulas of first order logic extended by rational quantifiers. Zero-one law of the logic $\mathcal{L}_{\infty \omega}^{\omega}(Q)$ is studied in (Kaila 2001) in a more general setting. We give a complete characterization for the zero-one law of $\mathcal{L}_{\infty \omega}^{k}\left(\exists^{s / t}\right)$. We establish a connection between the threshold of the quantifier and the number of variables allowed in the sentences.

Definition 12. A $\operatorname{logic} \mathcal{L}$ is said to have the zero-one law, if for all sentences $\phi \in \mathcal{L}, \mu(\phi)$ is defined and is either 0 or 1 .

We will show that strong extension axiom $S E A_{k-1}^{\alpha}$ gives a winning strategy for Eloise in the monotone ( $k, \exists \geq s / t$ )-pebble game if $t \neq 2^{m}$ for all $m \in \mathbb{N}$, or $t=2^{m}$ for some $m \in \mathbb{N}$ and $k \leq m$. For the logic $\mathcal{L}_{\infty \omega}^{m+1}\left(\exists^{\geq s / 2^{m}}\right)$ we can find a sentence that breaks the zero-one law. We define two characteristics for $\operatorname{logic} \mathcal{L}_{\infty \omega}^{k}(\exists \geq s / t)$. A lower bound $\kappa$ for the cardinality of the graphs and a lower bound $\lambda$ for the parameter $\alpha$ of strong extension axioms $S E A_{k}^{\alpha}$.

For every $k \in \mathbb{N}$ a proper $k$-type has $2^{k}$ different proper extensions.
Definition 13. Assume $k \in \mathbb{N}$. Let $p_{k}$ be the least natural number such that

$$
p_{k} / 2^{k}>s / t
$$

$p_{k}$ is the least number of proper extensions of a $k$-type we need to fill a set in $\exists \geq s / t$, when taking all the realizers of those extensions, assuming the elements are uniformly distributed between all the proper extensions.

When playing the monotone $\left(k, \exists^{\int / \sqcup}\right)$-pebble game, we want to be sure that the number of realizers of $p_{k}$ different $k$-types given us by $S E A_{k}^{\alpha}$ fills a set in the quantifier $\exists \geq s / t$. We also want to be sure that the realizers of any $\left(p_{k}-1\right)$ different types plus already played vertices can not fill a set in $\exists \geq s / t$. The following lemma shows that we can have that as long as we consider large enough graphs.

Lemma 2. Assume $k$ is a positive natural number, $s / t$ a rational, $0<s / t<1$ and $t \neq 2^{m}$ for all $m \in \mathbb{N}$, or $t=2^{m}$ and $k<m$. Then there is $\alpha_{k} \in \mathbb{R}, 0 \leq \alpha_{k}<1$, and $n_{k} \in \mathbb{N}$, such that the following two conditions hold whenever $\alpha \geq \alpha_{k}$ and $n \geq n_{k}$.

$$
\begin{gathered}
\alpha \cdot p_{k} / 2^{k} \geq s / t . \\
\alpha \cdot\left(p_{k}-1\right) n / 2^{k}+k+(1-\alpha) n<s / t \cdot n .
\end{gathered}
$$

Proof. Assume $t \neq 2^{m}$ for all $m \in \mathbb{N}$, or $t=2^{m}$ and $k<m$. Now it holds by Definition 13 and by the assumption on $t$ that $p_{k} / 2^{k}>s / t$, thus we can choose $\alpha_{k}$ such that $\alpha \cdot p_{k} / 2^{k} \geq$ $s / t$, for all $\alpha \geq \alpha_{k}$. Second condition is equivalent to

$$
\left(p_{k}-1\right) \alpha \cdot / 2^{k}+k / n+(1-\alpha)<s / t .
$$

We can see that the left side of the equation tends to $\left(p_{k}-1\right) / 2^{k}$, as $n$ tends to infinity and $\alpha$ tends to one. By Definition 13 and assumptions on $t$ it holds $\left(p_{k}-1\right) / 2^{k}<s / t$. Thus we can choose $n_{k}$ and $\alpha_{k}$ such that also the second condition holds for all $\alpha, \alpha_{k}<\alpha<1$ and $n \geq n_{k}$.

Definition 14. Assume $\exists \geq s / t$ is rational quantifier and $k \in \mathbb{N}$. Let $n_{i}$ and $\alpha_{i}$ be obtained from Lemma 2 for each $i \leq k$. Let

$$
\begin{aligned}
& \kappa=\max \left\{n_{i} \mid i \leq k\right\} . \\
& \lambda=\max \left\{\alpha_{i} \mid i \leq k\right\} .
\end{aligned}
$$

Eloise wins the monotone $\left(k+1, \exists^{\geq s / t}\right)$-pebble game on graphs $\mathbb{G}$ and $\mathbb{G}^{\prime}$, if graphs $\mathbb{G}$ and $\mathbb{G}^{\prime}$ are larger than $\kappa$ and they both satisfy the strong extension axiom $S E A_{k}^{\alpha}$ where $\alpha \geq \lambda$.

Theorem 3. Assume $t \neq 2^{m}$ for all $m \in \mathbb{N}$ or $t=2^{m}$ for some $m \in \mathbb{N}$ and $k<m$. Let $\mathbb{G}$ and $\mathbb{G}^{\prime}$ be graphs, $\left|V^{G}\right|,\left|V^{G^{\prime}}\right| \geq \kappa$, and suppose both satisfy $S E A_{k}^{\alpha}$ for some $\alpha>\lambda$. Then Eloise has a winning strategy in the monotone $\left(k+1, \exists^{\geq s / t}\right)$-pebble game on graphs $\mathbb{G}$ and $\mathbb{G}^{\prime}$.

Proof. Eloise can choose the first vertice arbitrarily since graph relation is irreflexive.
Assume that $l$ pebbles have been played, $l<k+1$, on vertices $v_{1}, \ldots, v_{l} \in G$ and $v_{1}^{\prime}, \ldots, v_{l}^{\prime} \in G^{\prime}$, and it holds $t_{\bar{v}}^{\mathbb{G}}=t_{\bar{v}^{\prime}}^{\mathbb{G}^{\prime}}$. We can assume that no more than one pebble is played on one element.

If Abelard plays the ordinary pebble move, $S E A_{k}^{\alpha}$ gives Eloise a good move. Assume Abelard plays the quantifier move and some set $A \subseteq V^{G}$, such that $\left(V^{G}, A\right) \in \exists \geq s / t$. By lemma $2 A$ consists of vertices that realize at least $p_{l}$ different proper extensions of $t_{\bar{v}}^{\mathbb{G}}$. The strategy for Eloise is to play a set $B \subseteq V^{G^{\prime}}$ such that $v^{\prime} \in B$ iff $t_{v^{\prime}, v^{\prime}}^{\mathbb{G}^{\prime}}=t_{\bar{v}, v}^{\mathbb{G}}$ for some $v \in A$. Now $B$ consists of vertices that realize at least $p_{l}$ different proper extensions of $t t_{v^{\prime}}^{\mathbb{G}^{\prime}}$ in $\mathbb{G}^{\prime}$. It follows from 2 that $|B| \geq s / t\left|V^{G^{\prime}}\right|$, thus $B$ is a legal move for Eloise. When Abelard chooses some $v^{\prime} \in B$, Eloise finds a corresponding vertices $v \in A$, such that $t_{\bar{v}^{\prime}, v^{\prime}}^{\mathbb{G}^{\prime}}=t_{\bar{v}, v}^{\mathbb{G}}$.

### 6.1 Sentences that Break the Zero-One Law

We show that the zero-one law does not hold for the logic $\mathcal{L}_{\infty \omega}^{m+1}\left(\exists \geq s / 2^{m}\right)$. For each logic $\mathcal{L}_{\infty \omega}^{m+1}\left(\exists \geq s / 2^{m}\right)$ there is a sentence that is true in almost all the graphs of cardinality divisible by $2^{m}$ and false in all the other graphs. Idea is that we can split the vertice set of a graph into $2^{m}$ disjoint subsets of equal size with probability tending to 1 when the cardinality of the graph is divisible by $2^{m}$. If the cardinality is not divisible by $2^{m}$, then the division is not possible. Thus the probability of this sentence will oscillate between zero and some probability converging to one.
Theorem 4. There is a sentences $\Phi_{s / 2^{m}} \in \mathcal{L}_{\infty \omega}^{m+1}\left(\exists{ }^{s / 2 m}\right)$ such that $\mu\left(\Phi_{s / 2^{m}}\right)$ does not converge to a limit.

Proof. Lets consider first the case where $s=1$. Assume $\mathbb{G}$ is a graph of size $n$ divisible by $2^{m}$. The idea is to first split the vertice set $V^{G}$ of a graph in half with one vertice. After that the both halves are split with two new vertice and so on. We want to split the vertice set into $2^{m}$ disjoint sets $T_{\bar{i}}$, such that for each $i \leq 2^{m}$ holds $\left|T_{\bar{i}}\right|=n / 2^{m}$. The parameter $\bar{i}=(i(1), \ldots, i(m))$ is a sequence of zeros and ones, which characterizes the set $T_{\bar{i}}$ with the corresponding characteristic formula $\phi_{\bar{i}}(\bar{x}, y)$.

$$
\phi_{\bar{i}}(\bar{x}, y)=\bigwedge_{i=1}^{m} \pm E y x_{i}
$$

in which $\pm E y x_{n}$ means $E y x_{n}$ if $i_{n}=1$ and $\neg E y x_{n}$ if $i_{n}=0$. The sentence gets the following form.

$$
\Phi_{1 / 2^{m}}=\exists x_{1}\left(\bigwedge_{i_{1}=0}^{1} \exists x_{2}\left(\ldots \bigwedge_{i_{m-1}=0}^{1} \exists x_{m}\left(\bigwedge_{i_{m}=0}^{1} \quad \exists^{\geq 1 / 2^{m}} y \phi_{\bar{i}}(\bar{x}, y)\right)\right) \ldots\right) .
$$

Sentence $\Phi_{1 / 2^{m}}$ says: " There is a partition of $V^{G}$ to $2^{m}$ sets $T_{\bar{i}}$ such that each set $T_{\bar{i}}$ contains at least $\left|V^{G}\right| / 2^{m}$ vertices"

We search for the splitting element in turn $i$ from the set $V^{G} \backslash\left\{v_{1}, \ldots v_{i-1}\right\}$, where $v_{j}$ is the splitting element used in turn $j, j \leq i$. There might be some edges that are fixed by the previous splittings, but asymptotically this does not affect the probability of an element to split a set. It holds that the probability for a vertice to split a set of size $n$ is larger than $\frac{1}{\sqrt{\pi n}}$ with large enough $n$, even if we have already some fixed number of edges. Probability $\mu_{n}\left(\phi_{1 / 2^{m}}\right)$ has the following approximation.

$$
\mu_{n}\left(\Phi_{1 / 2^{m}}\right) \geq \prod_{i \leq 2^{m}-1}\left(1-\left(1-\frac{1}{\sqrt{\pi n_{i}}}\right)^{n-i+1}\right)
$$

where $n_{i}$ is the cardinality of the set splitted in turn $i$. For each $i \leq 2^{m}$ it holds

$$
\lim _{n \rightarrow \infty}\left(1-\left(1-\frac{1}{\sqrt{\pi n_{i}}}\right)^{n-i+1}\right) \rightarrow 1
$$

Thus the limit probability $\mu\left(\Phi_{1 / 2^{m}}\right)$ does not exist, since the probability $\mu_{n}\left(\Phi_{1 / 2^{m}}\right)$ oscillates between 0 and some positive probability.

For quantifiers $\exists \geq^{\geq s / 2^{m}}, s \neq 1$, we construct a similar sentences with no limit probability. The idea is the same. We get the sentence $\Phi_{s / 2^{m}}$ by changing the characteristic formula $\phi_{\bar{i}}(\bar{x}, y)$ and by replacing the quantifiers $\exists \geq 1 / 2^{m}$ with $\exists \geq s / 2^{m}$ in $\Phi_{1 / 2^{m}}$. For every characteristic sequence $\bar{i}$ of $T_{\bar{i}}$ we define a new $\Psi_{\bar{i}}(\bar{x}, y)$, such that $\Psi_{\bar{i}}(\bar{x}, y)$ characterizes a set of size $n \cdot s / t$, which we denote by $T_{\Psi_{\bar{I}}}$.

We can represent the rational $s / 2^{m}$ the following way, where every $c(j), j \leq m$, is either 0 or 1 .

$$
s / t=c(1) \cdot 1 / 2^{1}+c(2) \cdot 1 / 2^{2}+\ldots+c(m) \cdot 1 / 2^{m} .
$$

For every sequence $\bar{i}$ and for every index $c(j)=1$, we define a new sequence $\overline{i_{j}}$ such that following conditions hold.

$$
\begin{aligned}
& i_{j}(l)=i(l) \text { when } l<j . \\
& i_{j}(j)=0 \Leftrightarrow i(j)=1 .
\end{aligned}
$$

We cut the sequence $\bar{i}$ after index $j$ and change the value $i(j)$ to 0 if it was originally 1 and vise versa and thus get a sequence $i_{j}$. This new sequence has a corresponding characteristic formula $\phi_{\overline{i_{j}}}(\bar{x}, y)$. It characterizes a subset of the vertice set $V^{G}$ of size $n / 2^{j}$. All the edited sequences $\bar{i}_{j}$ characterize together a subset of $V^{G}$ of size $n \cdot s / t$. The characteristic formula of $\Psi_{\bar{i}}(\bar{x}, y)$ for $\bar{i}$ is following

$$
\Psi_{\bar{i}}(\bar{x}, y)=\bigvee_{c(j)=1} \phi_{\overline{i_{j}}}(\bar{x}, y)
$$

Sentence $\Phi_{s / 2^{m}}$ gets the following form.

$$
\Phi_{s / 2^{m}}=\exists x_{1}\left(\bigwedge_{i_{1}=0}^{1} \exists x_{2}\left(\ldots \bigwedge_{i_{m-1}=0}^{1} \exists x_{m}\left(\bigwedge_{i_{m}=0}^{1} \quad \exists^{\geq s / t} y\left(\Psi_{\bar{i}}(\bar{x}, y)\right)\right) \ldots\right)\right)
$$

Claim: $\mu\left(\phi_{s / 2^{m}}\right)$ does not exist.

Proof. We show that $\models \Phi_{1 / 2^{m}} \rightarrow \Psi_{s / 2^{m}}$.
Assume $\mathbb{G}$ a graph such that $\mathbb{G} \models \Phi_{1 / 2^{m}}$. One can observe from the construction of $\Psi_{\bar{i}}$ that $T_{\Psi_{\bar{i}}}$ is a union of $s$ different $T_{\bar{i}}$. Since all $T_{\bar{i}}$ are disjoint, it holds $\left|T_{\Psi_{\bar{i}}}\right|=s \cdot n / 2^{m}$. Thus $\mathbb{G} \models \Phi_{s / 2^{m}}$.

Next we show that $\Phi_{s / t}$ fails in all graphs of cardinality not divisible by $2^{m}$. Assume $\mathbb{G}$ is a graph of cardinality $n, n$ not divisible by $2^{m}$ and $\mathbb{G} \models \Phi_{s / 2^{m}}$. Since $T_{\Psi_{\bar{i}}}$ is a union of $s$ different $T_{\bar{i}}$ and each $T_{\bar{i}}$ is a subset of exactly $s$ different $T_{\Psi_{\bar{i}}}$ we get $\sum_{\bar{i}}\left|T_{\Psi_{\bar{i}}}\right|=s \cdot n$. Since $\mathbb{G}$ satisfies $\Phi_{s / 2^{m}}$ and $n$ is not divisible by $2^{m}$ it holds that $\left|T_{\Psi_{\bar{i}}}\right| \geq\left\ulcorner s / 2^{m} \cdot n\right\urcorner>s / 2^{m} \cdot n$. Thus we get $\sum_{\bar{i}}\left|T_{\Psi_{\bar{i}}}\right|>s \cdot n$, which is a contradiction.

We have shown that for graphs of cardinality divisible by $2^{m}$, sentence $\Phi_{s / 2^{m}}$ gets the same limit value as $\Phi_{1 / 2^{m}}$, that is 1 . For graphs of cardinality not divisible by $2^{m}$, sentence $\Phi_{s / 2^{m}}$ fails. Thus $\mu\left(\Phi_{s / 2^{m}}\right)$ is not defined.

### 6.2 Conclusion

We see that the zero-one law for the logic $\mathcal{L}_{\infty \omega \omega}^{\omega}\left(\exists^{\geq s / t}\right)$ depends only on the denominator $t$ of threshold of the quantifier.

Theorem 5. Assume $t \neq 2^{m}$ for all $m \in \mathbb{N}$. The zero-one law holds for logic $\mathcal{L}_{\infty \omega}^{k}(\exists \geq s / t)$ for all $k \in \mathbb{N}$.

Proof. Let $\phi \in \mathcal{L}_{\infty \omega}^{k}(\exists \geq s / t)$. Assume $\mathbb{G}$ is a graph such that $\mathbb{G} \models S E A_{k-1}^{\alpha}$ and $\mathbb{G} \models \phi$. Then by Theorem 1 and 3 every $\mathbb{G}^{\prime}$ that satisfies $S E A_{k-1}^{\alpha} \quad$ also satisfies $\phi$. Thus by 2 $\mu(\phi)=\mu\left(S E A_{k-1}^{\alpha}\right)=1$.

Assume $\mathbb{G} \not \models \phi$. Then $\mathbb{G} \models \neg \phi$. By Theorems 1 and 3 every $\mathbb{G}^{\prime}$ that satisfies $S E A_{k-1}^{\alpha}$ also satisfies $\neg \phi$. Thus by $2 \mu(\neg \phi)=\mu\left(S E A_{k-1}^{\alpha}\right)=1 \leftrightarrow \mu(\phi)=0$.

Theorem 6. Assume $t=2^{m}$ for some $m \in \mathbb{N}$. The zero-one law holds for the logic $\mathcal{L}_{\infty \omega \omega}^{k}\left(\exists^{\geq s / t}\right)$, when $k \leq m$ and does not hold when $k>m$.

Proof. Assume $k \leq m$. Let $\phi \in \mathcal{L}_{\infty \omega}^{k}(\exists \geq s / t)$. Assume $\mathbb{G}$ is a graph such that $\mathbb{G} \models S E A_{k-1}^{\alpha}$ and $\mathbb{G} \models \phi$. Then by Theorems 1 and 3 every $\mathbb{G}^{\prime}$ that satisfies $S E A_{k-1}^{\alpha} \quad$ also satisfies $\phi$. Thus by $2 \mu(\phi)=\mu\left(S E A_{k-1}^{\alpha}\right)=1$.

Assume $\mathbb{G} \not \vDash \phi$. Then $\mathbb{G} \models \neg \phi$. By Theorems 1 and 3 every $\mathbb{G}^{\prime}$ that satisfies $S E A_{k-1}^{\alpha}$ also satisfies $\neg \phi$. Then by $2 \mu(\neg \phi)=\mu\left(S E A_{k-1}^{\alpha}\right)=1 \leftrightarrow \mu(\phi)=0$.

Assume $k>m$. Sentence $\Phi_{s / 2^{m}} \in \mathcal{L}_{\infty \omega}^{k}(\exists \geq s / t)$ and $\mu\left(\Phi_{s / 2^{m}}\right)$ is not defined.
(2) Consider quantifiers $\exists^{\geq 1 / 3}$ and $\exists^{\geq 1 / 2}$. Logic $\mathcal{L}_{\infty \omega}^{k}(\exists \geq 1 / 3)$ has the zero-one law for every $k \in \mathbb{N}$. Thus also $\mathcal{L}_{\infty \omega}^{\omega}\left(\exists^{\geq 1 / 3}\right)$ has the zero-one law. Therefore we know that every property of graphs with limit probability different to 0 or 1 can not be defined in $\mathcal{L}_{\infty \omega}^{\omega}(\exists \geq 1 / 3)$. For example even cardinality.

Logic $\mathcal{L}_{\infty \omega}^{k}\left(\exists^{\geq 1 / 2}\right)$ has the zero-one law if $k=1$. With two variables, we can write the sentence $\Phi_{1 / 2}$, which separates almost all graphs of even cardinality from the graphs of odd cardinality.

$$
\Phi_{1 / 2}=\exists x\left(\exists^{\geq 1 / 2} y(E x y) \wedge \exists^{\geq 1 / 2} y(\neg E x y)\right) .
$$

## Acknowledgements

I would like to thank Professor Jouko Väänänen, who is the supervisor of my doctoral studies. This research was supported by the Finnish Academy of Science and Letters, Vilho, Yrjö and Kalle Väisälä Foundation.

## References

A. Blass, Y. Gurevich (2000). New Zero-One Law and Strong Extension Axioms. Bulletin of the European Association for Theoretical Computer Science, 72, 103-122.
R. Fagin (1976). Probabilities on finite models. J. Symbolic Logic, 41, (1), 50-58.

Yu.V. Glebskii, D.I. Kogan, M.I. Liogon'kii and V.A. Talanov (1969). Range and degree of realizability of formulas in the restricted predicate calculus. Kibernetika, 5, 17-28.
R. Kaila (2001). On Probabilistic elimination of generalized quantifiers. Random Structures and Algorithms, 19, 1-36.

Ph.G. Kolaitis, J.A. Väänänen (1995). Generalized quantifiers and pebble games on finite structures. Annals of Pure and Applied Logic, 74, 23-75.

Ph.G. Kolaitis and M.Y. Vardi (1990). 0-1 law for infinitary logics. Proc. 5th IEEE Symp. on Logic in Computer Science, 156-167.
V.V. Knyazev (1990). Zero-one law for an extension of first order predicate language. Kibernetika, 2, 110-113.
P. Lindström (1966). First Order Predicate Logic with generalized quantifiers. Theoria, 32, 186-195.
A. Mostowski (1957). On a generalization of quantifiers. Fundam. Math. 44, 12-36.
S. Shelah (2000) Choiceless polynomial time logic: inability to express. Computer Science Logic, Springer Lecture Notes in Computer Science 1862, 72-125.
A. Tarski (1961). A model theoretic result concerning infinitary logic. Notices Amer. Math Soc. 8, 260.

# Hybrid Logics with Concrete Domains 

Sergio Mera<br>Computer Science Department, FCEyN, University of Buenos Aires, Argentina<br>smera@dc.uba.ar


#### Abstract

In this paper we present the hybrid logic $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$, an extension of $\mathcal{H} \mathcal{L}(@, \downarrow)$, whose models have a concrete domain (such as the natural or real numbers). This logic extends the language of $\mathcal{H} \mathcal{L}(@, \downarrow)$ including terms with equality to deal with concrete domain values. Similar languages have already been investigated in other areas like knowledge representation (e.g., description logics with concrete domains (Baader and Hanschke 1991)) and languages for verification (e.g., half-order logic (Alur and Henzinger 1990)). Our main result is a sound and complete axiomatization for $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$. We also present an embedding of description logics with concrete domains and half-order logic within our framework.


## 1 Introduction and Motivation

The hybrid logic $\mathcal{H} \mathcal{L}(@, \downarrow)$ has been extensively investigated in recent years. This logic is the result of enriching standard modal logic (Blackburn, de Rijke, and Venema 2001) with nominals (typically written $i, j, k, \ldots$ ) and the @ and $\downarrow$ operators. The fact that each nominal is assumed to be true at a unique state in every model implies that nominals in effect name states. When nominals have been added to the language, it is quite natural to look for a way to evaluate a formula at a named state, and that is the purpose of the @ operator. Informally speaking, the formula $@_{i} \phi$ (read 'at $i, \phi$ ') moves the point of evaluation to the state named by $i$ and checks whether $\phi$ is true there. The next natural step is to think of nominals not as names, but as variables over individual states, and to add quantifiers. The classical first order notion of quantifiers (like $\forall$ and $\exists$ ) does not reflect the intrinsically local behavior of modal logic, in which the evaluation of formulas takes place at a given point. This is the main motivation to introduce the $\downarrow$ binder, that enable us to create a name "on the fly" for the current state of evaluation and let us refer to it later in the formula. That is, when evaluating $\downarrow x . \phi$ in a state $w$, the variable $x$ will act in $\phi$ as a nominal that names $w$. Further details about hybrid logics can be found in (Areces, Blackburn, and Marx 2001; ten Cate 2005).

Generally speaking, $\mathcal{H} \mathcal{L}(@, \downarrow)$ permits the proper treatment of qualitative properties. An example from the system verification area could be the following. We will consider a "safe bank" to be a bank who keeps its safety box locked after the alarm has rang. We can express this property by saying that after the system has reached the alarm state, all
the following states must have the property safe-locked:

$$
[R] \downarrow x .\left(@_{\text {alarm }}\langle R\rangle x \rightarrow \text { safe-locked }\right) .
$$

In this example we are interpreting the standard modalities in a temporal way. The operator $[R]$ means "for every future state" and $\langle R\rangle$ "for some future state". The $\downarrow$ binder enable us to name every future state, and together with the @ operator, we can "jump" to the alarm state and use $\langle R\rangle$ to identify its successors and force them to have the safe-locked property. ${ }^{1}$

It is clear that we are abstracting over the actual time at which events occur, and we do not model quantitative information. We might want to state that, after we have reached the alarm state, the safety box should be locked within 5 time units. Such kind of properties are typical in critical real-time systems (e.g., protocols, embedded systems).

In this paper we present an extension of $\mathcal{H} \mathcal{L}(@, \downarrow)$, called $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$, that adds a concrete domain (that is, a set of concrete values, such as the natural numbers) to the standard "abstract" domain representing the states of the system. Each state in the model can be related to one or more values through concrete functions that take states to concrete values. We also extend the language adding terms with equality and $n$-ary predicates to deal with concrete values. Returning to the bank example, we can now provide a finer grained requirement that models the timing constraint as follows

$$
[R] \downarrow x .\left(@_{\text {alarm }}\langle R\rangle x \wedge(\text { time }(x)>\operatorname{time}(\text { alarm })+5) \rightarrow \text { safe-locked }\right) .
$$

If we look at this example, we can see that we are identifying the successors of the alarm state in the same way the previous example did (by means of the @ ${ }_{\text {alarm }}\langle R\rangle x$ formula), but now we can impose more expressive conditions on the states. We are using the concrete function time to retrieve the time associated to each state, and the 2 -ary predicate $>$ to identify the successors that are 5 time units after the alarm state.

The idea of adding concrete domains to $\mathcal{H} \mathcal{L}(@, \downarrow)$ is inspired by similar languages in other areas. We will discuss in detail the connections with Description Logics (Baader, Calvanese, McGuinness, Nardi, and Patel-Schneider 2003) and Half-order Logics (Alur and Henzinger 1990) in Sections 4 and 5 respectively. We will see that $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$ can be taken as a unifying framework for the two logical formalisms mentioned above.

## 2 Syntax and Semantics of $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$

Definition 1 (Syntax). Let ReL $=\left\{R_{1}, R_{2}, \ldots\right\}$ (the relational symbols), PRED $=\left\{p_{1}, p_{2}, \ldots\right\}$ (the predicate symbols), $\mathrm{FUN}=\left\{f_{1}, f_{2}, \ldots\right\}$ (the functional symbols), $\mathrm{CON}=\left\{h_{1}, h_{2}, \ldots\right\}$ (the concrete functional symbols), NOM $=\left\{i_{1}, i_{2}, \ldots\right\}$ (the nominal symbols) and VAR $=$ $\left\{x_{1}, x_{2}, \ldots\right\}$ (the variable symbols) be pairwise disjoint, countable infinite sets of symbols.

[^30]The set $\operatorname{SSYM}=$ NOM $\cup$ VAR is called the set of state symbols. 0 -ary predicate symbols are called propositions. The sets TERMS of terms and FORMS of formulas of $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$ in the signature $\langle$ REL, PRED, FUN, CON, NOM, VAR〉 are defined as

$$
\begin{gathered}
\text { TERMS }:=h\left(s_{1}, \ldots, s_{n}\right) \mid f\left(t_{1}, \ldots, t_{n}\right) \\
\text { FORMS }:=s\left|p\left(t_{1}, \ldots, t_{n}\right)\right| t_{1}=t_{2}|\neg \phi| \phi \wedge \psi|\langle R\rangle \phi| @_{s} \phi \mid \downarrow x . \phi .
\end{gathered}
$$

where $s, s_{1} \ldots s_{n} \in \operatorname{SSYM}, p \in \operatorname{PRED}$ (an $n$-place predicate symbol), $f \in$ FUN (an $n$-place functional symbol), $h \in \operatorname{CON}$ (an $n$-place concrete functional symbol), $R \in \operatorname{REL}, \phi, \psi \in$ FORMS, $t_{1}, \ldots, t_{n} \in$ TERMS and $x \in \operatorname{VAR}$. We define the operator $[R]$ as $[R] \phi:=\neg\langle R\rangle \neg \phi$. We call formulas of the form $s, p\left(t_{1}, \ldots, t_{n}\right)$ and $t_{1}=t_{2}$ atomic formulas.

As we mentioned above, models of $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$ extend hybrid models with a concrete domain of values, and the valuation should let us interpret terms:

Definition 2 (Models). $\mathcal{H} \mathcal{L}_{\mathcal{L}}(@, \downarrow)$ is interpreted over structures of the form $\mathcal{M}=\left\langle W, U,\left\{R_{i}\right\}\right.$, $\llbracket \rrbracket\rangle$, where $W$ is a nonempty set of states, $U$ is a set of values and $\left\{R_{i}\right\}$ is a set of binary relations on $W$. The valuation function assigns meaning to elements in CON, FUN, PRED and NOM, and we can think of it as different functions, one for each type of non-logical symbol. $\llbracket \rrbracket_{\text {CoN }}$ : CON $\times W^{n} \rightarrow U$ assigns a partial function to each $n$-ary concrete functional symbol, $\llbracket \rrbracket_{\mathrm{FUN}}:$ FUN $\times U^{n} \rightarrow U$ assigns a partial function to each $n$-ary functional symbol, $\llbracket \rrbracket_{\text {PRED }}:$ PRED $\times W \rightarrow \wp\left(U^{n}\right)$ is an interpretation function for $n$-ary predicate symbols ${ }^{2}$, and $\llbracket \rrbracket_{\text {мом }}:$ NOM $\times W \rightarrow\{$ true, false $\}$ is an interpretation function for nominals, with the requirement that $\llbracket i \rrbracket_{\text {Nом }}(w)$ must be true for exactly one $w \in W$. Given a model $\mathcal{M}=\left\langle W, U,\left\{R_{i}\right\}, \llbracket \rrbracket\right\rangle$ we call $\mathcal{F}=\left\langle W, U,\left\{R_{i}\right\}\right\rangle$ its frame.

In the definition above, we chose a state-dependent (flexible) interpretation for predicates. This definition is more general than a state-independent (rigid) one, but there is also another motivation. Let's go back to our example. We can call a bank "successful" if the difference between the profit of one year and the next is sufficiently high

$$
[R] \downarrow x .([R] \downarrow y .(\operatorname{year}(y)-\text { year }(x))=1 \rightarrow \operatorname{enough}(\operatorname{profit}(y)-\operatorname{profit}(x))) .
$$

But what does enough mean? Clearly, the criteria used nowadays to define "enough" is different to the one used, say, 50 years ago. Therefore, we would like the truth value of enough $(x)$ to depend on the state of evaluation. In any case, we will see in Section 4 that we can impose rigidity on predicates by adding an extra axiom to $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$.

We also took the decision of working with partial functions. Returning to the bank example, we can imagine that there is a generic error state, representing an unexpected failure of the system. It is quite natural to think that the concrete function time we used before could be undefined for the error state.

We are now ready to define the satisfiability relation for $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$. An assignment $g$ for $\mathcal{M}$ is a mapping $g:$ VAR $\rightarrow W$. Given an assignment $g$, we define $g_{w}^{x}$ (an $x$-variant of

[^31]$g)$ by $g_{w}^{x}(x)=w$ and $g_{w}^{x}(y)=g(y)$ for $y \neq x$. We start by extending the valuation $\llbracket \rrbracket$ to interpret arbitrary terms. Given a model $\mathcal{M}$ and an assignment $g$ we define
\[

$$
\begin{aligned}
\llbracket s \rrbracket^{\mathcal{M}, g} & =g(s) \quad s \in \operatorname{VAR} \\
\llbracket s \rrbracket^{\mathcal{M}, g} & =w \quad s \in \operatorname{NOM} \text { and } w \text { is such that } \llbracket s \rrbracket_{\text {NOM }}(w) \text { is true } \\
\llbracket h\left(s_{1}, \ldots, s_{n}\right) \rrbracket^{\mathcal{M}, g} & =\llbracket h \rrbracket_{\mathrm{CON}}\left(\llbracket s_{1} \rrbracket^{\mathcal{M}, g}, \ldots, \llbracket s_{n} \rrbracket^{\mathcal{M}, g}\right) \quad s_{i} \in \operatorname{SSYM}, h \in \mathrm{CON} \\
\llbracket f\left(t_{1}, \ldots, t_{n}\right) \rrbracket^{\mathcal{M}, g} & =\llbracket f \rrbracket_{\mathrm{PUN}}\left(\llbracket t_{1} \rrbracket^{\mathcal{M}, g}, \ldots, \llbracket t_{n} \rrbracket^{\mathcal{M}, g}\right) \quad t_{i} \in \mathrm{TERMS}, f \in \mathrm{FUN} .
\end{aligned}
$$
\]

Notice that $\llbracket h\left(s_{1}, \ldots, s_{n}\right) \rrbracket^{\mathcal{M}, g}$ and $\llbracket f\left(t_{1}, \ldots, t_{n}\right) \rrbracket^{\mathcal{M}, g}$ might be undefined as $\llbracket h \rrbracket_{\text {CoN }}$ and $\llbracket f \rrbracket_{\mathrm{FUN}}$ are partial functions.

Definition 3 (Satisfiability). Given a model $\mathcal{M}=\left\langle W, U,\left\{R_{i}\right\}, \llbracket \rrbracket\right\rangle$, an assignment $g$ for $\mathcal{M}$ and a state $w \in W$, the satisfiability relationship is defined as

```
\(\mathcal{M}, g, w \models s\)
iff \(\llbracket s \rrbracket^{\mathcal{M}, g}=w, s \in \operatorname{SSYM}\)
\(\mathcal{M}, g, w=p\left(t_{1}, \ldots, t_{n}\right)\)
iff \(\exists \bar{u} \in U^{n}: \bar{u}=\left(\llbracket t_{1} \rrbracket^{\mathcal{M}, g}, \ldots, \llbracket t_{n} \rrbracket^{\mathcal{M}, g}\right)\) and \(\bar{u} \in \llbracket p \rrbracket_{\text {PRED }}(w)\)
\(\mathcal{M}, g, w=t_{1}=t_{2} \quad\) iff \(\exists u_{1}, u_{2} \in U: u_{1}=\llbracket t_{1} \rrbracket^{\mathcal{M}, g}\) and \(u_{2}=\llbracket t_{2} \rrbracket^{\mathcal{M}, g}\) and \(u_{1}=u_{2}{ }^{3}\)
\(\mathcal{M}, g, w \models \neg \phi \quad\) iff \(\mathcal{M}, g, w \not \models \phi\)
\(\mathcal{M}, g, w \models \phi_{1} \wedge \phi_{2} \quad\) iff \(\mathcal{M}, g, w \models \phi_{1}\) and \(\mathcal{M}, g, w \models \phi_{2}\)
\(\mathcal{M}, g, w=\langle R\rangle \phi \quad\) iff \(\quad \exists w^{\prime} \in W: R\left(w, w^{\prime}\right)\) and \(\mathcal{M}, g, w^{\prime} \models \phi\)
\(\mathcal{M}, g, w=@_{s} \phi \quad\) iff \(\quad \mathcal{M}, g, \llbracket s \rrbracket^{\mathcal{M}, g}=\phi\)
\(\mathcal{M}, g, w \models \downarrow x . \phi \quad\) iff \(\quad \mathcal{M}, g_{w}^{x}, w \models \phi\)
```

We say that $\phi$ is valid on a model $\mathcal{M}$ iff for all assignments $g$ on $\mathcal{M}$, and all states $w, \mathcal{M}, g, w \models \phi$, and we write $\mathcal{M} \vDash \phi$. We say that a formula $\phi$ is valid on a frame $\mathcal{F}=\left\langle W, U,\left\{R_{i}\right\}\right\rangle$ (written $\mathcal{F} \models \phi$ ) iff for all valuations $\llbracket \rrbracket, \phi$ is valid on $\langle\mathcal{F}, \llbracket \rrbracket\rangle$.

## 3 Axiomatization

The axioms shown in Figure 1.1 (an extension of the axiomatization for $\mathcal{H} \mathcal{L}(@, \downarrow)$ given in (Blackburn and ten Cate 2004)) are sound and complete for the class of all models of $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$. In the axioms, the expression $\phi\left\langle t_{1}:=t_{2}\right\rangle\left(\phi\left[t_{1}:=t_{2}\right]\right)$ denotes a formula that results from $\phi$ by safely replacing zero, one, or more (all, respectively) free occurrences of $t_{1}$ by $t_{2} .{ }^{4} \phi, \psi$ range over arbitrary formulas, $s, r$ over SSYM, $x$ over VAR and $t, t_{1}, t_{2}$ over arbitrary terms. The expression $\phi[t]$ means that $t$ is a subterm of $\phi$.

Given a set $\Gamma \cup\{\phi\}$ of formulas, we define the syntactic entailment $\Gamma \vdash \phi$ in the usual way. We say that a set of formulas $\Gamma$ is consistent if $\Gamma \nvdash \perp$.

Some intuitive comments about the axiomatization. Comparing with the axiomatization given in (Blackburn and ten Cate 2004), the new axioms are Term, TEQ and Def. Term relates state equality with term equality, $T E Q$ provides congruence at the term level

[^32]
## Axioms:

| $C T$ | All classical tautologies | $K_{\square}$ | $\vdash[R](\phi \rightarrow \psi) \rightarrow[R] \phi \rightarrow[R] \psi$ |
| :---: | :---: | :---: | :---: |
| $K_{@}$ | $\vdash @_{s}(\phi \rightarrow \psi) \rightarrow @_{s} \phi \rightarrow @_{s} \psi$ | Selfdual@ | $\vdash @_{s} \phi \leftrightarrow \neg @_{s} \neg \phi$ |
| Ref@ | $\vdash @_{s} s$ | Agree | $\vdash @_{s} @_{r} \phi \leftrightarrow @_{r} \phi$ |
| Intro | $\vdash s \rightarrow\left(\phi \leftrightarrow @_{s} \phi\right)$ | Back | $\vdash\langle R\rangle @_{s} \phi \rightarrow @_{s} \phi$ |
| DA | $\vdash @_{s}(\downarrow x . \phi \leftrightarrow \phi[x:=s])$ | $B G_{\downarrow}$ | $\vdash @_{s}[R] \downarrow$. $@_{s}\langle R\rangle x$ |
| Name $\downarrow$ | $\vdash \downarrow x .(x \rightarrow \phi) \rightarrow \phi$ provided | Term | $\vdash @_{s} r \rightarrow(t=t \rightarrow t[s:=r]=t)$ |
|  | that $x$ does not occur in $\phi$ | TEQ | $\vdash t_{1}=t_{2} \rightarrow\left(\phi \rightarrow \phi\left\langle t_{1}:=t_{2}\right\rangle\right)$ |
|  |  | Def | $\vdash \phi[t] \rightarrow t=t$ where $\phi$ is atomic |
| Rules: |  |  |  |
| MP | If $\vdash \phi$ and $\vdash \phi \rightarrow \psi$ then $\vdash \psi$ | Gen@ | If $\vdash \phi$ then $\vdash @_{i} \phi$ |
| $G e n_{\downarrow}$ | If $\vdash \phi$ then $\vdash \downarrow$ s. $\phi$ | $G e n_{\square}$ | If $\vdash \phi$ then $\vdash[R] \phi$ |

Figure 1.1: Axiomatization for $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$.
and Def assures that a valid formula must have all its terms defined. Checking soundness for this new axioms is trivial. We now give some details on the completeness proof. We start by introducing maximal consistent sets.

Definition 4 (MCSs). A set of formulas $\Gamma$ is maximal consistent (we will say that $\Gamma$ is an MCS) iff $\Gamma$ is consistent, and any set of formulas properly containing $\Gamma$ is inconsistent. We say that an MSC $\Gamma$ is pasted iff $@_{i} \diamond \phi \in \Gamma$ implies that for some nominal $j \in$ NOM, $@_{i} \diamond j \wedge @_{j} \phi \in \Gamma$. An mCS $\Gamma$ is var-saturated iff for all $x \in \operatorname{VAR}$ there is $i \in$ NOM such that $@_{i} x \in \Gamma$. Finally, an MCS $\Gamma$ is named iff there is a nominal $j \in$ NOM such that $j \in \Gamma$.

We can now provide an extension of the standard Lindenbaum Lemma that will ensure that MCSS satisfy the additional properties mentioned before.

Lemma 1 (Extended Lindenbaum Lemma). Consider the language of $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$ in the signature $\left\langle\right.$ REL, PRED, FUN, CON, NOM, VAR〉. Let $\mathrm{NOM}^{\prime}$ be a countably infinite collection of nominals disjoint from NOM, and let $S^{\prime}$ be the signature obtained by adding these new nominals to $S$. Then every consistent set of formulas over $S$ can be extended to a named, pasted and var-saturated MCS over $S^{\prime}$.

Proof. We use the standard construction in hybrid logic (see Chapter 7 of (Blackburn, de Rijke, and Venema 2001) for details), but we also need MCSs to be var-saturated. Split $\mathrm{NOM}^{\prime}$ in two disjoint, ordered, infinite sets $\mathrm{NOM}_{v a r}^{\prime}$ and $\mathrm{NOM}_{n o m}^{\prime}$. Given a consistent set of formulas $\Sigma$, define $\Sigma_{k}$ to be $\Sigma \cup\{k\} \cup\left\{@_{j_{x}} x \mid x \in \operatorname{VAR}\right\}$ where $k$ is the first nominal in $\mathrm{NOM}_{\text {nom }}^{\prime}$ and $j_{x}$ is a nominal in $\mathrm{NOM}_{\text {var }}^{\prime}$ such that for each $x, y \in \operatorname{VAR}, x \neq y$ implies $j_{x} \neq j_{y}$. $\Sigma_{k}$ is consistent, for suppose not. Then, for some finite conjunction of formulas $\theta$ from $\Sigma$, and some finite conjunction $\psi$ of formulas from $\left(\Sigma_{k}-\Sigma\right), \vdash \psi \rightarrow \neg \theta$. We can take $\psi=@_{j_{x_{1}}} x_{1} \wedge \ldots \wedge @_{j_{x_{n}}} x_{n} \wedge k$, for $\left\{@_{j_{x_{1}}} x_{1}, \ldots, @_{j_{x_{n}} x_{n}}\right\}$ a finite subset of the new identities. Now, by $R e f_{@}$ we eliminate the identities and by $N a m e_{\downarrow}$ we conclude $\vdash \neg \theta$ a contradiction.

Following the standard hybrid procedure, we can expand $\Sigma$ to $\Sigma^{+}$, a maximal consistent set that is named (by $k$ ) and pasted. Furthermore, it is var-saturated because we explicitly added $@_{i_{x}} x$ for each variable $x \in$ VAR.

Now we are ready to build a canonical model, but as this language has terms, we will need to define a proper equivalence relation to construct the canonical set of values.

Definition 5. Let $\Gamma$ be a named, pasted and var-saturated mCs. We say that a term $t$ is $\Gamma$-defined when $t=t \in \Gamma$. We define $\sim^{\Gamma}$ over $\Gamma$-defined terms as $t_{1} \sim^{\Gamma} t_{2}$ iff $t_{1}=t_{2} \in \Gamma$.

It is not difficult to see that $\sim^{\Gamma}$ is an equivalence relation. The fact that its elements are $\Gamma$-defined terms, together with the TEQ axiom, implies its reflexivity, symmetry and transitivity. We will write $[t]$ for the equivalence class of $t$ in $\sim^{\Gamma}$.

Now we show how to build a canonical model for $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$.
Definition 6 (Canonical Model). Let $\Gamma$ be a named, pasted and var-saturated MCS. The canonical model yielded by $\Gamma$ is $\mathcal{M}^{\Gamma}=\left\langle W^{\Gamma}, U^{\Gamma},\left\{R_{i}^{\Gamma}\right\}, \llbracket \rrbracket^{\Gamma}\right\rangle$. $W^{\Gamma}$ is the set of all named sets yielded by $\Gamma$, where $\Delta_{i}$ is a set yielded by $\Gamma$ iff $\Delta_{i}=\left\{\phi \mid @_{i} \phi \in \Gamma\right\}$ for some $i \in$ nOM. The set of canonical values $U^{\Gamma}$ is the set of equivalent classes of $\sim^{\Gamma}$. The canonical set of binary relations on $W$ is defined as: $R_{i}^{\Gamma} u v$ iff for all formula $\phi \in v,\left\langle R_{i}\right\rangle \phi \in u$. The canonical interpretations for concrete and abstract functions are

Canonical interpretations for nominals, predicates, and the assignment function are:

$$
\left.\begin{array}{c}
\llbracket i \rrbracket_{\text {NOM }}^{\Gamma}(w)=\text { true } \\
\left.\left(\left[t_{1}\right], \ldots,\left[t_{n}\right]\right) \in \llbracket p\right]_{\text {PRED }}^{\Gamma}(w) \\
\text { iff } \\
g^{\Gamma}(x)=w
\end{array} \quad \text { iff } \quad x \in w, \ldots, t_{n}\right) \in w, \text { for } i \text { a nominal. } p \text { a predicate symbol. } .
$$

We should prove that the construction showed above is actually a $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$ model. Let us recall first some properties of named sets.

Lemma 2 (Blackburn, de Rijke, and Venema 2001). Let $\Gamma$ be a MCS and let $\Delta_{i}$ and $\Delta_{j}$ be mCS yielded by $\Gamma$. Then, (i) $i \in \Delta_{i}$, (ii) if $i \in \Delta_{j}$, then $\Delta_{i}=\Delta_{j}$, (iii) @ ${ }_{i} \phi \in \Delta_{j}$ iff $@_{i} \phi \in \Gamma$, and (iv) if $i \in \Gamma$ then $\Gamma=\Delta_{i}$.

Proposition 1. The canonical model is a model for $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$.
Proof. By Lemma 2, $W^{\Gamma}$ is a nonempty set of worlds.
We show that $\llbracket \rrbracket^{\Gamma}$ is well defined. For $\llbracket \rrbracket_{\text {cov }}^{\Gamma}$, fix $w_{1}, \ldots, w_{n} \in W^{n}$. Then there must be a unique term assigned by $\llbracket h\left(w_{1}, \ldots, w_{n}\right) \rrbracket_{\mathrm{CoN}}^{\Gamma}$. If $\llbracket h\left(w_{1}, \ldots, w_{n}\right) \rrbracket_{\mathrm{CON}}^{\Gamma}$ is defined then by Lemma 2 we know that there is an $i_{j}$ such that $i_{j} \in w_{j}, 1 \leq j \leq n$. If $i_{j}, i_{j}^{\prime} \in w_{j}$, again by Lemma $2, @_{i_{j}} i_{j}^{\prime} \in w_{j}$. By Term and the fact that $h\left(i_{1}, \ldots, i_{n}\right)$ is $\Gamma$-defined, $h\left(i_{1}, \ldots, i_{n}\right)=h\left(i_{1}^{\prime}, \ldots, i_{n}^{\prime}\right) \in \Gamma$. We conclude $\left[h\left(i_{1}, \ldots, i_{n}\right)\right]=\left[h\left(i_{1}^{\prime}, \ldots, i_{n}^{\prime}\right)\right]$. In the case $\llbracket h\left(w_{1}, \ldots, w_{n}\right) \rrbracket_{\text {CON }}^{\Gamma}$ is undefined, we know that $h\left(i_{1}, \ldots, i_{n}\right)$ is not $\Gamma$-defined. By a similar
reasoning we can conclude that $h\left(i_{1}^{\prime}, \ldots, i_{n}^{\prime}\right)$ is not $\Gamma$-defined for all $i_{j}^{\prime}$ such that $i_{j}, i_{j}^{\prime} \in w$. Let's consider $\llbracket \rrbracket_{\text {PRED }}^{\Gamma}$. If $\left[t_{1}\right]=\left[t_{1}^{\prime}\right], \ldots,\left[t_{n}\right]=\left[t_{n}^{\prime}\right]$, we know that $t_{i}=t_{i}^{\prime} \in w$, and if $p\left(t_{1}, \ldots, t_{n}\right) \in w$, by $T E Q, p\left(t_{1}^{\prime}, \ldots, t_{n}^{\prime}\right) \in w$. For the case of $\llbracket \rrbracket_{\mathrm{FUN}}^{\Gamma}$, if $f\left(t_{1}, \ldots, t_{n}\right)$ is $\Gamma$ defined we reason as before. If $\left[t_{1}\right]=\left[t_{1}^{\prime}\right], \ldots,\left[t_{n}\right]=\left[t_{n}^{\prime}\right]$ we know that $t_{i}=t_{i}^{\prime} \in w$, and by $T E Q, f\left(t_{1}, \ldots, t_{n}\right)=f\left(t_{1}^{\prime}, \ldots, t_{n}^{\prime}\right) \in w$. Therefore, $\left[f\left(t_{1}, \ldots, t_{n}\right)\right]$ and $\left[f\left(t_{1}^{\prime}, \ldots, t_{n}^{\prime}\right)\right]$ are the same equivalence class. On the other hand, if $f\left(t_{1}, \ldots, t_{n}\right)$ is not $\Gamma$-defined, by $T E Q$ we can conclude that $f\left(t_{1}^{\prime}, \ldots, t_{n}^{\prime}\right)$ cannot be $\Gamma$-defined either.

Now we only have to look at the canonical assignment function. As the canonical model is var-saturated, the function is total. Furthermore, by Lemma 2, there is exactly one world $w$ such that $x \in w$.

Lemma 3 (Truth Lemma). Let $\Gamma$ be a named, pasted and var-saturated -mcs. Let $\mathcal{M}^{\Gamma}$ be the canonical model yielded by $\Gamma$ and let $u^{\Gamma} \in W^{\Gamma}$. Then, for all formulas $\phi$, $\phi \in u^{\Gamma}$ iff $\mathcal{M}^{\Gamma}, g^{\Gamma}, u^{\Gamma} \models \phi$.

Proof. Induction on $\phi$. To prove the atomic cases we must first show that, for any $\Gamma$-defined term $t,[t]=\llbracket t \rrbracket^{\Gamma}$. Case $h \in \mathrm{CON}: \llbracket h\left(s_{1}, \ldots, s_{n}\right) \rrbracket^{\Gamma}={ }^{\operatorname{def}} \llbracket h \rrbracket^{\Gamma}\left(\llbracket s_{1} \rrbracket^{\Gamma}, \ldots, \llbracket s_{n} \rrbracket^{\Gamma}\right)$. In the case $s_{i} \in \mathrm{VAR}, \llbracket s_{i} \rrbracket^{\Gamma}=g^{\Gamma}\left(s_{i}\right)=w_{i}$. Otherwise, if $s_{i} \in$ NOM, $\llbracket s_{i} \rrbracket^{\Gamma}=\llbracket s_{i} \rrbracket_{\text {NOM }}^{\Gamma}=w_{i}$. In both cases, $s_{i} \in w_{i}$. Therefore, $\llbracket h \rrbracket^{\Gamma}\left(\llbracket s_{1} \rrbracket^{\Gamma}, \ldots, \llbracket s_{n} \rrbracket^{\Gamma}\right)=\llbracket h \rrbracket^{\Gamma}\left(w_{1}, \ldots, w_{n}\right)={ }^{\text {def }}\left[h\left(s_{1}, \ldots, s_{n}\right)\right]$. Note that $\left[h\left(s_{1}, \ldots, s_{n}\right)\right]$ is $\Gamma$-defined by hypothesis. Case $f \in$ FUN: $\llbracket f\left(t_{1}, \ldots, t_{n}\right) \rrbracket^{\Gamma}=$ def $\llbracket f \rrbracket^{\Gamma}\left(\llbracket t_{1} \rrbracket^{\Gamma}, \ldots, \llbracket t_{n} \rrbracket^{\Gamma}\right)$. Because of Def, we know that $\left[t_{1}\right], \ldots,\left[t_{n}\right]$ are $\Gamma$-defined, therefore we can apply the induction hypothesis and conclude that $\llbracket f \rrbracket^{\Gamma}\left(\llbracket t 1 \rrbracket^{\Gamma}, \ldots, \llbracket t_{n} \rrbracket^{\Gamma}\right)=^{I H}$ $\llbracket f \rrbracket^{\Gamma}\left(\left[t_{1}\right], \ldots,\left[t_{n}\right]\right)==^{d e f}\left[f\left(t 1, \ldots, t_{n}\right)\right]$

From this, the truth lemma for atomic formulas is straightforward, because Def guarantees that terms in an atomic formula $\phi \in u^{\Gamma}$ are $\Gamma$-defined. The rest of the cases are handled in the same way as in Chapter 7 of (Blackburn, de Rijke, and Venema 2001).

We are now ready to enunciate a very general completeness result. We say that a $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$ formula $\phi$ is pure when its only atoms are nominals or variables. Furthermore, if $\phi$ is a pure formula, we say that $\psi$ is a pure instance of $\phi$ if $\psi$ is obtained from $\phi$ by uniform substitution of nominals and variables.

Theorem 1 (Completeness). Every consistent set of formulas of $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$ is satisfiable in a countable named model. Moreover, if $\Pi$ is a set of pure formulas and $\mathbf{P}$ is the normal hybrid logic obtained by adding all the formulas in $\Pi$ as extra axioms schemes to the axioms in Figure 3, then every P-consistent set of sentences is satisfiable in a countable named model based on a frame which validates every formula in $\Pi$.

Proof. For the first claim, given a consistent set of formulas $\Sigma$, use the extended Lindenbaum lemma to obtain a named, pasted and var-saturated set $\Sigma^{+}$. Let $\mathcal{M}=\left\langle W, U,\left\{R_{i}\right\}, \llbracket \rrbracket\right\rangle$ be the named model yielded by $\Sigma^{+}$. By Lemma 2, because $\Sigma^{+}$is named, $\Sigma^{+} \in W$. By the truth lemma, $\mathcal{M}, g^{\Sigma^{+}}, \Sigma^{+} \models \Sigma$. The model is countable because each state is named by some nominal. For the second claim, given a $\mathbf{P}$-consistent set of formulas $\psi$, use the extended Lindenbaum lemma to expand it to a named, pasted and var-saturated $\mathbf{P}$-mCS $\psi^{+}$. The named model $\mathcal{M}^{\psi}$ that $\psi^{+}$gives rise to will satisfy $\psi$ at $\psi^{+}$. In addition, as every
formula in $\Pi$ belongs to every $\mathbf{P}-\mathrm{MCS}$, we have that $\mathcal{M}^{\psi} \vDash \Pi$. As validity of pure formulas is preserved when moving from a model to its underlying frame, $\mathcal{M}^{\psi}$ validates $\Pi$.

## 4 Description Logics with Concrete Domains

Description logics (DLs) are a family of logical formalisms designed as a tool for knowledge representation and ontology engineering (Baader, Calvanese, McGuinness, Nardi, and Patel-Schneider 2003). To describe concrete qualities of objects, such as time, weight and temperature in a natural way, DLs have been extended with concretes domains. This gives rise to a new family of logics, such as $\mathcal{A L C}(\mathcal{D})$ (Baader and Hanschke 1991; Lutz 2002). To introduce the basic description logic with concrete domains $\mathcal{A L C}(\mathcal{D})$, we first define the notion of concrete domains and then define syntax and semantics.

Definition 7 (Concrete Domain). A concrete domain $\mathcal{D}$ is a pair $\left(\Delta_{\mathcal{D}}, \Phi_{\mathcal{D}}\right)$ where $\Delta_{\mathcal{D}}$ is a set and $\Phi_{\mathcal{D}}$ a set of predicate names. Each predicate name $P \in \Phi_{\mathcal{D}}$ is associated with an arity $n$ and an $n$-ary predicate $P^{\mathcal{D}} \subseteq \Delta_{\mathcal{D}}^{n}$.

Definition $8\left(\mathcal{A L C}(\mathcal{D})\right.$ Syntax and Semantics). Let $N_{c}, N_{R}$ and $N_{c F}$ be pairwise disjoint, countable infinite sets of concept names, role names and concrete features. Let $N_{a F}$ be a countably infinite subset of $N_{R}$. The elements of $N_{a F}$ are called abstract features. A path $u$ is a composition $f_{1} \ldots f_{n} g$ of $n$ abstract features $f_{1}, \ldots, f_{n}(n \geq 0)$ and a concrete feature $g$. For $\mathcal{D}$ a concrete domain, the set of $\mathcal{A L C}(\mathcal{D})$-concepts is the smallest set such that every concept name is a concept, and if $C$ and $D$ are concepts, $R$ is a role name, $g$ is a concrete feature, $u_{1}, \ldots, u_{n}$ are paths, and $P \in \Phi_{\mathcal{D}}$ is a predicate of arity $n$, then the following expressions are also concepts: $\neg C, C \sqcap D, C \sqcup D, \exists R . C, \forall R . C, \exists u_{1}, \ldots, u_{n} . P$ and $g \uparrow$.

An interpretation $\mathcal{I}$ is a pair $\left(\Delta_{\mathcal{I}},{ }^{\mathcal{I}}\right)$ where $\Delta_{\mathcal{I}}$ is a nonempty set called the domain and $\cdot{ }^{\mathcal{I}}$ is the interpretation function. The interpretation function maps each concept name $C$ to a subset $C^{\mathcal{I}}$ of $\Delta_{\mathcal{I}}$, each role name $R$ to a subset $R^{\mathcal{I}}$ of $\Delta_{\mathcal{I}} \times \Delta_{\mathcal{I}}$, each abstract feature $f$ to a partial function $f^{\mathcal{I}}: \Delta_{\mathcal{I}} \rightarrow \Delta_{\mathcal{I}}$, and each concrete feature $g$ to a partial function $g^{\mathcal{I}}: \Delta_{\mathcal{I}} \rightarrow \Delta_{\mathcal{D}}$. If $u=f_{1} \ldots f_{n} g$ is a path, then $u^{\mathcal{I}}(d)$ is defined as $g^{\mathcal{I}}\left(f_{n}^{\mathcal{I}} \ldots\left(f_{1}^{\mathcal{I}}(d)\right) \ldots\right)$.

The interpretation function is extended to arbitrary concepts as follows

$$
\begin{aligned}
(\neg C)^{\mathcal{I}} & :=\Delta_{\mathcal{I}} \backslash C^{\mathcal{I}} \quad(\exists R . C)^{\mathcal{I}}
\end{aligned}:=\left\{d \in \Delta_{\mathcal{I}} \mid\left\{e \in \Delta_{\mathcal{I}} \mid(d, e) \in R^{\mathcal{I}}\right\} \cap C^{\mathcal{I}} \neq \emptyset\right\}
$$

$\mathcal{A} \mathcal{L C}(\mathcal{D})$ translation. We provide a translation that embeds $\mathcal{A L C}(\mathcal{D})$ into $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$.
Definition 9. Let $N_{c}, N_{R}, N_{c F}$ and $N_{a F}$ be as in Definition 8 , and $\mathcal{D}=\left(\Delta_{\mathcal{D}}, \Phi_{\mathcal{D}}\right)$ be a given concrete domain. The translation $\operatorname{Tr}$ taking $\mathcal{A L C}(\mathcal{D})$ formulas to $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$ formulas over the signature $\left\langle N_{R}, N_{c} \cup \Phi_{\mathcal{D}}, \mathrm{FUN}, N_{c F}\right.$, NOM, VAR $\rangle$ is defined as follows.

We first define the translation over paths. Consider a sequence of paths $u_{1}, \ldots, u_{n}$ such that $u_{i}=f_{1}^{u_{i}} \ldots f_{n}^{u_{i}} g^{u_{i}}$, then

$$
\begin{aligned}
& \operatorname{Tr}_{x}\left(u_{1} \ldots u_{n} \cdot P\right)=@_{x}\left\langle f_{1}^{u_{1}}\right\rangle \ldots\left\langle f_{n}^{u_{n}}\right\rangle \downarrow y_{1} \cdot T r_{x, g^{u_{1}}}\left(u_{2} \ldots u_{n} \cdot P\right) \\
& \operatorname{Tr}_{x, g^{u_{1}, \ldots, g^{u_{n}}}(P)}=@_{x} P\left(g^{u_{1}}\left(y_{1}\right), \ldots, g^{u_{n}}\left(y_{n}\right)\right) .
\end{aligned}
$$

Now we can give the translation for formulas

$$
\begin{aligned}
\operatorname{Tr}(p) & =p\left(p \in N_{c}\right) & \operatorname{Tr}(\exists R . C) & =\langle R\rangle \operatorname{Tr}(C) \\
\operatorname{Tr}(C \sqcap D) & =\operatorname{Tr}(C) \wedge \operatorname{Tr}(D) & \operatorname{Tr}\left(\exists u_{1}, \ldots, u_{n} \cdot P\right) & =\downarrow x \cdot \operatorname{Tr}\left(u_{1} \ldots u_{n} . P\right) \\
\operatorname{Tr}(\neg C) & =\neg \operatorname{Tr}(C) & \operatorname{Tr}(g \uparrow) & =\downarrow x . \neg(g(x)=g(x)) .
\end{aligned}
$$

To each $\mathcal{A L C}(\mathcal{D})$ interpretation $\mathcal{I}$, we can associate a hybrid logic model $\mathcal{M}_{\mathcal{I}}$ with state domain $\Delta_{\mathcal{I}}$ and value domain $\Delta_{\mathcal{D}}$. To interpret the accessibility relations $\left\{R_{i}\right\}$ we can use the interpretation for roles names. The valuation $\llbracket \rrbracket_{\text {con }}$ can be defined using the interpretation for concrete features, and to define 【 $\rrbracket_{\text {pred }}$ we can use $\Phi_{\mathcal{D}}$ and the interpretation for concept names. The definition of $\llbracket \rrbracket_{\text {Nом }}$ and $\llbracket \rrbracket_{\text {fuN }}$ can be arbitrary.

Note that the formulas in the image of the translation do not have free variables and hence which assignment function we use is irrelevant. The following result can be established by formula induction.

Proposition 2 (Satisfiability preservation). Let $\phi$ be an $\mathcal{A L C}(\mathcal{D})$ formula. Given an interpretation $\mathcal{I}$ and an element $w \in \Delta_{\mathcal{I}}, w \in \phi^{\mathcal{I}}$ iff $\mathcal{M}_{\mathcal{I}}, w \models \operatorname{Tr}(\phi)$.

Axiomatic Extension. It is possible to extend the axiomatization of $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$ to characterize $\mathcal{A L C}(\mathcal{D})$ proof theoretically. To force $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$ models to behave as $\mathcal{A L C}(\mathcal{D})$ models, the abstract features modalities should behave in a functional way and the concrete predicates should be state-independent. Take the following extension of $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$ axiomatization, where $s$ ranges over NOM, $p$ over the subset of PRED used as concrete predicates, and $\langle F\rangle$ over the modalities used as abstract features.

| $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)_{\mathcal{A L C}(\mathcal{D})}$ Axiomatization |  |
| :--- | :--- |
| Axioms: | Rules: |
| All axioms for $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$ | All rules for $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$ |
| Rig $\vdash p\left(t_{1}, \ldots, t_{n}\right) \rightarrow @_{s} p\left(t_{1}, \ldots, t_{n}\right)$ |  |
| Fun $\vdash\langle F\rangle s \rightarrow[F]_{s}$ |  |

To show completeness, we build the canonical model $\mathcal{M}^{\Gamma}$ in the same way as in Definition 6 and we show that $\mathcal{M}^{\Gamma}$ has the desired properties. Because Fun is pure, by Theorem 1 we know that the canonical model is based on a frame which validates Fun, and it is clear that Fun characterizes functional modalities. We only have to look at Rig. Suppose that there is a world $w^{\Gamma} \in W^{\Gamma}$ and a concrete predicate $p$ such that $\mathcal{M}^{\Gamma}, g^{\Gamma}, w^{\Gamma} \models p$. We want to show that $\mathcal{M}^{\Gamma}, g^{\Gamma}, v^{\Gamma} \models p$ for an arbitrary world $v^{\Gamma} \in W^{\Gamma}$. Because $\mathcal{M}^{\Gamma}$ is named, let $i$ and $j$ be the nominals that name $w^{\Gamma}$ and $v^{\Gamma}$ respectively. Rig guarantees that $p \rightarrow @_{j} p \in w^{\Gamma}$, and by Lemma $3, \mathcal{M}^{\Gamma}, g^{\Gamma}, w^{\Gamma} \models p \rightarrow @_{j} p$. That implies that $\mathcal{M}^{\Gamma}, g^{\Gamma}, v^{\Gamma} \models p$, and therefore $p$ is state-independent.

As a final remark, with these requirements we are not characterizing the concrete domain of $\operatorname{ALC}(\mathcal{D})$, only its "abstract" domain. Once the concrete domain is fixed, an axiomatic extension should be given to complete the characterization.

## 5 Half-order Logic

Half-order logics $(\mathcal{H O})$ were introduced in (Henzinger 1990; Alur and Henzinger 1990) in the verification community. Standard modal logic is extended with a concrete domain in which each state is associated with a value. These values can only be accessed by the so-called "freeze" quantifier, that binds a variable to the value of the current state, in a way similar to $\downarrow$. We start by defining the syntax and semantics of half-order logic $\mathcal{H O}$.

Definition $10(\mathcal{H O}$ Syntax and Semantics). Let $V$ be a countably infinite set of variables, and $F$ and $P$ be countably sets of $n$-ary function symbols and $n$-ary predicate symbols respectively. Given $x \in V$ and $f \in F$, the $\mathcal{H O}$ terms are defined as follows: $x$ is a term, and if $t_{1}, \ldots, t_{n}$ are terms, $f\left(t_{1}, \ldots, t_{n}\right)$ is also a term. For $p \in P$ and terms $t_{1}, \ldots, t_{n}$, the atomic formulas are $t_{1}=t_{2}$ and $p\left(t_{1}, \ldots, t_{n}\right)$, and if $\phi$ and $\psi$ are formulas, then the following expressions are also formulas: $\neg \phi, \phi \wedge \psi, \square \phi, x . \phi$.

The language of $\mathcal{H O}$ is interpreted over structures of the form $\mathcal{S}=\left\langle W, U, \rightarrow_{\square}, \|, \llbracket f \rrbracket_{f \in F}\right.$, $\left.\llbracket p \rrbracket_{p \in P}\right\rangle$, where $W$ is a set of states, $U$ is a set of values, and $\rightarrow \square \subseteq W^{2}$ is an accessibility relation on the states. $\|: W \rightarrow U$ is a value function that associates a value $|w|$ with every state, $\llbracket f \rrbracket: U^{n} \rightarrow U$ is an assignment function for all functional symbols $f \in F$, and $\llbracket p \rrbracket: W \rightarrow \wp\left(U^{n}\right)$ is an assignment function for all predicate symbols $p \in P$.

Given a model $\mathcal{S}$, an assignment $g: V \rightarrow U$ for $\mathcal{S}$ and a state $w \in W$. Let $\llbracket x \rrbracket=g(x)$ and $\llbracket f\left(t_{1}, \ldots, t_{n}\right) \rrbracket=\llbracket f \rrbracket\left(\llbracket t_{1} \rrbracket, \ldots, \llbracket t_{n} \rrbracket\right)$, then

$$
\begin{aligned}
& \mathcal{M}, g, w \models t_{1}=t_{2} \quad \text { iff } \quad \llbracket t_{1} \rrbracket=\llbracket t_{2} \rrbracket \\
& \mathcal{M}, g, w=p\left(t_{1}, \ldots, t_{n}\right) \quad \text { iff } \quad\left(\llbracket t_{1} \rrbracket, \ldots, \llbracket t_{n} \rrbracket\right) \in \llbracket p \rrbracket^{w} \\
& \mathcal{M}, g, w \models \neg \phi \quad \text { iff } \mathcal{M}, g, w \not \models \phi \\
& \mathcal{M}, g, w \models \phi \wedge \psi \quad \text { iff } \quad \mathcal{M}, g, w \models \phi \text { and } \mathcal{M}, g, w \models \psi \\
& \mathcal{M}, g, w=\square \phi \quad \text { iff } \mathcal{M}, g, t \models \phi \text {, for all } t \in W \text { with } w \rightarrow \square t \\
& \mathcal{M}, g, w \models x . \phi \quad \text { iff } \quad \mathcal{M}, g_{|w|}^{x}, w \models \phi \text {. }
\end{aligned}
$$

Half-order translation. We can now define a translation from $\mathcal{H O}$ to $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$.
Definition 11. Let $V, F$ and $P$ be as in Definition 10. The translation $\operatorname{Tr}$ taking $\mathcal{H O}$ formulas to $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$ formulas over the signature $\left\langle\left\{R_{\square}\right\}, P, F,\left\{h_{\|}\right\}\right.$, nOM, $\left.V\right\rangle$ is defined as follows

$$
\begin{aligned}
\operatorname{Tr}(x) & =h_{\|}(x) & \operatorname{Tr}(\neg \phi) & =\neg \operatorname{Tr}(\phi) \\
\operatorname{Tr}\left(t_{1}=t_{2}\right) & =\operatorname{Tr}\left(t_{1}\right)=\operatorname{Tr}\left(t_{2}\right) & \operatorname{Tr}\left(p\left(t_{1}, \ldots, t_{n}\right)\right) & =p\left(\operatorname{Tr}\left(t_{1}\right), \ldots, \operatorname{Tr}\left(t_{n}\right)\right) \\
\operatorname{Tr}(\phi \wedge \psi) & =\operatorname{Tr}(\phi) \wedge \operatorname{Tr}(\psi) & \operatorname{Tr}\left(f\left(t_{1}, \ldots, t_{n}\right)\right) & =f\left(\operatorname{Tr}\left(t_{1}\right), \ldots, \operatorname{Tr}\left(t_{n}\right)\right) \\
\operatorname{Tr}(x . \phi) & =\downarrow x . \operatorname{Tr}(\phi) & \operatorname{Tr}(\square \phi) & =\left[R_{\square}\right] \operatorname{Tr}(\phi) .
\end{aligned}
$$

We can establish a correspondence between a half-order model $\mathcal{S}$ and a hybrid model $\mathcal{M}_{\mathcal{S}}$. To interpret the accessibility relation $R_{\square}$ we can use $\rightarrow_{\square}$, and to interpret the concrete function $h_{\|}$we can use the function $\|$. The valuation $\llbracket f \rrbracket_{\mathrm{FUN}}$ can be defined using $\llbracket f \rrbracket_{f \in F}$. To define $\llbracket p \rrbracket_{\text {pred }}$ we can use $\llbracket p \rrbracket_{p \in P}$. The definition of $\llbracket \rrbracket_{\text {Noм }}$ can be arbitrary.

Note that a proper assignment function needs only to agree with the value of variables through $h_{\|}$. Given a $\mathcal{H O}$ model $\mathcal{S}$ and an assignment function $g$ for $\mathcal{S}$, we define $g_{\mathcal{S}}$ : var $\rightarrow W$ as an assignment function such that $\llbracket h_{\|} \rrbracket_{\cos }\left(g_{\mathcal{S}}(x)\right)=g(x)$ for all $x \in V$. The following result can be established by formula induction.

Proposition 3 (Satisfiability preservation). Let $\phi$ be a $\mathcal{H O}$ formula. Given a model $\mathcal{S}$, an assignment $g$ for $\mathcal{S}$ and a state $w \in W, \mathcal{S}, g, w \models \phi$ iff $\mathcal{M}_{\mathcal{S}}, g_{\mathcal{S}}, w \models \operatorname{Tr}(\phi)$.

Axiomatic Extension. It is possible to extend the axiomatization of $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$ to be able to characterize $\mathcal{H O}$ proof theoretically. In this case, we should force functions to be total. The axiomatization is the following, where $t$ ranges over arbitrary terms.

| $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)_{\text {half-order }}$ |  |
| :--- | :--- |
| Axioms: | Rules: |
| All axioms for $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$ | All rules for $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$ |
| $T o t \quad \vdash t=t$ |  |

To show completeness for this case, we again build the canonical model $\mathcal{M}^{\Gamma}$ in the same way as in Definition 6. Because of Tot, all terms are $\Gamma$-defined, and therefore $\llbracket \rrbracket_{\text {Con }}^{\Gamma}$ and $\llbracket \rrbracket_{\mathrm{FUN}}^{\Gamma}$ become total functions. As for the case of $\mathcal{A L C}(\mathcal{D})$, we are not characterizing concrete domains of $\mathcal{H O}$, only its "abstract" domain.

## 6 Conclusion

In this paper we presented an extension of $\mathcal{H} \mathcal{L}(@, \downarrow)$ that incorporates a concrete domain to the standard hybrid models. The language and the satisfiability notion were extended accordingly to make them adequate for concrete values. The main result of this paper was a complete axiomatic system for $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$. We also showed faithful embeddings from $\mathcal{A L C}(\mathcal{D})$ and $\mathcal{H O}$ into $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$ via translations, and established that both translations preserve satisfiability. Furthermore, we presented an extension of $\left.\mathcal{H} \mathcal{L}_{\mathcal{C}} @, \downarrow\right)$ axiomatization that characterizes the abstract models of $\mathcal{A L C}(\mathcal{D})$ and $\mathcal{H O}$ proof theoretically. All the necessary requirements can be expressed in the language of $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$, and this shows that $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$ can be taken as a proper unifying framework for both logics.

With respect to complexity, in (Blackburn and Seligman 1995) (and originally in unpublished work by V. Goranko), it has been proved that already the $\downarrow$ fragment of $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$ is undecidable, and this result directly implies undecidability of $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$. Our future work will be focused on decidability results for fragments of $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$. We expect to be able to "import" decidability results from $\mathcal{A L C}(\mathcal{D})$ and $\mathcal{H O}$ to fragments of $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$. As an example, $\mathcal{H O}$ is decidable for linear frames with the natural numbers as concrete domain,
in which there is a unique concrete monotonic function and a limited set of functions. This result might transfer to a natural fragment of $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$.

## References

Alur, R. and T. Henzinger (1990). Real-Time Logics: Complexity and Expressiveness. In 5th Annual IEEE Symp. on Logic in Computer Science, Washington, D.C., pp. 390-401. IEEE Computer Society Press.

Areces, C., P. Blackburn, and M. Marx (2001). Hybrid logics: characterization, interpolation and complexity. The J. of Symbolic Logic 66(3), 977-1010.
Baader, F., D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider (Eds.) (2003). The description logic handbook: theory, implementation, and applications. Cambridge University Press.
Baader, F. and P. Hanschke (1991). A scheme for integrating concrete domains into concept languages. In Proc. of IJCAI-91, Sydney, pp. 452-457.
Blackburn, P., M. de Rijke, and Y. Venema (2001). Modal logic. Cambridge University Press.

Blackburn, P. and J. Seligman (1995). Hybrid languages. J. of Logic, Language and Information 4, 251-272.

Blackburn, P. and B. ten Cate (2004). Pure extensions, proof rules and hybrid axiomatics. In Preliminary Proc. of AiML 2004, pp. 16-29. University of Manchester.
Henzinger, T. (1990). Half-order modal logic: how to prove real-time properties. In PODC '90: Proc. of the 9th annual ACM Symp. on Principles of Distributed Computing, pp. 281-296. ACM Press.
Lutz, C. (2002). The Complexity of Reasoning with Concrete Domains. Ph. D. thesis, Teaching and Research Area for Theoretical Computer Science, RWTH Aachen.
ten Cate, B. (2005). Model Theory for Extended Modal Languages. Ph. D. thesis, University of Amsterdam. ILLC Dissertation Series DS-2005-01.

# A Fully Internalized Sequent Calculus for Hybrid Categorial Logics 

Magdalena Ortiz<br>Institute of Information Systems, Vienna University of Technology.<br>Favoritenstraße 9-11, Vienna, Austria<br>magdalen@logic.at


#### Abstract

In this paper, a sequent calculus for a hybrid categorial type logic (HCTL) is obtained following Seligman's internalization strategy (Seligman 2001). With this strategy, a sequent calculus for a hybrid language can be developed starting from a first order sequent calculus. Seligman exemplifies his strategy developing a calculus for hybrid modal logics, but rises the question of whether the strategy works in general. We investigate this issue in the particular case of categorial type logics. As categorial type logics lack Boolean structure, a successful hybridization would indicate that the strategy is indeed rather general and does not depend on the availability of Boolean connectives. In this paper, we will see that this is the case, and moreover, since we can easily arrive to an intuitionistic version of the calculus, we will see that a classical base is not needed either.


## 1 Introduction

The basic categorial type language (CTL) is called NL and it was introduced by Lambek in (Lambek 1961) to model linguistic composition. It is a non-associative, non-commutative modal calculus of pure residuation containing just three operators: $\bullet, \rightarrow$ and $\leftarrow$. NL was extended by Moortgat to $\mathrm{NL}(\diamond)$ by the addition of a pair of residuated unary modalities $\vec{\diamond}$ and $\overleftarrow{\square}$ (Moortgat 1996).
$\mathrm{NL}(\diamond)$ is interesting as an example of a purely modal language: all its operators are modalities defined in terms of accessibility relations and it contains no Boolean structure. Despite its simplicity, it is considered to be suitable for reasoning about different linguistic phenomena and useful to describe complex relational structures (Moortgat 1996). But like all standard modal languages, $\mathrm{NL}(\diamond)$ lacks the means to directly refer to particular elements in a model. Hybrid modal languages overcome this shortcoming by providing a way to assign names to points and mechanisms to access them. In many cases, hybridization of a modal language does not only provide additional expressive power, but it also has a positive impact on its proof and model theory. Hybridization for the basic modal logic (e.g., the $\mathcal{H}(@, \downarrow)$ language) has been by now extensively investigated (Areces, Blackburn, and Marx 2001; HyLo 2006).

In (2001) Seligman shows how, using a strategy called internalization, a sequent calculus for hybrid modal logics can be defined starting from a calculus for first order logic (FO). Seligman describes the internalization strategy by applying it to the basic modal logic, but mentions that it is unclear whether it will also work for other non-standard modal languages. In this sense, CTL is an specially interesting test case, as the absence of Boolean operators makes it hard to tell at first sight how far internalization would go.

In this paper, we apply Seligman's strategy to $\mathrm{NL}(\diamond)$. The purpose of this paper is two-fold. First, we want to test Seligman's method in a fairly non-classical modal logic. Second, we aim to obtain a fully internalized sequent calculus for hybrid categorial type logics.

## 2 Hybrid Categorial Logic

We start by introducing the syntax and semantics of the hybrid categorial logic HCTL. Defining HCTL is simple: take the syntactic and semantic clauses of both $\mathrm{NL}(\diamond)$ and $\mathcal{H}(@, \downarrow)$ and pool them together.

Formally, let $\mathcal{S}$ be a signature with variables $X=\left\{x, x_{1}, x_{2}, \ldots\right\}$, propositional symbols $P=\left\{p, p_{1}, p_{2}, \ldots\right\}$, a binary relational symbol $r_{\diamond}$ and a ternary one $r_{\bullet}$. The atomic formulas of HCTL are the individual variables in $X$ and the symbols in $P$. Complex formulas are defined as follows: if $x$ is a variable in $X$, and $\varphi, \psi$ are formulas, then $\vec{\diamond} \varphi$, $\Xi_{\varphi} \mathrm{E} \varphi, x: \varphi, \downarrow_{x} \varphi, \varphi \bullet \psi, \varphi \leftarrow \psi$ and $\varphi \rightarrow \psi$ are formulas. Note that the Boolean operators $\neg, \wedge, \vee, \supset, \equiv$ do not occur in formulas of HCTL. Moreover, all connectives must be defined as primitive, and none of them can be expressed in terms of the others.

Like in any modal language, the semantics of HCTL is defined in terms of relational (Kripke) structures over the signature $\mathcal{S}$.

Definition 1. A model $\mathcal{M}=\langle\mathbf{M}, \cdot \mathcal{M}\rangle$ for HCTL is a relational structure with a non-empty domain $\mathbf{M}$, and an interpretation function ${ }^{\mathcal{M}}$ such that $\left(r_{\diamond}\right)^{\mathcal{M}} \subseteq \mathbf{M}^{2},\left(r_{\bullet}\right)^{\mathcal{M}} \subseteq \mathbf{M}^{3}$ and $p^{\mathcal{M}} \subseteq \mathbf{M}$ for each $p \in P$. Given a model $\mathcal{M}$, elements $a, b, c$ in $\mathbf{M}$ and an assignment $g: X \rightarrow \mathbf{M}$ for the variables, the satisfiability relation is inductively defined as follows:

```
\(\mathcal{M}, a, g \models x \quad\) iff \(\quad g(x)=a\) where \(x \in X\)
\(\mathcal{M}, a, g \models p \quad\) iff \(\quad a \in p^{\mathcal{M}}\) where \(p \in P\)
\(\mathcal{M}, a, g \models x: \varphi \quad\) iff \(\quad \mathcal{M}, g(x), g \models \varphi\)
\(\mathcal{M}, a, g \models \downarrow_{x} \varphi \quad\) iff \(\quad \mathcal{M}, a, g[x / a] \models \varphi\)
\(\mathcal{M}, a, g \models \mathrm{E} \varphi \quad\) iff \(\quad \exists b \in \mathbf{M}, \mathcal{M}, b, g \models \varphi\)
\(\mathcal{M}, a, g \models \vec{\diamond}_{\varphi} \quad\) iff \(\quad \exists b \in \mathbf{M},\langle a, b\rangle \in\left(r_{\diamond}\right)^{\mathcal{M}}\) and \(\mathcal{M}, b, g \models \varphi\)
\(\mathcal{M}, b, g \models \overleftarrow{\square}_{\varphi} \quad\) iff \(\quad \forall a \in \mathbf{M}\), if \(\langle a, b\rangle \in\left(r_{\diamond}\right)^{\mathcal{M}}\) then \(\mathcal{M}, a, g \models \varphi\)
\(\mathcal{M}, a, g \models \varphi \bullet \psi \quad\) iff \(\quad \exists b, c \in \mathbf{M},\langle a, b, c\rangle \in\left(r_{\bullet}\right)^{\mathcal{M}}, \mathcal{M}, b, g \models \varphi\) and \(\mathcal{M}, c, g \models \psi\)
\(\mathcal{M}, b, g \models \varphi \leftarrow \psi \quad\) iff \(\quad \forall a, c \in \mathbf{M}\), if \(\langle a, b, c\rangle \in\left(r_{\bullet}\right)^{\mathcal{M}}\) and \(\mathcal{M}, c, g \models \psi\) then \(\mathcal{M}, a, g \models \varphi\)
\(\mathcal{M}, c, g \models \varphi \rightarrow \psi \quad\) iff \(\quad \forall a, b \in \mathbf{M}\), if \(\langle a, b, c\rangle \in\left(r_{\bullet}\right)^{\mathcal{M}}\) and \(\mathcal{M}, b, g \models \varphi\) then \(\mathcal{M}, a, g \models \psi\)
```

Note that the connectives $\overleftarrow{\square}$ and $\vec{\diamond}$ form a residuated pair (they are not duals as the $\diamond$ and $\square$ in the basic modal logic). Analogously, $\bullet, \leftarrow$ and $\rightarrow$ form a residuated triple.
$\Gamma \vdash \Delta$ is a sequent of HCTL if $\Gamma$ and $\Delta$ are lists of HCTL formulas. $\mathcal{M}$ is a model of such a sequent if every $\mathcal{M}, a, g$ satisfying all formulas in $\Gamma$ satisfies some formula in $\Delta$. If every model is a model of $\Gamma \vdash \Delta$, then $\Gamma \vdash \Delta$ is a valid sequent.

## 3 Proving HCTL Theorems in FO

As can be seen from the definitions in the previous section, every HCTL model $\mathcal{M}$ is a first order model. Moreover, the semantics of HCTL can be defined in terms of FO, and hence we can use first order machinery to prove valid sequents of HCTL. Following Seligman, we will now cast HCTL formulas as 'contextualized' first order formulas and express their semantics in FO.

We introduce the first order language obtained by extending the signature $\mathcal{S}$ with a new $n+1$-ary predicate symbol $p_{\varphi}$ for each HCTL formula $\varphi$ with $n$-free variables. Intuitively, the formula $p_{\varphi} x$ means that the point at which $x$ is interpreted has the property of satisfying the formula $\varphi$, i.e., that $\varphi$ is true at point $g(x)$, for $g$ some assignment, according to the semantics of HCTL. Since the semantic conditions in Definition 1 can be easily expressed in FO, we can use this extended language to give an FO characterization of HCTL. For example, consider the formula $\vec{\nabla} \varphi$, for $\varphi$ without free variables. As defined above, this formula will be true at some point $a$ of the model $\mathcal{M}$ iff $\mathcal{M}$ satisfies the formula $\exists y\left(r_{\diamond} x y \wedge\right.$ $p_{\varphi} y$ ), where $x$ is a variable interpreted as $a$.

In order to make things more readable, we will write $x: \varphi$ instead of the formula $p_{\varphi} x$. The free variables of $x: \varphi$ are $x$ and the free variables of $\varphi$. Note that in the previous section we have already introduced : as a connective of HCTL. Now we are introducing : with a different meaning, not as a logical connective (neither of FO nor of HCTL), but as a 'metasymbol', i.e., as a way to abbreviate certain fist order formulas. The reason for this double meaning will be made clear in Section 4, when we take the internalization step.

The semantics of HCTL can be expressed in FO through the following set $\Theta$ of formulas:

$$
\begin{array}{llll}
\left(\theta_{1}\right) & \forall z \forall x(z: x \equiv x=z) & \left(\theta_{6}\right) & \forall z\left(z: \vec{\diamond} \varphi \equiv \exists x\left(r_{\diamond} z x \wedge x: \varphi\right)\right) \\
\left(\theta_{2}\right) & \forall z(z: p \equiv p z) & \left(\theta_{7}\right) & \forall z\left(z: \boxed{\square} \varphi \equiv \forall x\left(r_{\diamond x z \supset x: \varphi))}^{\left(\theta_{3}\right)}\right.\right.
\end{array} \forall z \forall x(z:(x: \varphi) \equiv x: \varphi) \quad 1\left(\theta_{8}\right) \quad \forall z\left(z: \varphi \bullet \psi \equiv \exists x \exists y\left(r_{\bullet} z x y \wedge x: \varphi \wedge y: \psi\right)\right)
$$

Note that each $\theta_{i}$ is simply the formalization of a semantic condition in Definition 1. The second occurrence of the symbol : in $\theta_{3}$ corresponds to the connective in the syntax of HCTL, while : stands for the 'metasymbol' in FO in all remaining occurrences in $\Theta$.

The set of sentences $\Theta$ will be our background theory. We say that a sequent $\Gamma \vdash \Delta$ (where $\Gamma$ and $\Delta$ are now lists of FO formulas) is $\Theta$-valid if every (first order) model of $\Theta$ is a model of $\Gamma \vdash \Delta$. For any list $\Gamma$ of formulas of HCTL, $u: \Gamma$, denotes the list of (first order) formulas $u: \varphi$ for each $\varphi \in \Gamma$. Since $\Theta$ captures the semantics of HCTL, the following lemma holds:

$$
\left.\begin{array}{ll}
\hline \varphi, \Gamma \vdash \Delta, \varphi
\end{array} \mathrm{Ax}\right] \quad \frac{\Gamma \vdash \Delta}{\Gamma^{\prime} \vdash \Delta^{\prime}}[\mathrm{S}]^{1}
$$

${ }^{1}$ if the lists $\Gamma$ and $\Delta$ contain the same set of formulas as $\Gamma^{\prime}$ and $\Delta^{\prime}$ resp.

## Structural rules

$$
\left.\begin{array}{cc}
\frac{\Gamma \vdash \Delta, \varphi}{\neg \varphi, \Gamma \vdash \Delta}[\neg \mathrm{L}] & \frac{\varphi \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \neg \varphi}[\neg \mathrm{R}] \\
\frac{\varphi, \Gamma \vdash \Delta \quad \psi, \Gamma \vdash \Delta}{\varphi \vee \psi, \Gamma \vdash \Delta}[\mathrm{VL}] & \frac{\Gamma \vdash \Delta, \varphi, \psi}{\Gamma \vdash \Delta, \varphi \vee \psi}[\mathrm{VR}] \\
\frac{\varphi[u / x], \Gamma \vdash \Delta}{\exists x \varphi, \Gamma \vdash \Delta}[\exists \mathrm{~L}] u \text { new } & \frac{\Gamma \vdash \Delta, \varphi[u / x]}{\Gamma \vdash \Delta, \exists x \varphi}[\exists \mathrm{R}] \\
\frac{u: v, \Gamma[w / u] \vdash \Delta[w / u]}{u: v, \Gamma[w / v] \vdash \Delta[w / v]}\left[=\mathrm{L}_{1}\right] & \frac{u: v, \Gamma[w / v] \vdash \Delta[w / v]}{u: v, \Gamma[w / u] \vdash \Delta[w / u]}\left[=\mathrm{L}_{2}\right]
\end{array} \frac{\Gamma \vdash \Delta, u=u}{\Gamma \vdash \mathrm{R}]} \mathrm{l} \mathrm{C}\right]
$$

## Logical Rules

Figure 1.1: Sequent Calculus $\mathbf{S}$ for FO

Lemma 1. Let $\Gamma \vdash \Delta$ be a sequent of HCTL and $u$ an arbitrary variable not occurring in $\Gamma \vdash \Delta$. The sequent $\Gamma \vdash \Delta$ is valid iff $u: \Gamma \vdash u: \Delta$ is $\Theta$-valid.

Clearly, we can use the proof machinery of FO to prove valid HCTL sequents. For example, any standard sound and complete Gentzen sequent calculus for FO with equality will do. We pick (arbitrarily) the sequent calculus introduced in (Seligman 2001), given in Figure 1.1, and call it $\mathbf{S}$.
$\mathbf{S}$ has two structural rules, $[\mathrm{Ax}]$ and $[\mathrm{S}]$, plus the standard (context sharing) rules for the logical connectives and the Barwise rules for equality. The connectives $\wedge, \supset, \equiv$ and $\forall$ are expressed as usual, and their respective rules can be derived.

Finally, the standard cut rule

$$
\frac{\Gamma \vdash \Delta, \varphi \quad \varphi, \Gamma^{\prime} \vdash \Delta^{\prime}}{\Gamma, \Gamma^{\prime} \vdash \Delta, \Delta^{\prime}}[C u t]
$$

is admissible in $\mathbf{S}$, ensuring the subformula property. ${ }^{1}$ This makes $\mathbf{S}$ a modular calculus: if we take any fragment $F$ of FO closed under subformulas, the calculus obtained as a restriction of $\mathbf{S}$ to the operators mentioned in $F$ will be sound and complete for validity of $F$ sequents. As we will see in Section 4, choosing a modular calculus will be crucial for our aims.

[^33]Let's state clearly how we would use $\mathbf{S}$ to prove $\Theta$-validity. Since predicate logic is compact and $\mathbf{S}$ is sound and complete, it follows that a sequent $\Gamma \vdash \Delta$ of our first order language is $\Theta$-valid iff there are formulas $\theta_{1}, \ldots, \theta_{n}$ in $\Theta$ such that $\theta_{1}, \ldots, \theta_{n}, \Gamma \vdash \Delta$ is a theorem of $\mathbf{S}$. Thus we easily obtain a calculus for $\Theta$-validity by adding the sentences of $\Theta$ as axioms to $\mathbf{S}$, i.e., for each sentence $\theta_{i}$ in $\Theta$ we add to $\mathbf{S}$ the $\Theta$-axiom:

$$
\overline{\Gamma \vdash \Delta, \theta_{i}}\left[\mathrm{~A} \theta_{i}\right] .
$$

Let us call such calculus $\mathbf{S}+\Theta$-Axioms. Then the following holds:
Theorem 1. A sequent of FO is $\Theta$-valid iff it can be proved in $\mathbf{S}+\Theta$-Axioms $+[\mathrm{Cut}]$.
Proof. As the new rules are $\Theta$-valid and the rules of $\mathbf{S}$ preserve validity and $\Theta$-validity, all theorems of $\mathbf{S}+\Theta$-Axioms $+[\mathrm{Cut}]$ are $\Theta$-valid. For the converse, observe that since $\vdash \theta_{i}$ is a $\Theta$-axiom for each $\theta_{i}$, a proof of $\theta_{1}, \ldots, \theta_{n}, \Gamma \vdash \Delta$ in $\mathbf{S}$ can be transformed into a proof of $\Gamma \vdash \Delta$ in $\mathbf{S}+\Theta$-Axioms $+[\mathrm{Cut}]$ with $n$ applications of the cut rule.

Note the use of the cut rule in the proof of Theorem 1. Indeed, the cut rule cannot be directly eliminated from $\mathbf{S}+\Theta$-Axioms $+[\mathrm{Cut}]$ and the subformula property is lost. In particular, the problematic cuts are those involving $\Theta$-axioms. This is a standard problem when adding axioms to systems where cut was eliminated.

Lemma 2. A proof of $\mathbf{S}+\Theta$-Axioms $+[\mathrm{Cut}]$ can be transformed into a proof in which all cuts have the form

$$
\left.\frac{}{\frac{\pi}{\Gamma \vdash \Delta, \theta}\left[\mathrm{A} \theta_{i}\right]} \begin{array}{|c}
\stackrel{\pi}{\vdots} \Gamma^{\prime} \vdash \Delta^{\prime} \\
\Gamma, \Gamma^{\prime} \vdash \Delta, \Delta^{\prime}
\end{array} \mathrm{Cut}\right]
$$

where $\theta$ is a formula of $\Theta$ and the principal formula of the last rule of $\pi$.

## 4 Regaining Cut Elimination

By inspecting Lemma 2, we can realize that instead of the $\Theta$-axioms, which generate unwanted cuts, we can use rules that do the same job. Each $\Theta$-axiom is required to prove some particular kind of formula, and in each case we can define a rule that has the same effect. As an example, we show how we can prove $u: \overleftarrow{\square} \varphi$ on the left of a sequent using the $\Theta$-axiom $\theta_{7}$.

In this way, from a sequent of the form $\Gamma \vdash \Delta, r_{\diamond} v u$ and a sequent of the form $v: \varphi, \Gamma \vdash \Delta$, we can prove a sequent of the form $u: \overleftarrow{\square} \varphi, \Gamma \vdash \Delta$. However, we could obtain the same result by using the following rules:

$$
\frac{\Gamma \vdash \Delta, r_{\diamond} u v}{\Gamma \vdash \Delta, u: \overrightarrow{\diamond v}^{\prime}}\left[r_{\diamond} \mathrm{R}\right] \quad \frac{\Gamma \vdash \Delta, v: \vec{\diamond} u \quad v: \varphi, \Gamma \vdash \Delta}{u: \overleftarrow{\square} \varphi, \Gamma \vdash \Delta}[: \overleftarrow{\mathrm{L}}] .
$$

Rule $\left[r_{\diamond} R\right]$ does not have the subformula property, but we can improve it. It is easy to verify that when proving $\Theta$-validity of a sequent with our new rules, the application of rule $\left[r_{\diamond} \mathrm{R}\right]$ commutes with all other rules and thus it can be pushed up to the leaves of the proof-tree. At the leaves, its application can be replaced by axioms. So, instead of the rule $\left[r_{\diamond} \mathrm{R}\right]$ we can use the axiom $[\mathrm{Ax} \mathrm{R}]$ :

$$
\left.\frac{\overline{r_{\diamond} u v, \Gamma \vdash \Delta, r_{\diamond} u v}}{r_{\diamond} u v, \Gamma \vdash \Delta, u: \stackrel{\rightharpoonup}{\diamond}}\left[\mathrm{Ax}_{\diamond}\right] \mathrm{R}\right] \rightsquigarrow \overline{r_{\diamond} u v, \Gamma \vdash \Delta, u: \overrightarrow{\diamond v}}[\mathrm{Ax} \diamond \mathrm{R}] .
$$

We can apply this method systematically and obtain a set of rules that replace all $\Theta$ axioms, given in Figure 1.2. They are divided into two groups, which we call IA (Interface Axioms) and LLR (Labeled Logical Rules).

Since the rules in $\mathbf{L L R}+\mathbf{I A}$ do the same work as $\Theta$-axioms, we can replace $\mathbf{S}+\Theta$ Axioms $+[\mathrm{Cut}]$ by $\mathbf{S}+\mathbf{L L R}+\mathbf{I} \mathbf{A}+[\mathrm{Cut}]$.

Lemma 3. A sequent of FO is a theorem of $\mathbf{S}+\Theta$-Axioms $+[\mathrm{Cut}]$ iff it is theorem of $\mathbf{S}+\mathbf{L L R}+\mathbf{I A}+[\mathrm{Cut}]$.

Moreover, [Cut] can now be eliminated.
Lemma 4. [Cut] is admissible in $\mathbf{S}+\mathbf{L L R}+\mathbf{I A}$.
Proof. We have to prove two things:

1) Cuts can be pushed up though the new rules when the cut formula is not principal. For most of the rules this is trivial, since they do not alter any of the non-principal formulas. Only the $\left[: \mathrm{L}_{1}\right]$ and $\left[: \mathrm{L}_{2}\right]$ are different, but for them the proof is exactly as for the similar rules for handling equality in $\mathbf{S}$.
2) Cut rank can be decreased when the cut formula is the main formula of one of the new rules. The proof of this part must be done rule by rule, but it is also quite straightforward. As an example, we show how cut rank is decreased when the cut formula is the main formula of the [: $\overleftarrow{\square} \mathrm{R}]$ and [: $\overleftarrow{\square}]$ rules. The proof:

$$
\begin{array}{ccc}
\pi_{1} & \pi_{2}^{\prime} & \pi_{2}^{\prime \prime} \\
\vdots & \vdots & \vdots \\
v: \stackrel{\rightharpoonup}{\nabla} u, \Gamma, \vdash \Delta, v: \varphi \\
\frac{\Gamma, \vdash \Delta, u: \overleftarrow{\square} \varphi}{}[: \overleftarrow{\square} \mathrm{R}] & \frac{\Gamma^{\prime} \vdash \Delta^{\prime}, w: \stackrel{\rightharpoonup}{ } u}{} & w: \varphi, \Gamma^{\prime}, \vdash \Delta^{\prime} \\
\Gamma, \Gamma^{\prime} \vdash \Delta, \Delta^{\prime} & \overleftarrow{\square} \varphi, \Gamma^{\prime} \vdash \Delta^{\prime} \\
& \boxed{C u t}]
\end{array}
$$

$$
\begin{aligned}
& \frac{u: v, \Gamma[w / u] \vdash \Delta[w / u]}{u: v, \Gamma[w / v] \vdash \Delta[w / v]}\left[: \mathrm{L}_{1}\right] \quad \frac{u: v, \Gamma[w / v] \vdash \Delta[w / v]}{u: v, \Gamma[w / u] \vdash \Delta[w / u]}\left[: \mathrm{L}_{2}\right] \quad \frac{}{\Gamma \vdash \Delta, u: u}[: \mathrm{R}] \\
& \frac{u: \diamond v, v: \varphi, \Gamma \vdash \Delta}{u: \overparen{\diamond} \varphi, \Gamma \vdash \Delta}[: \vec{\diamond} \mathrm{L}] v \text { new } \\
& \frac{\Gamma \vdash \Delta, u: \overrightarrow{\diamond v} \quad \Gamma \vdash \Delta, v: \varphi}{\Gamma \vdash \Delta, u: \diamond \varphi}[: \vec{\diamond} \mathrm{R}] \\
& \frac{\Gamma \vdash \Delta, v: \overleftrightarrow{\diamond} u \quad v: \varphi, \Gamma \vdash \Delta}{u: \overleftarrow{\square} \varphi, \Gamma \vdash \Delta}[: \overleftarrow{\square} \mathrm{L}] \\
& \frac{v: \vec{\diamond} u, \Gamma \vdash \Delta, v: \varphi}{\Gamma \vdash \Delta, u: \overleftarrow{\square} \varphi}[: \overleftarrow{R}] v \text { new } \\
& \frac{u: v \bullet w, v: \varphi, w: \psi, \Gamma \vdash \Delta}{u: \varphi \bullet \psi, \Gamma \vdash \Delta}[: \bullet L] v, w \text { new } \\
& \frac{\Gamma \vdash \Delta, u: v \bullet w \quad \Gamma \vdash \Delta, v: \varphi \quad \Gamma \vdash \Delta, w: \psi}{\Gamma \vdash \Delta, u: \varphi \bullet \psi}[: \bullet R] \\
& \frac{v: \varphi, \Gamma \vdash \Delta \quad \Gamma \vdash \Delta, v: u \bullet w \quad \Gamma \vdash \Delta, w: \psi}{u: \varphi \leftarrow \psi, \Gamma \vdash \Delta}[: \leftarrow \mathrm{L}] \quad \frac{v: u \bullet w, w: \psi, \Gamma \vdash \Delta, v: \varphi}{\Gamma \vdash \Delta, u: \varphi \leftarrow \psi}[: \leftarrow \mathrm{R}] v, w \text { new } \\
& \frac{v: \psi, \Gamma \vdash \Delta \quad \Gamma \vdash \Delta, v: w \bullet u \quad \Gamma \vdash \Delta, w: \varphi}{u: \varphi \rightarrow \psi, \Gamma \vdash \Delta}[: \rightarrow \mathrm{L}] \quad \frac{v: w \bullet u, w: \varphi, \Gamma \vdash \Delta, v: \psi}{\Gamma \vdash \Delta, u: \varphi \rightarrow \psi}[: \rightarrow \mathrm{R}] v, w \text { new } \\
& \frac{v: \varphi, \Gamma \vdash \Delta}{u: v: \varphi, \Gamma \vdash \Delta}[\because: \mathrm{L}] \\
& \frac{u: \varphi[u / v], \Gamma \vdash \Delta}{u: \downarrow_{v} \varphi, \Gamma \vdash \Delta}[: \downarrow \mathrm{L}] \\
& \frac{\Gamma \vdash \Delta, v: \varphi}{\Gamma \vdash \Delta, u: v: \varphi}[\because R] \\
& \frac{\Gamma \vdash \Delta, u: \varphi[u / v]}{\Gamma \vdash \Delta, u: \downarrow_{v} \varphi}[: \downarrow \mathrm{R}] \\
& \frac{v: \varphi, \Gamma \vdash \Delta}{u: \mathrm{E} \varphi, \Gamma \vdash \Delta}[: \mathrm{EL}] v \text { new } \\
& \frac{\Gamma \vdash \Delta, v: \varphi}{\Gamma \vdash \Delta, u: \mathrm{E} \varphi}[: \mathrm{ER}]
\end{aligned}
$$

## Labeled logical rules LLR

$$
\begin{array}{cc}
\overline{u: p, \Gamma \vdash \Delta, p u}\left[\mathrm{Ax}_{p} \mathrm{~L}\right] & \overline{p u, \Gamma \vdash \Delta, u: p}\left[\mathrm{Ax}_{p} \mathrm{R}\right] \\
\overline{u: \overrightarrow{\diamond v}, \Gamma \vdash \Delta, r_{\diamond} u v}\left[\mathrm{Ax}_{\diamond} \mathrm{L}\right] & \overline{r_{\diamond} u v, \Gamma \vdash \Delta, u: \overrightarrow{\diamond v}}\left[\mathrm{Ax}_{\diamond} \mathrm{R}\right] \\
\overline{u: v \bullet w, \Gamma \vdash \Delta, r_{\bullet} u v w}\left[\mathrm{Ax}_{\bullet} \mathrm{L}\right] & \overline{r_{\bullet} u v w, \Gamma \vdash \Delta, u: v \bullet w}[\mathrm{Ax} \bullet \mathrm{R}]
\end{array}
$$

Interface axioms IA
Figure 1.2: Labeled sequent calculus for HCTL
becomes
where $\pi_{1}^{*}[w / v]$ is the result of renaming the occurrences of $v$ by $w$ in $\pi_{1}$. Note that such replacement leaves the contexts $\Gamma$ and $\Delta$ unchanged since $w$ does not occur in them.
$\mathbf{S}+\mathbf{L L R}+\mathbf{I A}$ brings us very close to a real hybrid calculus for validity in HCTL. From the previous results, we know that since $\mathbf{S}+\mathbf{L L R}+\mathbf{I A}$ is a calculus for $\Theta$-validity, $\Gamma \vdash \Delta$ is valid in HCTL iff $u: \Gamma \vdash u: \Delta$ is a theorem of $\mathbf{S}+\mathbf{L L R}+\mathbf{I A}$. But we can go a step further. The LLR rules that we derived in our search for cut elimination, are actually rules for the logical operators in HCTL. Since the calculus $\mathbf{S}+\mathbf{L L R}+\mathbf{I A}$ has the subformula property, the proof of $u: \Gamma \vdash u: \Delta$ in $\mathbf{S}+\mathbf{L L R}+\mathbf{I A}$ will not require the rules for the logical operators of $\mathbf{S}$ or the interface axioms IA. We only need to keep the structural rules of $\mathbf{S}$ and the rules of LLR and we will still have a sound and complete calculus for HCTL. Let's call such a calculus HS, i.e., HS has as rules the structural rules of $\mathbf{S}$ and the rules of LLR.

Even more, up to now, we have been using a FO calculus to prove HCTL sequents 'indirectly', and the symbol : in the rules of $\mathbf{S}$ was a metasymbol of FO. Now everything is in place for the internalization flip:

From now on, consider that the symbol : in each rule of HS is the HCTL connective, and think of the rules as treating directly HCTL sequents.

Note that the connective : allows us to easily capture in the formal language of HCTL the labels that we added to formulas in the metalanguage of FO. Without this operator, it is unclear how this would be achieved. This is precisely what makes hybrid logics so suitable for the study of internalization.

Theorem 2. A sequent $\Gamma \vdash \Delta$ of HCTL is valid iff $u: \Gamma \vdash u: \Delta$ is a theorem of HS.

## 5 A Fully Internalized Sequent Calculus for HCTL

HS is cut-free and enjoys the subformula property (with a suitable definition of subformula). It is also internalized, since no metalogical operators are required and only formulas of HCTL occur in proofs. However, it still has a minor drawback. It only covers a fragment of the language, namely we can prove only sequents where all formulas are of the form $u: \varphi$. Such a calculus is called labeled or :-driven. In practice this is not a big problem, since we know that a sequent $\Gamma \vdash \Delta$ of HCTL is valid iff the sequent $u: \Gamma \vdash u: \Delta$ is valid, where $u$ is any arbitrary variable not occurring in $\Gamma \vdash \Delta$. But it would be nice if we could
deal directly with arbitrary sequents of HCTL. This is easily achieved using nominals, the individual variables occurring as formulas in HCTL.

Recall that a nominal is true only when it is evaluated at the point it denotes, so a sequent $u, \Gamma \vdash \Delta$ will only be valid if it is evaluated at the point denoted by $u$ and the sequent $\Gamma \vdash \Delta$ is true at this point. Thus when a single nominal occurs on the left hand side of a sequent, it anchors all non :-prefixed formulas to the same element, and hence they do not need to explicitly share a prefix. In Figure 1.3 we give the so called nominal rules NR, which control the interaction between :-prefixed formulas and nominals. We will use the context sharing conditions given by the nominals to do some cleaning up. We will rewrite our rules as follows:

$$
\frac{u: \Gamma_{1}, \Gamma \vdash \Delta, u: \Delta_{1}}{u: \Gamma_{2}, \Gamma^{\prime} \vdash \Delta^{\prime}, u: \Delta_{2}}[\text { rule }] \rightsquigarrow \frac{u, \Gamma_{1}, \Gamma \vdash \Delta, \Delta_{1}}{u, \Gamma_{2}, \Gamma^{\prime} \vdash \Delta^{\prime}, \Delta_{2}}[\text { rule }] .
$$

If additionally $u$ does not occur in the rest of the rule, we can delete it. ${ }^{2}$ This yields the nominal-based hybrid rules HR, given in Figure 1.3. Note that some of the rules of LLR become redundant when the :-prefixes are removed, so we do not include them in HR. Let's call NHS the calculus shown in Figure 1.3 comprising the structural rules of S, the nominal rules NR, and the nominal-based hybrid rules HR.

Now we have finally achieved our goal: NHS is a sound and complete, cut free, fully internalized calculus for HCTL.

Theorem 3. A sequent of HCTL is valid iff it is a theorem of NHS.
Proof. First we prove that $u: \Gamma \vdash u: \Delta$ is a theorem of HS iff $\Gamma \vdash \Delta$ can be proved using HS and the nominal rules NR. For the only if direction, if we can prove $u: \Gamma \vdash u: \Delta$ in HS, then by weakening we get $u, u: \Gamma \vdash u: \Delta$ and then, by applying repeatedly the $[\wedge: \mathrm{L}]$ and $[\wedge: R]$ rules, we get $u, \Gamma \vdash \Delta$. Finally, using [name] we get a proof of $\Gamma \vdash \Delta$. For the other direction, since all the rules of NR are sound, all theorems that can be proved with HS and NR are valid in HCTL. In a second step, it must be verified that any proof of NHS can be translated into a proof using the rules in HS and NR, and vice versa. The translation is straightforward: any occurrence of a rule [:rule] in a proof of HS can be replaced by its equivalent rule [rule] and the other way around. In both cases, some transformations on the sequent must be done using the nominal and structural rules, but the resulting proof is isomorphic to the original one.

## 6 Examples

As an example, we give two short derivations in the calculus NHS. The first one corresponds to a sequent with both hybrid connectives and CTL formulas, and the second one to a sequent containing only CTL connectives. On the left column we prove the sequent $\vec{\diamond} \varphi \vdash \mathrm{E} \varphi$. If $\vec{\diamond} \varphi$ is true at some point $a$ in $\mathcal{M}$, then there is some $b$ in $\mathcal{M}$ (connected to $a$

[^34]by the $r_{\diamond}$ relation) where $\varphi$ holds, thus $\mathrm{E} \varphi$ holds in $a$ as well. On the right we prove that $\varphi \vdash \overleftarrow{\square} \vec{\diamond} \varphi$. Since $\overleftarrow{\square}$ and $\vec{\diamond}$ form a residuated pair, $\overleftarrow{\square} \vec{\diamond} \varphi$ holds in any world where $\varphi$ holds. Note that, although we are proving a theorem of CTL, hybrid operators are used during the proof.

## 7 Conclusions and Future Work

Despite the peculiarities of CTL, the internalization strategy works fine and it yields an elegant, fully internalized calculus for HCTL. Note that the rules governing the modal operators in NHS contain also hybrid operators. This is a common feature of calculi for hybrid modal logics. We can see this as evidence that, in the absence of the hybrid machinery, it might be difficult to devise a fully internalized calculus. We also want to point out that the calculus NHS is given for sequents of the form $\Gamma \vdash \Delta$ where both $\Gamma$ and $\Delta$ are lists of formulas. An intuitionistic version of the calculus (which is more natural in the context of categorial logics) can be obtained in a straightforward way, by restricting the right-hand-side of sequents to contain at most one formula.

It remains as future work to analyze in more detail the calculus we just obtained, and to compare it with other calculi. For example, in (Areces and Bernardi 2003) CTL has already been extended with hybrid operators in order to model some binding phenomena in natural language. The calculus given there is a labeled calculus. It would be interesting to compare it with our calculus HS, and even to explore the nominal technique to eliminate :-labels on it. Another issue that must be explored is a comparison of the calculus NHS with Moortgat's calculus $\mathrm{NL}(\diamond)$. It is clear that we do not require any structural connectives, due to the expressiveness of the hybrid operators. Another relevant difference is that in our calculus sequents have more than one formula on the left-hand-side. This multiformula sequents imply a restricted form of conjunction, which is usually not present in CTL. Additionally, hybrid operators allow to simulate by themselves and to some extent, the behavior of Boolean operators. For example, if the formula $(w: p) \bullet(w: q)$ holds at some point of a model $\mathcal{M}$, then there is some point in $\mathcal{M}$ (namely, the point named $w$ ), where the conjunction of $p$ and $q$ holds. Similar formulas can be built that simulate other Boolean constructors at named worlds. Clearly HCTL is more expressive than ordinary CTL, but it remains as an open question to find out how much.

$$
\overline{\varphi, \Gamma \vdash \Delta, \varphi}[\mathrm{Ax}] \quad \frac{\Gamma \vdash \Delta}{\Gamma^{\prime} \vdash \Delta^{\prime}}[\mathrm{S}]^{1}
$$

${ }^{1}$ if the lists $\Gamma$ and $\Delta$ contain the same set of formulas as $\Gamma^{\prime}$ and $\Delta^{\prime}$ resp.

## Structural rules

$$
\begin{array}{cl}
\frac{u, \varphi, \Gamma \vdash \Delta}{u, u: \varphi, \Gamma \vdash \Delta}[\mathrm{V}: \mathrm{L}] & \frac{u, \Gamma \vdash \Delta, u: \varphi}{u, \Gamma \vdash \Delta, \varphi}[\mathrm{~V}: \mathrm{R}] \\
\frac{u, u: \varphi, \Gamma \vdash \Delta}{u, \varphi, \Gamma \vdash \Delta}[\wedge: \mathrm{L}] & \frac{u, \Gamma \vdash \Delta, \varphi}{u, \Gamma \vdash \Delta, u: \varphi}[\wedge: \mathrm{R}] \\
\frac{u, \Gamma \vdash \Delta}{\Gamma \vdash \Delta}[\text { name }]^{1} & \frac{u, \Gamma \vdash \Delta}{\Gamma \vdash \Delta}[\text { term }]^{2} \frac{\Gamma \vdash \Delta}{u, \Gamma \vdash \Delta}\left[\mathrm{term}^{-}\right]^{2}
\end{array}
$$

${ }^{1}$ if $u$ does not occur in $\Gamma, \Delta \quad{ }^{2}$ if all formulas in $\Gamma, \Delta$ are :-prefixed

## Nominal rules NR

$$
\begin{aligned}
& \frac{u, v, \Gamma[w / u] \vdash \Delta[w / u]}{u, v, \Gamma[w / v] \vdash \Delta[w / v]}[N \mathrm{~L}] \\
& \frac{\vec{\diamond} v, v: \varphi, \Gamma \vdash \Delta}{\diamond}[\vec{\diamond} \varphi, \Gamma \vdash \Delta \quad v \text { new } \\
& \frac{\Gamma \vdash \Delta, \stackrel{\rightharpoonup}{\diamond} u \quad \varphi, \Gamma \vdash \Delta}{u: \overleftarrow{\square} \varphi, \Gamma \vdash \Delta}[\overleftarrow{\square} \mathrm{L}] \\
& \frac{v \bullet w, v: \varphi, w: \psi, \Gamma \vdash \Delta}{\varphi \bullet \psi, \Gamma \vdash \Delta}[\bullet \mathrm{L}] v, w \text { new } \\
& \frac{\varphi, \Gamma \vdash \Delta \quad \Gamma \vdash \Delta, u \bullet w \quad \Gamma \vdash \Delta, w: \psi}{u: \varphi \leftarrow \psi, \Gamma \vdash \Delta}[\leftarrow \mathrm{L}] \\
& \frac{\Gamma \vdash \Delta, \vec{\diamond} v \quad \Gamma \vdash \Delta, v: \varphi}{\Gamma \vdash \Delta, \vec{\diamond} \varphi}[\vec{\diamond} \mathrm{R}] \\
& \frac{\vec{\diamond} u, \Gamma \vdash \Delta, \varphi}{\Gamma \vdash \Delta, u: \overleftarrow{ } \overleftarrow{\square}}[\overleftarrow{\square}] \\
& \frac{\Gamma \vdash \Delta, v \bullet w \quad \Gamma \vdash \Delta, v: \varphi \quad \Gamma \vdash \Delta, w: \psi}{\Gamma \vdash \Delta, \varphi \bullet \psi}[\bullet \mathrm{R}] \\
& \frac{u \bullet w, w: \psi, \Gamma \vdash \Delta, \varphi}{\Gamma \vdash \Delta, u: \varphi \leftarrow \psi}[\leftarrow \mathrm{R}] w \text { new } \\
& \frac{\psi, \Gamma \vdash \Delta \quad \Gamma \vdash \Delta, w \bullet u \quad \Gamma \vdash \Delta, w: \varphi}{u: \varphi \rightarrow \psi, \Gamma \vdash \Delta}[\rightarrow \mathrm{L}] \\
& \frac{w \bullet u, w: \varphi, \Gamma \vdash \Delta, \psi}{\Gamma \vdash \Delta, u: \varphi \rightarrow \psi}[\rightarrow \mathrm{R}] w \text { new } \\
& \frac{u, \varphi[u / i], \Gamma \vdash \Delta}{u, \downarrow_{i} \varphi, \Gamma \vdash \Delta}[\downarrow \mathrm{~L}] \\
& \frac{u, \Gamma \vdash \Delta, \varphi[u / i]}{u, \Gamma \vdash \Delta, \downarrow_{i} \varphi}[\downarrow \mathrm{R}] \\
& \frac{v: \varphi, \Gamma \vdash \Delta}{\mathrm{E} \varphi, \Gamma \vdash \Delta}[\mathrm{EL}] v \text { new } \\
& \frac{\Gamma \vdash \Delta, v: \varphi}{\Gamma \vdash \Delta, \mathrm{E} \varphi}[\mathrm{ER}]
\end{aligned}
$$

Figure 1.3: Internalized sequent calculus NHS for HCTL

## References

Areces, C. and R. Bernardi (2003, September). In situ binding: A hybrid approach. In Inference in Computational Semantics 4-ICoS-4, Nancy, France, Nancy, France.

Areces, C., P. Blackburn, and M. Marx (2001). Hybrid logics: characterization, interpolation and complexity. J. Symbolic Logic 66(3), 977-1010.

HyLo (2006). Hybrid logic's webpages. http://www.hylo.net.
Lambek, J. (1961). On the calculus of syntactic types. In R. Jakobson (Ed.), Structure of Languages and its Mathematical Aspects, pp. 166-178. American Mathematical Society.
Moortgat, M. (1996). Multimodal linguistic inference. Journal of Logic, Language and Information 5(3,4), 349-385.
Seligman, J. (2001). Internalization: The case of hybrid logics. Journal of Logic and Computation 11 (5), 671-689. Special Issue on Hybrid Logics. Areces, C. and Blackburn, P. (eds.).

# Semantic Cut-elimination for Two Explicit Modal Logics 

Bryan Renne<br>Computer Science, CUNY Graduate Center<br>bryan@renne.org


#### Abstract

Explicit modal logics contain modal-like terms that label formulas in a way that mimics deduction in the system. These logics have certain proof-theoretic advantages over the usual modal logics, perhaps the most important of which is conventional cut-elimination.

The present paper studies tableau proof systems for two explicit modal logics, LP and S4LP. Using a certain method to prove the correctness of these systems, we obtain a semantic proof of cut-elimination for these logics.


## 1 Introduction

Explicit modal logics differ from ordinary modal logic in that the former introduce formulalabeling terms into the language of propositional logic. These terms label formulas in a way that mimics deduction in the system, so the terms may be thought of as reasons or evidence as to why a formula holds (or is known). In this approach, if $t$ is such a term and $\varphi$ is a formula, then $t: \varphi$ is a new formula whose epistemic reading is " $\varphi$ is known for reason $t$." Compare this with the epistemic reading of the usual modal formula $\square \varphi$ : " $\varphi$ is known (for some reason)."

The present paper studies two explicit modal logics. The first is the the most elementary explicit modal logic, the Logic of Proofs (LP). The second is S4LP, whose language extends that of LP by introducing an S4 modality. We define tableau systems for both LP and S4LP, and prove the correctness (soundness and completeness) of these systems. As a corollary of the completeness argument, we also obtain a semantic proof of cut-elimination for both LP and S4LP. In the case of S4LP, this answers affirmatively the question left open by Fitting in (Fitting 2004) as to whether S4LP is cut-free.

We now present the Hilbert-style theories LP and S4LP.

## 2 The Syntax

### 2.1 The Logic LP

The language of LP extends that of propositional logic by introducing a countable collection of proof variables $x_{1}, x_{2}, x_{3}, \ldots$, a countable collection of proof constants $c_{1}, c_{2}, c_{3}, \ldots$, the binary function symbols + and $\cdot$, and the unary function symbol !. Proof terms are built up from proof variables and proof constants using the function symbols. The rules of formula formation are those of propositional logic in addition to the following: if $t$ is a proof term and $\varphi$ is an LP formula, then $t: \varphi$ is also an LP formula. Note that proof terms will sometimes be called evidence terms or perhaps even terms. Now, letting $t$ and $s$ be arbitrary terms and $\varphi$ and $\psi$ be arbitrary formulas, the theory LP consists of the following axiom and rule schemas:

## - Propositional logic

PL. A finite collection of axiom schemas for propositional logic
RPL. Modus ponens: infer $\psi$ from $\varphi \supset \psi$ and $\varphi$

- Evidence management

LP1. $t:(\varphi \supset \psi) \supset(s: \varphi \supset(t \cdot s): \psi)$
LP2. $t: \varphi \supset!t:(t: \varphi)$
LP3. $t: \varphi \vee s: \varphi \supset(t+s): \varphi$
LP4. $t: \varphi \supset \varphi$
RLP. Constant necessitation: infer $c: A$ for $A$ an axiom and $c$ a proof constant
LP1 is the property of application for evidence terms, which is an internalized modus ponens: if $t$ is evidence for an implication and $s$ is evidence for the antecedent, then $t \cdot s$ is evidence for the consequent. LP2 is the property of proof checking: if $t$ is evidence for $\varphi$, then ! (read "bang $t$ ") is evidence for the fact that $t: \varphi$, so $!t$ verifies that indeed $t$ is evidence for $\varphi$. LP3 is a sum or monotonicity property: if $t$ is evidence for $\varphi$, then combining $t$ with the information in $s$ to produce either $t+s$ or $s+t$ yields something that is still evidence for $\varphi \cdot{ }^{1} \mathbf{L P} 4$ is an explicit reflection property: if $t$ is evidence for $\varphi$, then $\varphi$ must be true. RLP says that the proof constants are atomic reasons for the most basic facts, the axioms. Since constants serve as justification for our basic facts, they may be viewed as the simplest sort of justification.

The following internalization property can be shown by induction on the length of the derivation in LP: for every LP theorem $\varphi$, there is an evidence term $t$ containing no proof variables such that $t: \varphi$ is an LP theorem. This provides a sense in which LP encodes its own derivations using evidence terms, which bolsters the intuitive conception of terms as reasons or evidence.

[^35]
### 2.2 The Logic S4LP

The language of S4LP extends that of LP by adding a unary S4 modality $\square$. The rules of formula formation are those of LP in addition to the following: if $\varphi$ is an S4LP formula, then so is $\square \varphi$. Now, letting $t$ be an arbitrary term and $\varphi$ and $\psi$ be arbitrary formulas, the theory S4LP consists of the axiom and rule schemas of LP in addition to the following:

- S4

K1. $\square(\varphi \supset \psi) \supset(\square \varphi \supset \square \psi)$
K2. $\square \varphi \supset \varphi$
K3. $\square \varphi \supset \square \square \varphi$
RK. $\square$ necessitation: infer $\square \varphi$ from $\varphi$

- Connection Principle
C. $t: \varphi \supset \square \varphi$

Assigning $\square \varphi$ the epistemic reading, " $\varphi$ is known," the Connection Principle can be read, "If $\varphi$ is known for a reason, then $\varphi$ is known." Note that the internalization property also holds of S4LP.

## 3 The Semantics

LP has an arithmetic semantics (Artemov 2001), a minimal semantics (Mkrtychev 1997), a Kripke-style semantics (Fitting 2003; Artemov 2006; Artemov 2004; Fitting 2005), and a game semantics (Renne 2006). In this paper, we will make use of the Kripke-style semantics - otherwise known in this area as the Fitting semantics - because this semantics also interprets S4LP.

### 3.1 The Fitting Semantics

A model in the Fitting semantics consists of an S4 Kripke model ${ }^{2}(G, R, V)$ together with a certain mapping $\mathcal{E}$ from worlds and terms to sets of formulas, with the intent that $\mathcal{E}(\Gamma, t)$ is the set of formulas for which $t$ serves as evidence at world $\Gamma$. For convenience, call a formula $\varphi$ knowable at a world $\Gamma$ if $\varphi$ is true at all worlds accessible from $\Gamma$; that is, $\Gamma R \Delta$ implies $\varphi$ is true at $\Delta$. We then say that a formula of the form $t: \varphi$ is true at $\Gamma$ if $t$ is evidence for $\varphi$-that is, $\varphi \in \mathcal{E}(\Gamma, t)$-and $\varphi$ is also knowable at $\Gamma$. Now for the details.

A model $M$ is a tuple $(G, R, \mathcal{E}, V)$, where $(G, R, V)$ is an S4 Kripke model and $\mathcal{E}$ is an evidence function. An evidence function is a map from worlds and terms to sets of formulas that satisfies each of the following properties:

[^36]- Evidence Closure
- Application: $\varphi \supset \psi \in \mathcal{E}(t)$ and $\varphi \in \mathcal{E}(s)$ implies $\psi \in \mathcal{E}(t \cdot s)$
- Proof Checker: $\varphi \in \mathcal{E}(t)$ implies $t: \varphi \in \mathcal{E}(!t)$
- Sum: $\mathcal{E}(t) \cup \mathcal{E}(s) \subseteq \mathcal{E}(t+s)$
- Constant Specification: $A \in \mathcal{E}(c)$ for $A$ an axiom and $c$ a proof constant
- Evidence Monotonicity: $\Gamma R \Delta$ implies $\mathcal{E}(\Gamma, t) \subseteq \mathcal{E}(\Delta, t)$ for every term $t$

Truth of a formula $\varphi$ in $M$ is then defined by induction on the complexity of $\varphi$, where the propositional cases are handled as is usual. A formula of the form $\square \psi$ is said to be true at a world $\Gamma$ in $M$ whenever $\psi$ is knowable at $\Gamma$. A formula of the form $t: \psi$ is said to be true at $\Gamma$ whenever $\psi \in \mathcal{E}(\Gamma, t)$ and $\psi$ is knowable at $\Gamma$. Notation: $M, \Gamma \models \varphi$ means that $\varphi$ is true at $\Gamma$ in $M$ and $M, \Gamma \not \models \varphi$ means that $\varphi$ is not true at $\Gamma$ in $M$. In addition, $M \models \varphi$ means that $M, \Gamma \models \varphi$ for every $\Gamma$ in $M$. As usual, a formula $\varphi$ is said to be valid if $M \models \varphi$ for every model $M$.

### 3.2 Artemov's Extension

In an S4LP model ( $G, R, \mathcal{E}, V$ ), the S4 modality and the evidence terms both use the same relation $R$ in their interpretation. Artemov observed in (Artemov 2006) that this need not be the case. In particular, S4LP may also be modeled by tuples $\left(G, R, R_{e}, \mathcal{E}, V\right)$, where $R_{e}$ is a new reflexive and transitive binary relation on $G$ satisfying $R \subseteq R_{e}$ (the other items of the tuple are as before). In these models, truth of formulas $t: \varphi$ is given by the relation $R_{e}$-as are the Evidence Closure and Evidence Monotonicity conditions-while truth of formulas $\square \varphi$ is given by the relation $R$. Notice that the condition $R \subseteq R_{e}$ guarantees that such models satisfy the Connection Principle. The basic Fitting semantics of the previous subsection (Section 3.1) is obtained by taking $R=R_{e}$.

Artemov's extension allows a greater degree of flexibility in modeling. In particular, in such models the modality $\square$ need not be an S4 modality. Artemov's extension accordingly allows us to extend LP so that the extension incorporates various multi-modal logics with unary modalities $\square_{i}$ (indexed by the subscript $i$ ) and corresponding interpreting relations $R_{i}$, each of which satisfies $R_{i} \subseteq R_{e}$. For example, $\mathrm{S}_{n} \mathrm{LP}$ is a logic which has S 5 modalities $\square_{1}, \ldots, \square_{n}$, each of which satisfies the Connection Principle $t: \varphi \supset \square_{i} \varphi$. Similarly, we have logics $T_{n} L P, S 4_{n} L P$, and various mixed logics such as S4S5LP (where $\square_{1}$ is $S 4, \square_{2}$ is $S 5$, and each satisfies the Connection Principle).

## 4 The Tableau Systems

In a technical report (Renne 2004), the author defined a tableau system for LP. ${ }^{3}$ This was extended by Fitting in (Fitting 2004) to S4LP and proved complete with respect to the class

[^37]of models with $R=R_{e}$, though Fitting's system is not cut-free. By a slight modification of Fitting's system, we obtain a system that is cut-free.

We first recall the author's LP tableau system and then show how it is extended to a cut-free system for S4LP. We will then prove that the S4LP system is sound and complete with respect to the class of Fitting models where $R \subseteq R_{e}$. That S4LP is cut-free follows as a consequence of completeness, but more on this later.

### 4.1 The LP Tableau System

A tableau for (or beginning with) $\varphi$ is a tree with $\varphi$ at the root constructed by nondeterministically applying a branch extension rule, called a tableau rule. ${ }^{4}$ The tableau rules for LP are of three basic types: non-branching (otherwise known as $\alpha$ ), branching (otherwise known as $\beta$ ), and $\psi$-branching (otherwise known as $\beta^{\psi}$, where $\psi$ is an arbitrary LP formula). Each of these types will be described shortly. A branch of a tableau is said to be closed if it satisfies at least one of the following three conditions:

1. the branch contains both $\varphi$ and $\neg \varphi$ for some formula $\varphi$,
2. the branch contains $\perp$, or
3. the branch contains $\neg(c: A)$ for $c$ a proof constant and $A$ an axiom.

If every branch of a tableau is closed, the tableau itself is said to be closed. A branch or tableau that is not closed is called open. A tableau proof of a formula $\varphi$ is a closed tableau beginning with $\neg \varphi$. We now give the tableau rules for LP.

The classical tableau rules are given in Figure 1.1. In this figure, the left rule is an $\alpha$. For convenience, we follow Smullyan's naming convention of (Smullyan 1963): the formula above the line is referred to as $\alpha$ and the formulas below the line are called $\alpha_{1}$ (the top formula) and $\alpha_{2}$. An $\alpha$ rule allows any branch on which $\alpha$ appears to be extended by adding either $\alpha_{1}$ or $\alpha_{2}$ to the end of the branch.

$$
\frac{\neg(\varphi \supset \psi)}{\varphi} \quad \frac{\varphi \supset \psi}{\neg \varphi \mid \psi}
$$

Figure 1.1: Tableau rules for classical logic.
The rule on the right in Figure 1.1 is a $\beta$. Following a similar naming convention as in the $\alpha$ case, the formula above the (horizontal) line is referred to as $\beta$ and the formulas below the line are called $\beta_{1}$ (the left formula) and $\beta_{2}$. A $\beta$ rule allows any branch on which $\beta$ appears to be extended by splitting at the end so that both $\beta_{1}$ and $\beta_{2}$ are new leaves.

[^38]Following the diagrammatic conventions suggested in Figure 1.1 for designating $\alpha$ rules, the tableau rules for the LP evidence operations are given in Figure 1.2. In this figure, the leftmost $\alpha$ rule only produces one formula $\alpha_{1}$, so we will thus adopt the convention that $\alpha_{1}$ and $\alpha_{2}$ name the same formula in such a situation.

$$
\frac{t: \varphi}{\varphi} \quad \frac{\neg(!t:(t: \varphi))}{\neg(t: \varphi)} \quad \frac{\neg((s+t): \varphi)}{\neg(s: \varphi)} \quad \begin{gathered}
\neg(s:(\psi \supset \varphi)) \mid \neg(t: \psi)
\end{gathered}
$$

Figure 1.2: Tableau rules for the LP evidence operations.
The rule at the far right in Figure 1.2 is rather odd, since the formula $\psi$ is arbitrary and, in particular, need not even appear as a subformula of $\neg((s \cdot t): \varphi)$. So, while this rule has the diagrammatic form of a $\beta$ rule, a formula $\psi$ must be given as a parameter in order to specify the formulas below the line. This rule is thus called a $\beta^{\psi}$ rule, where $\psi$ may be any formula, and the formulas below the line are then called $\beta_{1}^{\psi}$ (the left formula) and $\beta_{2}^{\psi}$. As is the case for $\beta$ formulas, a branch on which a $\beta^{\psi}$ formula occurs may be extended by splitting at the end so that both $\beta_{1}^{\psi}$ and $\beta_{2}^{\psi}$ are new leaves.

This completes the specification of the tableau system for LP. This system notably omits the cut rule, which is given in Figure 1.3. For any formula $\varphi$, the cut rule allows any branch to be extended by splitting at the end so that both $\varphi$ and $\neg \varphi$ are new leaves. While the LP tableau system is cut-free (that is, without cut), the system does not have the subformula property due to the presence of the $\beta^{\psi}$ rule. ${ }^{5}$

$$
\begin{array}{l|l}
\hline \varphi & \neg \varphi
\end{array}
$$

Figure 1.3: The (tableau) cut rule. This rule is not a part of the LP tableau system.

### 4.2 The S4LP Tableau System

The tableau system for S4LP is obtained from the system for LP by adding a rule corresponding to the Connection Principle along with rules to handle formulas of the form $\square \varphi$ and of the form $\neg \square \varphi$. These rules are given in Figure 1.4. In this figure, the rightmost rule is a new rule type, known as a destructive rule. A destructive rule modifies the tableau, in this case deleting a branch $S$ containing the formula $\neg \square \varphi$ and adding to the tableau a new branch consisting of those formulas in $S^{\#}$ along with the formula $\neg \varphi$. This deletion operation can be implemented in various ways. A simple implementation is to mark all formulas on the branch (including $\neg \square \varphi$ ) as "deleted" for this branch and then extend the

[^39]branch by adding $\neg \varphi$ and each of the formulas in $S^{\#}$ at the branch end (in any order). ${ }^{6}$ Tableau rules are then restricted so that a tableau rule may be applied to a formula $\psi$ appearing on a branch $\theta$ only if $\psi$ is not marked as "deleted" for $\theta$. And, as the reader might expect, a branch $\theta$ is closed only if one of the closure conditions applies to formulas not marked as "deleted" for $\theta$.
\[

$$
\begin{aligned}
& \frac{t: \varphi}{\square \varphi} \quad \frac{\square \varphi}{\varphi} \quad \frac{S, \neg \square \varphi}{S^{\#}, \neg \varphi} \\
S^{\#}:= & \{\square \psi \mid \square \psi \in S\} \cup\{t: \psi \mid t: \psi \in S\}
\end{aligned}
$$
\]

Figure 1.4: Additional tableau rules for S4LP.

## 5 Correctness of the Tableau Systems

We have presented two tableau systems, one for LP and another for S4LP. We will show the correctness results (soundness and completeness) for the latter system with respect to the class of models satisfying $R \subseteq R_{e}$. We indicate later how these correctness results may be modified to handle the LP system. So let us proceed with the proof of correctness of the S4LP system.

The proof of soundness is facilitated by a couple of definitions and a standard lemma. In particular, an open branch $\theta$ on a tableau is said to be satisfiable if there is a world $\Gamma$ of a model $M$ such that $M, \Gamma \models \varphi$ for every $\varphi \in \theta$ that is not "deleted" for $\theta$. A tableau is then said to be satisfiable if it has a satisfiable open branch. The following lemma, whose proof is standard and is thus omitted, then leads easily to the proof of soundness.

Lemma 1. If $\tau$ is a satisfiable tableau, any tableau produced from $\tau$ by application of a tableau rule is also satisfiable.

Theorem 1 (Soundness). If a formula $\varphi$ has a tableau proof, then $\varphi$ is valid.
Proof. If $\varphi$ is not valid, there is a world $\Gamma$ of a model $M$ such that $M, \Gamma \not \models \varphi$. Thus $M, \Gamma \models \neg \varphi$ and so $\neg \varphi$ is satisfiable. By Lemma 1 , no tableau beginning with $\neg \varphi$ can close, so $\varphi$ does not have a tableau proof.

Completeness of the tableau system uses a canonical model construction, with maximal consistency defined relative to the tableau system. Care is taken to avoid implicit use of the cut rule (which is not present in the system), but more on this later.

Since all of the sets mentioned in the remainder of the paper are sets of formulas, a set is assumed to be a set of formulas. Now let $S$ be a set. $S$ is said to be consistent if for no finite subset $S^{\prime}$ does a tableau beginning with $S^{\prime}$ close. If $S$ is not consistent, it is

[^40]called inconsistent. A consistent $S$ is then maximal consistent if adding any formula to $S$ that is not already present produces an inconsistent set. It is a well-known fact that every consistent set can be extended to a maximal consistent set. What's more, any maximal consistent set $S$ satisfies each of the following properties:

- $\alpha \in S$ implies both $\alpha_{1} \in S$ and $\alpha_{2} \in S$,
- $\beta \in S$ implies $\beta_{1} \in S$ or $\beta_{2} \in S$, and
- $\beta^{\psi} \in S$ implies $\beta_{1}^{\psi}$ or $\beta_{2}^{\psi} \in S$.

A set satisfying each of these properties is called downward saturated, so every maximal consistent set is downward saturated.

We now begin the proof of completeness. After constructing the canonical model (and verifying that it is in fact a model satisfying $R \subseteq R_{e}$ ), we verify a useful lemma known as the Truth Lemma. Completeness of the tableau system follows almost immediately from the Truth Lemma. Proceeding, we construct the canonical model $M=\left(G, R, R_{e}, \mathcal{E}, V\right)$ as follows:

- $G$ is the collection of all maximal consistent sets
- $\Gamma R_{e} \Delta$ holds if and only if both $\{t: \psi \mid t: \psi \in \Gamma\} \subseteq \Delta$ and $\{\neg t: \psi \mid \neg t: \psi \in \Delta\} \subseteq \Gamma$
- $\Gamma R \Delta$ holds if and only if both $\Gamma R_{e} \Delta$ and $\{\square \psi \mid \square \psi \in \Gamma\} \subseteq \Delta$
- $\varphi \in \mathcal{E}(\Gamma, t)$ holds if and only if $\neg t: \varphi \notin \Gamma$
- $p \in V(\Gamma)$ holds if and only if $p \in \Gamma$

Lemma 2. The canonical model is a model.
Proof. It's clear that $R$ and $R_{e}$ are transitive and reflexive and $G$ is nonempty, so $(G, R, V)$ is an S4 Kripke model. It is also obvious that $R \subseteq R_{e}$. What remains is to show that $\mathcal{E}$ is an evidence function; that is, $\mathcal{E}$ satisfies Evidence Closure and Evidence Monotonicity.

The Evidence Closure conditions follow from the tableau rules and the fact that each world is maximal consistent. As an example, we check Application. So suppose that $\varphi \supset \psi \in \mathcal{E}(\Gamma, t)$ and that $\varphi \in \mathcal{E}(\Gamma, s)$. We then have $\neg t:(\varphi \supset \psi) \notin \Gamma$ and $\neg s: \varphi \notin \Gamma$ by the definition of $\mathcal{E}$. Note that $\neg t:(\varphi \supset \psi)$ has the form of a $\beta_{1}^{\varphi}$ and that $\neg s: \varphi$ has the form of a $\beta_{2}^{\varphi}$. Now, since $\Gamma$ is downward saturated, it cannot be the case that $\neg(t \cdot s): \psi \in \Gamma$, for otherwise the $\beta^{\varphi}$ rule applies and $\beta_{1}^{\varphi} \in \Gamma$ or $\beta_{2}^{\varphi} \in \Gamma$, a contradiction. Hence $\neg(t \cdot s): \psi \notin \Gamma$ and thus $\psi \in \mathcal{E}(\Gamma, t \cdot s)$ by the definition of $\mathcal{E}$.

We now show $\mathcal{E}$ satisfies Evidence Monotonicity. So assume $\Gamma R_{e} \Delta$ and $\varphi \in \mathcal{E}(\Gamma, t)$. From the definition of $\mathcal{E}$, we have $\neg t: \varphi \notin \Gamma$. It then follows from the meaning of $\Gamma R_{e} \Delta$ that $\neg t: \varphi \notin \Delta$, and thus $\varphi \in \mathcal{E}(\Delta, t)$, as desired.

Lemma 3 (Truth Lemma). Let $M=\left(G, R, R_{e}, \mathcal{E}, V\right)$ be the canonical model and $\Gamma$ be a world of $M$. Then each of the following holds:

- $\varphi \in \Gamma$ implies $M, \Gamma \models \varphi$, and
- $\neg \varphi \in \Gamma$ implies $M, \Gamma \not \vDash \varphi$.

Proof. By induction on $\varphi$. The base cases and propositional inductive case are standard, so we restrict our attention to the other inductive cases. We use without mention the fact that worlds of $M$ are downward saturated (and thus closed under $\alpha$-rule applications).

- Case $t: \varphi \in \Gamma$.

Since $\Gamma$ is consistent, $\neg t: \varphi \notin \Gamma$ and thus $\varphi \in \mathcal{E}(\Gamma, t)$. Now let $\Delta$ be an arbitrary world satisfying $\Gamma R_{e} \Delta$. By the definition of $R_{e}$, we have $t: \varphi \in \Delta$ and thus $\varphi \in \Delta$ by an $\alpha$ rule. By the induction hypothesis, we have $M, \Delta \models \varphi$. Since $\Delta$ was arbitrary, we have shown that $\varphi$ is knowable at $\Gamma$.

- Case $\neg t: \varphi \in \Gamma$.

By the definition of $\mathcal{E}$, we have $\varphi \notin \mathcal{E}(\Gamma, t)$ and thus $M, \Gamma \not \models t: \varphi$.

- Case $\square \varphi \in \Gamma$.

Let $\Delta$ be an arbitrary world satisfying $\Gamma R \Delta$. By the definition of $R$, we have $\square \varphi \in \Delta$ and thus $\varphi \in \Delta$ by an $\alpha$ rule. By the induction hypothesis, we have $M, \Delta \models \varphi$. Since $\Delta$ was arbitrary, we have shown that $\varphi$ is knowable at $\Gamma$.

- Case $\neg \square \varphi \in \Gamma$.

By the destructive tableau rule, we have that $\Gamma^{\#} \cup\{\neg \varphi\}$ is consistent. We may thus extend this union to a maximal consistent set $\Delta$ and we then have $\Delta \in G$. Since $\neg \varphi \in \Delta$, it follows from the induction hypothesis that $M, \Delta \not \vDash \varphi$. We have shown that $\varphi$ is not knowable at $\Gamma$.

Theorem 2 (Completeness). A valid formula $\varphi$ has a tableau proof.
Proof. Suppose $\varphi$ does not have a tableau proof, so $\{\neg \varphi\}$ is consistent and may thus be extended to a maximal consistent set $\Gamma$. This set $\Gamma$ is a world of the canonical model $M$ and we thus have $M, \Gamma \not \models \varphi$ by the Truth Lemma.

In (Artemov 2006), it is shown that the Hilbert-style theory of S4LP presented in Section 2 is sound and complete with respect to the class of Fitting models satisfying $R \subseteq R_{e}$. Since the same is true of the tableau system (with respect to the same semantics), the tableau system describes the same theory as do the Hilbert-style axiom and rule schemas.

Call a set $S$ downward closed if it is downward saturated and, in addition, $S$ contains either $\varphi$ or its negation $\neg \varphi$ for every formula $\varphi$. In a tableau system with cut, every maximal consistent set $S$ is downward closed. In such a system with cut, it can be shown that for any world $\Gamma$ of the canonical model $M$, we have $\varphi \in \Gamma$ if and only if $M, \Gamma \models \varphi$. The Truth Lemma may thus take a sharper form and is a bit easier to prove. However, the added difficulty of working in a cut-free tableau system provides us with an additional payoff: not only do we obtain completeness but we also verify the admissibility of cut in the S4LP tableau system.

Theorem 3. Cut is an admissible rule.
Proof. Cut is certainly a sound rule. Since the S4LP tableau system without cut is complete, every formula provable in the S4LP tableau system with cut is also provable without. But this is what it means to say that cut is an admissible rule.

That the tableau system for LP is correct follows by an appropriate restriction of the above completeness argument (soundness is identical).

Theorem 4 (Completeness). A valid formula $\varphi$ in the language of LP has an LP tableau proof.

Proof. Construct the canonical model as above, except set $R=R_{e}$ (and thus omit the condition $\{\square \psi \mid \square \psi \in \Gamma\} \subseteq \Delta$ ). The verification that this canonical model is a model is as before. In the proof of the Truth Lemma, the cases $\square \varphi$ and $\neg \square \varphi$ are omitted. Completeness is as before.

Theorem 5. Cut is an admissible rule in the LP tableau system.
Proof. As in the S4LP case.
In (Artemov 2001), Artemov proved Theorem 5 via syntactic means. This has the added advantage of providing a procedure for converting proofs with cut to those without.

## 6 Conclusion

While our tableau systems are cut-free, they do not have the subformula property. It is unknown at present whether there is a cut-free tableau system for LP (or for S4LP) that does have this property. The troublemaker, of course, is the $\beta^{\psi}$ rule. Since LP is a decidable language, perhaps there is an efficient procedure for computing $\psi$, which would give the system a sort of computable subformula property. This might be a reasonable compromise if we otherwise wish to use the system we presented above.

## Acknowledgements

The author extends his thanks to two anonymous referees, whose comments improved the paper greatly. The author is also grateful for Melvin Fitting's tireless editing of previous versions of this paper. Evan Goris also provided useful feedback and assistance on an earlier version. Additionally, the author thanks Sergei Artemov for his support, advice, and teaching.

## References

Alberucci, L. and G. Jäger (2005). About cut elimination for logics of common knowledge. Annals of Pure and Applied Logic 133, 73-99.
Artemov, S. N. (2001). Explicit provability and constructive semantics. The Bulletin of Symbolic Logic 7(1), 1-36.

Artemov, S. N. (2004). Evidence-based common knowledge. Technical Report TR2004018, CUNY Graduate Center Ph.D. Program in Computer Science.

Artemov, S. N. (2006). Justified common knowledge. Theoretical Computer Science. Article in Press, doi:10.1016/j.tcs.2006.03.009.

Fitting, M. (1999). A simple propostional S5 tableau system. Annals of Pure and Applied Logic 96, 107-115.
Fitting, M. (2003). A semantics for the Logic of Proofs. Technical Report TR-2003012, CUNY Graduate Center Ph.D. Program in Computer Science.
Fitting, M. (2004). Semantics and tableaus for LPS4. Technical Report TR-2004016, CUNY Graduate Center Ph.D. Program in Computer Science.

Fitting, M. (2005). The logic of proofs, semantically. Annals of Pure and Applied Logic 132(1), 1-25.

Mkrtychev, A. (1997). Models for the logic of proofs. In S. I. Adian and A. Nerode (Eds.), Logical Foundations of Computer Science, Proceedings of the 4 th International Symposium, Volume 1234 of Lecture Notes in Computer Science, pp. 266-275. Springer.
Renne, B. (2004). Tableaux for the logic of proofs. Technical Report TR-2004001, CUNY Graduate Center Ph.D. Program in Computer Science.
Renne, B. (2006). Propositional games with explicit strategies. In Proceedings of the 13th Workshop on Logic, Language, and Computation (WoLLIC).

Smullyan, R. M. (1963). A unifying principal in quantification theory. In Proceedings of the National Academy of Sciences of the United States of America (PNAS), Volume 49, pp. 828-832.

## Part III <br> Language \& Computation

# "Drink me": Handling Actions through Planning in a Text Game Adventure 

Luciana Benotti<br>Technical University of Madrid, Spain / Free University of Bolzano, Italy<br>Luciana.Benotti@stud-inf.unibz.it


#### Abstract

The general aim of this work is to investigate how the addition of state-of-the-art reasoning capabilities can be useful in a dialogue system. In particular we will work with FrOz , a dialogue system that was developed at the University of Saarbrücken to explore this idea. FrOz is a text adventure game that uses Description Logics to codify a given game state. It is assisted by a theorem prover (RACER) for inference.

In this paper we discuss how to add a planning step to FrOz 's actions module. The actions module is in charge of executing commands indicated by the player whenever the specified preconditions are satisfied in the current state of the game. The point of adding planning capabilities to the actions module is to increase the flexibility of the dialogue system. To achieve this aim we will use a general purpose planning system, Blackbox, that implements modern planning techniques.


## 1 Planning in Dialogue Systems

Planning has been used to handle different aspects of a dialogue system for several decades now (Allen 1994). For example, plan-based models are suitable for recognizing speech acts performed by the user of a dialogue system, for infering her goals, and for cooperating in their achievement (Perrault and Allen 1980). Planning techniques also have been used in natural language generation: when the system needs to convey large amounts of information using multiple utterances, the organization of the information into utterance-size units is often viewed as a planning process (Reiter and Dale 1997).

The planning assistant TRAINS (Ferguson, Allen, and Miller 1996) is a classical example of the use of planning in a dialogue system. The aim of TRAINS is to help the user to solve routing problems in a transportation domain. In such environment, the human and the computer must work together in a tightly coupled way to solve problems that neither of them could manage alone. System and user collaborate in order to build a suitable plan for solving the problem at hand. Then, planning techniques are used to validate the feasibility of the plan that is being constructed, infer user goals and cooperate in their achievement. However, Ferguson, Allen and Miller report that, according to their experience in TRAINS, and other dialogue systems, traditional planning (finding courses of action from an initial situation to a goal) turns out not to be suitable. To start with, the initial state is usually incompletely specified because of changing conditions or simply because it is too huge to
represent. Similarly, the goals of the plan are also poorly specified. Not only they change over time as the user's concerns and preferences change, but also they typically cannot be extracted and codified in a way suitable for automatic processing. Hence the planning process needs to be tailored and specially implemented for each particular dialogue system.

In our paper we identify a case that can be directly handled by traditional planning, and we show how it is possible to take advantage of a state-of-the-art planning system to solve it. In the task we are going to tackle, initial states, goals and available actions can be completely specified and an off-the-shelf planner can be used to enhance the dialogue system capabilities. The dialogue system we are going to work with is the text game adventure FrOz developed at the University of Saarbrücken (Koller, Debusmann, Gabsdil, and Striegnitz 2004).

In FrOz , all the interaction between player and game is done in natural language. The game can understand commands the player presents as English sentences which verbalize actions that she (the player) wants to execute in the game world (such as "Open the door with the key"). After receiving a command, FrOz verifies if such an action can effectively be executed, checking whether its preconditions hold in the game world. In this case, the game world is updated according to the action effects. Otherwise, the action fails.

It is exactly this behaviour what we want to modify. In certain situations, the player specifies an action assuming that either all preconditions are satisfied, or that they can be easily satisfied by performing trivial additional actions. Let us consider, for example, the situation where the game has just described that a key is lying on a table in front of the player in a room with a locked door. Then the player might input the command, "Unlock the door with the key", which the current version of FrOz will fail to execute because the player is not actually holding the key (the unlock-with(Object Key) action has the precondition instance(Key inventory-object)). That is, the player is obliged to enter a command like "Take the key and unlock the door with it" for the action to succeed. In such cases, a collaborative dialogue system would try to fill the gaps in the input received from the player, trying to guess the missing actions that the player is assuming to be too obvious to specify explicitly. By doing so, the system would free the player from the nuisance of specifying simple extra actions necessary in order to meet the preconditions.

We would like FrOz to be able to compute autonomously the sequence of actions that should be executed in order to get from the world state where the action fails to the state in which this action can be executed (i.e., the state where all the action preconditions hold), and decide if this sequence is 'simple enough' to be executed automatically, even if the player has not specified it in detail. In order to determine which is the proper sequence of actions, a planning step is needed. In this paper we will discuss how a general purpose planner such as Blackbox (Kautz and Selman 1999) can be used to help FrOz do exactly this.

## 2 FrOz

As a dialogue system, FrOz's general architecture follows a standard pipeline (Bernsen, Dybkjaer, and Dybkjaer 1997) composed by six modules. In such architecture, depicted in Figure 1.1, a cycle of input-output in the game can be described briefly as follows. First, the player's input is parsed by the parsing module. This yields a semantic representation specifying the action that the player wants to execute, and also describing the objects that this action involves. Next, the object descriptions are resolved to individuals of the game world by the reference resolution module obtaining a ground term that specifies the action intended by the player. During the third step, the actions module looks up this action in an action database, checks whether its preconditions are met in the world, and, if so, updates the world state with the effects of the action. The changes introduced in the current situation are then reported to the player through natural sounding English text, which is automatically generated by the remaining three modules: content determination, reference generation and realization. FrOz implements state-of-the-art techniques from computational linguistics for each one of these modules.

The game can be instantiated with different scenarios. The functionality offered by the different modules is shared by all scenarios, while each scenario has its own action database and knowledge bases where it codifies its specific characteristics. FrOz already defines some scenarios like the "Space Station", the "Fairy Tale Castle", etc.


Figure 1.1: FrOz original architecture
FrOz uses Description Logic (DL) (Baader, Calvanese, McGuinness, Nardi, and PatelSchneider 2003) knowledge bases to codify the information concerning the state of the
game. A DL knowledge base is a pair $\langle T, A\rangle$ where $T$ is a set of definitions and $A$ a set of assertions. FrOz's knowledge bases are accessed by almost all modules in the pipeline (see Figure 1.1) via queries sent to the RACER reasoner (Haarslev and Möller 2001).

In particular, underlying the system there are two knowledge bases, which share a set of common definitions (the T-Box) and differ only in their set of assertions (the A-Boxes). The common T-box defines the key notions in the world and how they are interrelated. Some of these notions are basic concepts (such as object) or properties (such as alive), directly describing the game world, while others define more abstract notions like the set of all the individuals a player can interact with.

The A-Boxes specify the kind of an individual (for example, an individual can be an apple or a player). Relationships between individuals in the world are also represented here (such as the relationship between an object and its location).

One of the knowledge bases (the world A-Box) represents the true state of the world, while the other (the player A-Box) keeps track of the player's beliefs about the world. The assertions listed in the player A-Box will typically be a strict subset of the assertions in the world A-Box because the player will not have explored the world completely and therefore will not know about all the individuals and their properties. It may happen, however, that some effects of an action are deliberately hidden from the player; for example, if pressing some button in a room has some effect in another room which the player cannot notice. In this case, the player A-Box may actually contain information that is inconsistent with the world A-Box.

These possible inconsistences need to be taken into account when integrating FrOz and the planner Blackbox. We will return to this in Section 4. Now, let us focus in the actions module in FrOz , the module that will interact with the planner.

### 2.1 The Actions Module

As we have already mentioned when we described FrOz's architecture, the actions module receives a ground term that specifies the action intended by the player. For example, if the input is "Take the key", the ground term received by this module will be:

```
take(key1)
```

where key1 is the unique individual in the player A-Box that resulted from the resolution of the determiner "the key".

Given this representation of the action intended by the player, the actions module is responsible for finding the appropriate entry in the actions database, which specifies the action preconditions and effects. This database can be seen as the codification of the 'instructions' that guides the actions module in fulfilling its task. The actions module uses RACER to perform the necessary inferences in order to follow these instructions. The database is specified in a STRIPS-like format (Fikes, Hart, and Nils 1972) and it divides the effects of an action into those that modify the world A-Box (effects) and those that modify the player A-Box (player beliefs) when the action is executed.

An example of an entry in the actions database is given below:

| take (X) |  |
| :---: | :---: |
| preconditions | ```instance(X accessible), instance(X takeable), not(instance(X inventory-object))``` |
| effects | ```add: instance(X inventory-object) delete: related(X individual-filler(X has-location) has-location)``` |
| $\begin{aligned} & \text { player } \\ & \text { beliefs } \end{aligned}$ | ```add: instance(X inventory-object) delete: related(X individual-filler(X has-location) has-location)``` |

The term X in the action representation shown above is a variable that gets bound to the actual argument that the resolution module computed. In our previous example, X would be bound to the constant key1, and thus the preconditions and effects of the operators will become ground terms.

It is important that we grasp how actions are dealt with in FrOz if we want to understand how to integrate planning capabilities in the actions module. Let us see in detail how to read the specification of the action take when applied to our example.

The command "Take the key" issued by the player requires that the key is accessible to the player (instance(key1 accessible)), that it is small enough to be taken (instance (key1 takeable)) and that it is not carried by the player already (not (instance (key1 inventory-object))). When this command is executed, the key becomes an object in the player's inventory (instance( X inventory-object)) and it is no longer located where it used to be. This last effect includes an expression that requests RACER to return the individual in the world that represents the location of the key (individual-filler (key1 has-location)). A RACER expression is embedded in the action specification when the action cannot be specified completely in advance because it depends on the current state of the game (as is the case for most interesting actions). Finally, for this action, the effects on the player beliefs are identical to the effects on the world state.

Remember our example in Section 1 where the player tries to unlock a door with a key that she is not holding. FrOz current actions module will fail to execute the action unlock-with(door1 key1) because its precondition instance(key1 inventory-object) is not satisfied; the action we have just explained, take (key1), needs to be executed first. If we want the actions module to find autonomously the sequence of required actions that should be executed to bridge the gap, we need to enhance FrOz with planning capabilities.

## 3 Blackbox: an Off-the-shelf Planner

Blackbox (Kautz and Selman 1999) is a planning system that works by converting planning problems into Boolean satisfiability problems, and then solving them with a variety of
satisfiability engines. The front-end employs the graphplan system (Blum and Furst 1995) while the back-end role can be played by different satisfiability engines, allowing the use of the engine that is best suited for a particular type of problem.

Blackbox works by fixing the length of the plan in advance and iteratively deepening it. This behaviour makes it particularly well suited for our needs. To begin with, it finds optimal plans (minimal in the number of actions). Optimal plans are crucial because FrOz cannot force the player to do unnecessary actions. Moreover, Blackbox is extremely fast when searching for short plans, and these are exactly the kind of plans that we need in our framework, as it does not seem sensible to allow too much autonomy to the game. Of course, fast responses are critical for a natural interaction with the player.

The input required by Blackbox are STRIPS-style problems specified in the standard Planning Domain Definition Language (PDDL) (Gerevini and Long 2005). A PDDL specification consists of two parts: the domain and the problem. The longest and more complex of these two is the domain specification that contains a crucial element in any planning domain specification: the actions (with its associated parameters, preconditions and effects). On the other hand, the problem specification is relatively simple and contains the initial state (which describes the state of the world at the beginning of the plan), the goal (which represents the desired state of the world after the plan execution) and the objects (that can instantiate the actions) with their corresponding types.

When Blackbox is invoked with a domain, a problem and a maximum plan length it will return an optimal plan of smaller or equal length than the maximum specified, if such a plan exists. Otherwise, it will report that there is no such plan.

In the next section we will discuss how to generate suitable PDDL specifications so that Blackbox is able to find plans in the context of the game.

## 4 Blackbox in FrOz

In our redesigned version of FrOz , when an action fails because some of its preconditions do not hold in the world, the actions module will invoke Blackbox as depicted in Figure 1.2. Blackbox input, conforming the format described in the previous section, will include on one hand the domain specification describing the FrOz scenario that is being played. On the other hand, the problem specification will represent the current and the intended state of the game corresponding to the initial state and goal respectively. Also, we will instruct Blackbox to find plans of up to two actions only. Longer plans are probably not useful. Bear in mind that FrOz is only attempting to 'guess' trivial actions that were left unspecified by the user. Let us describe the domain and problem specification in more detail.

Codifying a Domain As we mentioned, the domain specification is complex, and hence difficult to generate, involving a number of important design decisions. However, as it is independent of the state of the game at a particular moment, it can be generated offline, once and for all for each game scenario. The information required is obtained mainly from


Figure 1.2: FrOz new architecture
the scenario actions database but we also need to query RACER about definitions in the knowledge bases in order to complete the domain specification.

Let us discuss the main intuitions behind the generation of suitable action specifications. As actions in both FrOz and Blackbox are described using a STRIPS-like format, we can expect that the translation is simple. This is true except for two difficulties. Consider the action take we discussed in Section 2.1. It would be encoded in PDDL as follows:

```
(:action take
    :parameters (?x - takeable ?y - top)
    :precondition
        (accessible ?x)
        (no-inventory-object ?x)
        (has-location ?x ?y)
    :effect
        (inventory-object ?x)
        (not(has-location ?x ?y))
)
```

The first obvious difference between the two representations is that action parameters in FrOz are not typed, while Blackbox allows typing. And indeed we want parameters to be typed, because this prevents Blackbox from trying out every action over every single individual in our domain (this would easily lead to a blow up in the plan search space). We can type the parameters in the following way. First, let us call a concept in the game knowledge bases static if it does not appear in any action effect in FrOz 's actions database (in this case, it will clearly never be affected during the game). All other concepts are called dynamic. Now, we say that a parameter ?x belongs to the type t if t is a static concept and instance ( $\mathrm{X} t$ ) is a precondition for the action. In this way we avoid the instantiation of the action take with individuals that do not satisfy this precondition (moreover, the
precondition is no longer necessary and we can eliminate it from the specification of the action). In cases when no such precondition exists then ?x belongs to the type top.

The second evident discrepancy in the proposed translation is that the action has now two parameters instead of only one. The second parameter (?y) represents the individual that the RACER expression individual-filler (X has-location) resolves to. But, how can we be sure that Blackbox will instantiate this parameter with the appropriate individual? We just need to tell the planner that ?x has to be related with ?y by the functional role has-location adding the precondition (has-location ?x ?y) to the action. In general, the embedding of RACER expressions in action specifications is the most complex issue we have to deal with and our approach is to add a parameter and its corresponding precondition for each such expression.

Having explained this, the required encoding can be directly obtained from the representation of the action take in FrOz 's actions database. ${ }^{1}$

Codifying a Problem The problem specification clearly depends on the state of the game and on the input of the player at a particular moment. Hence, this specification should be automatically generated on-line during the execution of the game.

A problem specification consists of three parts. The first one, is the definition of the objects in the problem, with their corresponding types. This can be easily handled by asking RACER which are all the individuals in the knowledge base and their corresponding types. The second part is the initial state, a description of a particular game state where an action fails because some of its preconditions do not hold. RACER can tell us which are the dynamic concepts each object belongs to, and we just have to assert this information in the specification. The last part is the goal of the planning problem. Let us analyse what this goal should be. If we want the player to be able to execute an action she was not able to execute before, we need the preconditions of such an action to hold. Then, the goals of our planning problem will be the preconditions of the action that the player wants to execute. This information can be obtained instantiating the preconditions of such action with the objects that the player is intending to manipulate.

In order to define the first two elements we need to query RACER about the objects and their properties in the knowledge base. However, FrOz has two A-Boxes (the game's and the player's) and we need to decide which one can give us the information we need. It seems natural to choose the player's because we do not want Blackbox to return plans including actions that the player is not aware she can perform. However, we should remember that this knowledge base may contain information that is inconsistent with respect to the current state of the game. So it is possible that Blackbox actually returns a plan that cannot be executed over the game world. But even in this case the plan is useful because it leads the player to find out about her misconception about the world. To clarify this point, let us return to the example introduced in Section 1, where the player tries to unlock a door

[^41]with a key. The key was originally lying on the table, but without the player knowing, it is now in possesion of a cat (who has the surprising ability to appear and disappear at will!). As a consequence, the key is on the table in the player's knowledge base, but in the game world the cat has it. With the added planning capabilities, FrOz would decide to take the key and unlock the door with it. But this sequence of actions will fail (when the preconditions are checked by the actions module) because the key is no longer accessible, and this situation will be informed to the player. If, instead of finding a plan according to the player's beliefs, we would have planned using the world knowledge, FrOz would have automatically taken the key from the cat (for example, by using the steal action) and opened the door for the player, while the player is not even aware where the key actually is. This is clearly inappropriate because we only want FrOz to take actions for the player if we can be sure that these actions agree with the player intentions.

Given the domain and problem specifications and a maximum length for a plan, Blackbox will be able to find the sequence of actions required in order to get from the state where the action fails to the state in which this action can be executed according to the player's beliefs, in case such a sequence exists. If this sequence is simple enough (we will return to this issue in Section 6) then FrOz will execute it and finally it will execute the action input by the player.

Given this setup, Blackbox's performance is impeccable. For the world domains provided with FrOz (around 20 actions schemes and 30 individuals that instantiate around 60 actions), it only takes the planner a couple of milliseconds to find a suitable plan or to answer that there is none. In order to check the scalability of our approach we tested Blackbox with up to 6000 instantiated actions, but even for this huge problem Blackbox still takes less than a second to return a plan or to say that there is none. Blackbox performance does not seem to be a problem in our setup.

To round up our discussion and make things concrete, let us discuss a worked out example in full detail.

## 5 Alice and the Bottle

Suppose we developed a FrOz's scenario based on "Alice in Wonderland," and Alice is now in the rabbit-hole.

The game has just described that there is a little bottle on a table in front of Alice. Around the neck of the bottle there is a paper label, with the words 'DRINK ME' beautifully printed on it in large letters. Then, Alice (the player) might input "Drink the bottle". Without the work described in this paper, FrOz would answer such a demand with a negative response: "You can't do this! You do not have the bottle!". Actually, two preconditions in the action $\operatorname{drink}(X)$ have failed as the player should have the bottle in her inventory (inventory-object bottle1) and it should be uncorked (uncorked bottle1) for the action to succeed.

In the new version of FrOz this would not happen. Instead Blackbox would be invoked and a suitable plan would be found. The specification used during the call would include the "Alice in Wonderland" domain, the current state of the game and the goal. The goal would include the preconditions for the action drink(bottle1):

```
(:goal
    (inventory-object bottle1)
    (uncorked bottle1)
)
```



Given this goal, the plan output by Blackbox would include two actions:

```
Begin plan
```

1 (take bottle1 table1)
2 (uncork bottle1)
End plan

The actions 'take the bottle' and 'uncork the bottle' can be performed, in that order, from Alice's current state in the game. With this information FrOz is able to execute these two actions on its own and respond: "You have the bottle. The bottle is uncorked. You drink the bottle.", a much more friendly and natural answer than FrOz's original reply.

## 6 Discussion

The ideas presented in this paper are still ongoing work, and there are many more interesting research directions to investigate.

To begin with, we will discuss how well our planning task is handled by Blackbox. Blackbox's behaviour can be tailored keeping in mind that FrOz is a computer game. In particular, it is very important that plans returned by Blackbox are both optimal and short. Optimal plans (i.e., plans containing no superfluous actions) are needed because otherwise we risk executing actions the player would not have thought of performing. Moreover, we need short plans because we do not want the planner to solve the game for the player! In this respect, we can easily instruct Blackbox to look only for plans with up to two actions. As we mentioned, given our setup, Blackbox performance is not a issue. However, there is room for improvements. In some cases, our new version of FrOz will not be able to come up with a plan, even if one exists, because the encoding described in Section 4 is incomplete. This is due to the fact that Blackbox input language is less expressive than the language supported by RACER. As arbitrary queries to RACER can be embedded in FrOz actions databases, we cannot expect to be able to cover them in full generality. Nonetheless, the
proposed encoding does improve FrOz behaviour in many common cases. Maybe in the future, planners will be able to handle more complex cases within the time restrictions imposed by an interactive system such as FrOz .

Once the planner outputs a plan, there is one additional issue that should be taken into account. We want the game to perform actions for the player only if the game can assume that the plan is sufficiently 'simple'. How can we guarantee this? One possibility is the following, given a game scenario some actions can be considered minor or trivial, while others cannot. Putting Humpty Dumpty back on the wall is definitely not a trivial action (after all, not even all the king's horses and all the king's men were able to do it!). This classification of actions into essential (which can only be performed explicitly by the player) and minor (which can be executed via planning) should be specified by the scenario designer; and the planning domain would include only those actions that are specified as minor. We could also think of making this distinction dynamic, so that actions which have been already performed at least once by the player can now be considered minor.

The work we discussed has also some interesting consequences for generation. The generation component could be enhanced so that it will render differently the case where the changes it is reporting over the world were explicitly indicated by the player, and the case where they were caused by actions automatically performed by the game. For example, if the player inputs "Take the bottle, uncork it and drink it" FrOz normal reply will be "You have the bottle. The bottle is uncorked. You drink the bottle." However, if the player's input was "Drink the bottle" and the other two actions are inferred by the planner as in our example in Section 5, a more natural (and informative) answer would be "You drink the bottle [taking and uncorking it first]".

## 7 Conclusions

One of the aims in the original FrOz design was to analyse the feasibility of integrating a state-of-the-art reasoning system like RACER, and proving that it could be useful for providing the reasoning capabilities that a dialogue system requires. Our paper discusses the addition of a different kind of inference: planning. And also this time our goal is to use an off-the-shelf system to perform this task. We believe that finding the way to take advantage of current technologies is an important issue if our goal is to design a generic dialogue system.

The integration of a generic planner we have described can be useful for other interesting extensions of the game. For example, offering 'hints' to the player during the game execution, or answering player's questions. Such extensions will be the topic of our future research. But the next issue in our research agenda is evaluation. We should empirically test whether the plans we obtain from Blackbox in our setup are indeed useful in real game situations. This can only be verified by compiling and analysing a corpus of actual interaction with the new version of FrOz .

## Acknowledgements

This work was partially carried out in the context of the research project "Técnicas de Inteligencia Computacional para el diseño e Implementación de Sistemas Multiagentes" (COD 04/E062) of the Universidad del Comahue, Argentina.

## Bibliography

Allen, J. (1994). Natural language understanding. Addison Wesley, 2nd edition.
Baader, F., D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider (2003). The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press.

Bernsen, N., L. Dybkjaer, and L. Dybkjaer (1997). Designing Interactive Speech Systems: From First Ideas to User Testing. Springer-Verlag New York, Inc.

Blum, A. and M. Furst (1995). Fast planning through planning graph analysis. In Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI 95), Montreal, Canada, pp. 1636-1642.

Ferguson, G., J. Allen, and B. Miller (1996). TRAINS-95: Towards a mixed-initiative planning assistant. In Proceedings of the Conference on Artificial Intelligence Planning Systems, Edinburgh, Scotland, pp. 70-77.
Fikes, R., P. Hart, and J. Nils (1972). Learning and executing generalized robot plans. Artificial Intelligence 3, 251-288.
Gerevini, A. and D. Long (2005). Plan constraints and preferences in PDDL3. Technical Report R.T. 2005-08-47, Università degli Studi di Brescia, Italy.
Haarslev, V. and R. Möller (2001). RACER system description. In Proceedings of International Joint Conference on Automated Reasoning (IJCAR 01), Number 2083 in LNAI, Siena, Italy, pp. 701-705. http://www.racer-systems.com/.
Kautz, H. and B. Selman (1999). Unifying SAT-based and graph-based planning. In Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI 99), Stockholm, Sweden, pp. 318-325. http://www.cs.washington.edu/ homes/kautz/satplan/blackbox/index.html.
Koller, A., R. Debusmann, M. Gabsdil, and K. Striegnitz (2004). Put my galakmid coin into the dispenser and kick it: Computational linguistics and theorem proving in a computer game. Journal of Logic, Language and Information 13(2), 187-206. http://www.coli.uni-saarland.de/~koller/projects/froz/.
Perrault, R. and J. Allen (1980). A plan-based analysis of indirect speech acts. Computational Linguistics 6(3-4), 167-182.
Reiter, E. and R. Dale (1997). Building applied natural language generation systems. Journal of Natural Language Engineering 3(1), 57-87.

# Maximum Entropy Tiered Tagging 

Alexandru Ceauşu<br>Research Institute for Artificial Intelligence, Romanian Academy<br>alceausu@racai.ro


#### Abstract

Data sparseness in tagging highly inflectional languages with large tagsets and scarce training resources is a problem that cannot be addressed using only common tagging techniques. Tiered tagging is a two-stage technique that uses for tagging a smaller "hidden" tagset and, in the second phase, recovers the original tagset using a lexicon and a set of hand-written rules. The recovering is possible only for the words contained in the lexicon. The paper describes an experiment that shows how the maximum entropy framework can be used for tiered tagging without a hand-written set of recovery rules and which works also for unknown words.


## 1 Tiered Tagging

The Romanian EAGLES compliant tagset, build within the MULTEXT-EAST initiative (Erjavec 2004), has 614 morpho-syntactic description codes (MSDtags), plus 10 punctuation tags.

Tiered tagging (Tufiş 1999; Tufiş 2000) is a two-stage technique addressing the issue of data-sparseness: (i) intermediary tagging using a reduced tagset (Ctag-set), (ii) replacing the Ctags with contextually appropriate MSD tags (called in (Tufiş 1999) MSD recovery).

The lexicon, underlying the tiered tagging approach, contains the words annotated with the MSD tags, an entry having the form: word lemma msd. For Romanian, this lexicon contains almost 600,000 entries. Based on the MSDtag-set lexicon a Ctag-set lexicon is automatically computed. The algorithm for Ctag-set generation and controlled information loss is described in (Tufiş 2000). This lexicon-based algorithm may produce many different Ctag-sets. The information-loss Ctag-set for Romanian consists of 92 tags, plus 10 punctuation tags. Selecting the most appropriate one was a matter of expertintrospection and required various experimental trials.

To eliminate this inconvenience, (Tufiş and Dragomirescu 2004) describe a language independent algorithm for automatic construction of the "optimal" information lossless Ctag-set. This new algorithm considers the frequency of the words in training corpora, as an additional parameter of the design procedure.

The Ctag-set is derived from the MSDtag-set by repeated generalisations by leaving out some attributes and their respective values from the original tagset specification. This procedure may be information lossless, meaning that the recovering of the left-out information
is deterministic, or may be an information-loss generalisation, meaning that the recovering process would face some ambiguities which have to be solved by using some additional knowledge resource. In (Tufis 1999) this new resource is a set of hand-written contextual disambiguation rules. Both deterministic and the rule-based successful recovering is applicable only to the words recorded in the MSDtag-set lexicon.

The unknown words are likely to appear in any realistic application that requires tagging and they are responsible for the most annotation errors. We replaced the second phase of the tiered tagging process with a maximum entropy-based MSD recovery. In this approach, the rules for Ctag to MSD conversion are automatically learnt from the corpus and their application does not require looking-up the MSD tagset lexicon. Therefore, even the Ctags assigned to unknown words can be converted into MSD tags. If an MSD-lexicon is available, replacing the Ctags for the known words by the appropriate MSD tags is almost $100 \%$ accurate. The estimated accuracy of the Ctag to MSD for unknown words is $95.2 \%$. Moreover, the ME model for Ctag-set - MSDtag-set may disregard the initially assigned Ctag for an unknown word and produce an unrelated MSD tag which better fits in the context. This way, some wrongly tagged unknown words may receive a correct MSD tag.

## 2 Maximum Entropy Framework

The maximum entropy framework is well suited for tagging since it can combine diverse forms of contextual information in a principled manner. Also, maximum entropy is one of the best tagging techniques reporting $96.43 \%$ total word accuracy and $86.23 \%$ unknown word accuracy on unseen Wall St. Journal data (Ratnaparkhi 1998).

Tagging can be re-formulated as a classification problem: the task of the classifier is to extract evidence from a linguistic "context" $b \in B$ and predict a linguistic "class" $a \in A$. The classifier will derive a conditional probability distribution $p$, where $p(a \mid b)$ is the probability of "class" $a$ given the "context" $b$.

The probability model combines the evidence using weights for each predicate of the context:

$$
\begin{align*}
& p(a \mid b)=\frac{1}{Z(b)} \prod_{k}^{j=1} \alpha_{j}^{f_{j}(a, b)}  \tag{1.1}\\
& Z(b)=\sum_{a} \prod_{k}^{j=1} \alpha_{j}^{f_{j}(a, b)} \tag{1.2}
\end{align*}
$$

where $k$ is the number of contextual predicates and $Z(b)$ is a normalization factor. Each contextual predicate $f_{j}$ has a "weight" $\alpha_{j} . p(a \mid b)$ represents the conditional probability of a tag $a$, given the context $b$.

|  | Ctag tagger | MSD tagger | Tagset converter |
| :--- | :---: | :---: | :---: |
| Wordform | x | x |  |
| character length | x | x | x |
| prefix (1-2) | x | x | x |
| suffix (1-4) | x | x | x |
| upper case (all, initial) | x | x | x |
| is abbreviation | x | x | x |
| has underscore | x | x | x |
| has number | x | x | x |
| hyphen position (start, middle, end, <br> none) | x | x | x |
| previous MSD features | x | x |  |
| previous MSD unigram, bigram and <br> trigram | x | x |  |
| previous Ctag unigram and bigram | x | x |  |
| next Ctag unigram and bigram |  | x | x |
| end of sentence punctuation mark | x | x |  |

Table 1.1: Contextual predicates.

A contextual predicate, given $f(a, b)$, may be activated for any word or tag in the context $b$, and must encode the information that help predicting $a$, such as the spelling of the current word, or the preceding unigram, bigram or trigram.

The search algorithm is a top $K$ breadth first search that maintains, for each new word, the $K$ highest probability tag sequence candidates.

When a lexicon is available, the tagger chooses only from the tags available for the respective word. (Ratnaparkhi 1998) reports minimal increases in performance when his maxent tagger for English uses a lexicon ( $0.12 \%$ ). We observed that this is not the case with Romanian - the tagger accuracy is increased by $1.81 \%$.

## 3 Tagging and Tagset Conversion

We developed three types of maximum entropy classifiers: (i) Ctag-tagger (Ctag-set - 102 descriptors); (ii) MSD-tagger (MSDtag-set - 624 descriptors); (iii) tagset converter (Ctag to MSD). They are based on SharpEntropy (Northedge 2005), a C\# port of the MaxEnt toolkit (http://opennlp.sourceforge.net).

The set of contextual predicates used by each of them is detailed in table 1.1.

### 3.1 Ctag-tagger

The Ctag-tagger has basically the same architecture as the one from the OpenNlp Maxent package. Only the context generator of the tagger was modified in order to accommodate

| Wordform | Ctag | MSD |
| :--- | :--- | :--- |
| holul | NSRY | Ncmsry |
| blocului | NSOY | Ncmsoy |
| mirosea | V3 | Vmii3s |
| a | S | Spsa |
| varză | NSRN | Ncfsrn |
| călită | ASN | Afpfsrn |
| şi | CR | Crssp |
| a | TS | Spsa |
| preşuri | NPN | Ncfp-n |
| vechi | APN | Afp-p-n |
| - | PERIOD | PERIOD |

Table 1.2: Sample data.
features that we considered important for Romanian (like the position of the hyphen, the numbers of the characters for suffix and prefix analysis, end of sentence punctuation mark, etc.).

The tagger uses the Ctag-set (around 100 tags). In this familiar tagging scenarion, the Ctag-tagger outperforms an HMM tagger with $1.5 \%$ in accuracy when tagging the "1984" Romanian corpus.

### 3.2 MSD-tagger

To demonstrate the need of an intermediary tagging with a reduced tagset when tagging a highly inflectional language as Romanian is, we developed a tagger that uses the MSDtagset (624 descriptors).

The MSD-tagger has basicaly the same context generator as the Ctag-tagger. To improve its accuracy the MSD-tagger also uses as contextual predicates the feature description encoded in the MSD labels. For example, the features generated for the morpho-syntactical descriptor "Ncmsry" will be "N0." (PoS=noun), "N1.c" (Type=common), "N2.m" (Gender=masculine), "N3.s" (Num-ber=singular), "N4.r" (Case=direct) and "N5.y" (Definiteness=yes).

The MSD-tagger accuracy outperforms a HMM MSD tagger with more than $3 \%$.

### 3.3 Tagset Converter

The tagset converter maps the C-tags to MSD-tags. The classifier of the tagset converter makes use of both Ctag and MSDtag contextual predicates having thus more information than the Ctag-tagger and MSD-tagger.

From the training data the tagset converter learns a partial conversion lexicon (similar to word-form lexicon) the entries of which have the form: word $m s d T a g_{1} \cdots m_{1} d T a g_{n}$.

| Name | Values |
| :---: | :---: |
| Wordform | "călită" |
| character length | 6 |
| prefix (1-2) | "c", "că" |
| suffix (1-4) | "ă", "tă", "ită", "lită" |
| upper case (all, initial) | false |
| is abbreviation | false |
| has underscore | false |
| has number | false |
| hyphen position (start, middle, end, none) | false |
| previous MSD features | "PoS=noun", "Type=common", "Gender=feminine", ber=singular", "Case= ${ }^{\text {Num- }}$ direct", "Defi- niteness=no" |
| previous MSD unigram, bigram and trigram | "Ncfsrn", "Ncfsrn,Spsa", "Ncf- srn,Spsa,Vmii3s" |
| previous Ctag unigram and bigram | "NSRN", "NSRN,S" |
| next Ctag unigram and bigram | "CR", "CR,TS" |
| end of sentence punctuation mark | "." |

Table 1.3: Contextual predicates for the word "călită".

| Unknown word accuracy without word-form lexicon | $95.20 \%$ |
| :--- | :--- |
| Total word accuracy without word-form lexicon | $98.66 \%$ |
| Total word accuracy with word-form lexicon | $99.04 \%$ |

Table 1.4: Tagset converter accuracy on the "1984" corpus.

It also uses an a-priori non-lexicalised resource containing the complete correspondences between Ctagset and MSD tagset of the form: Ctag $m s d T a g_{1} \cdots m s d T a g_{n}$. If the mapping between the tagsets is not available, it is learned from the corpus. This additional resource allows the tagset converter to generate, with high accuracy, MSD tags even for unknown or partially known words (i.e. either missing from the learnt lexicon or learnt with an incomplete ambiguity class).

In the tables 1.2 and 1.3 is an example of how the conxtual predicates are selected by the context generator of the tagset converter.

The tagset converter has an accuracy of $99.04 \%$ (1.4) when an additional lexicon is available. This performance cannot be compared to the rule-based conversion approach because in the corpus we tested there are also unknown words.

### 3.4 Tiered Tagger

The tiered tagger is the combination of the Ctag-tagger and the MSD-tagger.

| Tagging method | Ctag- <br> tagger | MSD- <br> tagger | Tiered <br> tagging |
| :--- | :--- | :--- | :--- |
| Unknown word accuracy without <br> word-form lexicon | $82.24 \%$ | $78.65 \%$ | $78.76 \%$ |
| Total word accuracy without <br> word-form lexicon | $96.81 \%$ | $96.22 \%$ | $96.56 \%$ |
| Total word accuracy with word- <br> form lexicon | $98.62 \%$ | $98.45 \%$ | $98.58 \%$ |

Table 1.5: Accuracy on the "1984" corpus.

Because it uses more contextual information than an usual tagger (it runs on the already tagged corpus), when employed by the tiered tagger, the tagset converter can also correct tagging errors on unknown words. If a word is not in the wordform-MSD lexicon the MSD tag that the model predicts may not be among the Ctag to MSD mapping alternatives. In this case, the MSD tag the model predicted is taken into account in the $K$ breadth first search.

## 4 Evaluation

For our experiments we used the CONCEDE edition (Erjavec 2004) of the parallel corpus "1984" (118025 words). We kept out $1 / 10$ of the corpus for evaluation.

In table 1.5 are presented the evaluation results. The Ctag-tagger and MSD-tagger columns display the accuracy of the ME taggers trained with the respective tagsets annotated corpora. The tagset converter column shows the accuracy of the tagset converter. The Tiered tagging column shows the accuracy of the combination between Ctag-tagger and the tagset converter.

We were especially interested in evaluating the tagging accuracy of the unknown or partially known words, and accuracy of Ctag-MSD conversion for these words. The table 1.5 shows that the tagging accuracy is significantly better when our large word-form lexicon is used, but also it shows that the C-tag to MSD conversion is reliable even without this additional resource.

The accuracy of the tiered tagging approach is better than the one of the direct MSD tagging. The difference is higher when the domain of the evaluation corpus is different from the corpus used for training (as observed in (Tufis 1999)) . Our maximum entropy tiered tagging application can reliably handle unknown words ( $78.76 \%$ ). At a closer inspection of the conversion "errors" we noticed that several generated MSD tags which were different from the ones in the gold standard contained more information than a lexicon can provide. The most frequent case was the specification of the gender or case attributes for invariable or unmarked adjectives. This contextually deduced information appeared as result of learning an agreement rule in Romanian: the noun and its modifier must agree in gender number and case.

## Bibliography

Tufiss, Dan and Liviu Dragomirescu (2004). Tiered Tagging Revisited. In Proc. of the Fourth Intl. Conf. on Language Resources and Evaluation, LREC'2004, ELRA, Paris, pp. 39-42.

Tufis, Dan (2000). Using a Large Set of EAGLES-compliant Morpho-Syntactic Descriptors as a Tagset for Probabilistic Tagging. International Conference on Language Resources and Evaluation LREC'2000, Athens, pp. 1105-1112
Tufiş, Dan (1999). Tiered Tagging and Combined Classifiers. In F. Jelinek, E. Nth (eds) Text, Speech and Dialogue, Lecture Notes in Artificial Intelligence, Springer, pp. 28-33

Erjavec, Tomă̌ (2004). MULTEXT-East Version 3: Multilingual Morphosyntactic Specifications, Lexicons and Corpora. In Proc. of the Fourth Intl. Conf. on Language Resources and Evaluation, LREC'2004, ELRA, Paris

Erjavec, Tomaz̆ (2001). Harmonised Morphosyntactic Tagging for Seven Languages and Orwell's 1984. In Proceedings of the 6th Natural Language Processing Pacific Rim Symposium, Tokyo, pp. 487-492
Ratnaparkhi, Adwait (1998). Maximum Entropy Models for Natural Language Ambiguity Resolution. Ph.D. thesis, University of Pennsylvania, Philadelphia
Northedge, Richard (2005). Maximum Entropy Modeling Using SharpEntropy. http://www.codeproject.com/csharp/sharpentropy.asp

Northedge, Richard (2005). OpenNLP Maxent machine learning package. http://opennlp.sourceforge.net

# Automatically Extending the Lexicon for Parsing 

Tim van de Cruys<br>Alfa-informatica<br>University of Groningen<br>t.van.de.cruys@rug.nl


#### Abstract

This paper describes a method for automatically extending the lexicon of wide-coverage parsers. The method is an extension to the automatic detection of coverage problems of natural language parsers, based on large amounts of raw text (van Noord 2004). The goal is to extend grammar coverage, focusing in particular on the acquisition of lexical information for missing and incomplete lexicon entries (including subcategorization frames). In order to assign lexical entries for unknown words, or for words for which the lexicon only contains a subset of its possible lexical categories, we propose to apply a parser to a set of unannotated sentences containing the unknown word, or to a set of unannotated sentences (found by error mining) in which the word apparently was used with a missing lexical category. The parser will assign all universal lexical categories to the problematic word. Once the parser has found a result for the sentence, it can output the lexical category that was eventually used in its best parse. If this process is repeated for a large enough sample of sentences, it is expected that either a single or a small number of lexical categories can then be identified which are to be taken as the correct lexical categories of this word. A maximum entropy classifier is trained to select the correct lexical categories.


## 1 Introduction

Deep grammar parsing techniques have improved tremendously in the last few years. The emergence of adequate grammar descriptions and efficient parsing techniques have made it theoretically feasible to parse instances of raw text. The consequences of Moore's law (computational power doubles every 24 months) ensure that this is also practically feasible.

The main factor that needs to be improved is the coverage, also called the robustness of the parser. Hand-crafted linguistic descriptions such as wide-coverage grammars remain despite the tremendous improvements - still quite incomplete. The hand-crafted lexicon is often the problem child. As previous work has noted (Baldwin, Bender, Flickinger, et al. 2004), most coverage problems are due to missing or incomplete lexicon entries. In this paper, a method is described that is able to automatically detect and correct those missing or incomplete entries. The method builds on the error mining technique described in van Noord (2004), that is able to automatically discover systematic mistakes in a parser by using very large but unannotated corpora. The technique is summarized in section 3.1.

In the experiments described here, the Alpino wide-coverage parser for Dutch is used (Bouma, van Noord, and Malouf 2001; van der Beek, Bouma, and van Noord 2002). This
parser is based on a large ( $\pm 600$ grammar rules) constructionalist Head-driven Phrase Structure Grammar (HPSG) for Dutch, as well as a very large ( $>100 \mathrm{~K}$ words) lexicon. The parser is robust in the sense that it essentially always produces a parse. If a full parse is not possible for a given sentence, then the parser returns a (minimal) number of parsed nonoverlapping sentence parts.

## 2 Previous Work

### 2.1 Unification-based Grammar Induction

In unification-based frameworks such as HPSG, information in the lexicon may be underspecified, to become more specific only when it is actually used in a parse tree. An example is an English verb form like drink, which is only specified as 'present tense' (except for the third person singular). It is only when the form combines with a subject (such as the pronoun $I$ ) that it can be fully specified ('present tense first person singular'). This property of unification-based frameworks can be used to induce the grammatical features of unknown words. The feature structure of an unknown word is incrementally updated, as more and more sentences with different occurrences of the unknown word form are read by the unification algorithm. This kind of technique is elaborated in Erbach (1990) and Barg and Walther (1998). Fouvry (2003) applies the technique to a large-coverage grammar for German.

A problem with this approach is that the feature structure will be partly too general and partly too specific. Barg and Walther (1998) discuss in this view the concepts of generalisable and revisable information. The former are values that are too specific, while the latter are values that may be overwritten. The algorithm as such also is not able to cope with errors in text: it doesn't take any statistics into account.

### 2.2 Statistical Grammar Induction

Next to the more rigid framework of inducing feature structures, quite some authors have taken a statistical approach in lexicon induction. One of the first efforts to incorporate statistics in order to induce lexicon entries (and more specifically subcategorization frames) has been made by Brent (1993). His work makes use of statistics (more specifically hypothesis testing on binomial frequency data) to prevent noise from blurring the results. However, his work does not rely on full parsing of sentences, but on certain 'morpho-syntactic cues', to discover verbs and their arguments.

Schulte im Walde (2002) also pursued a statistical approach in inducing subcategorization frames, by making use of a probabilistic context free grammar (PCFG). The context free grammar yields for each verb a frequency distribution of subcategorization frames; applying a cut-off yields the subcategorization frames found by the algorithm.

The approach described in this paper differs from the previous approaches in two ways:

- the use of a broad-coverage deep grammar parser in order to find the possible lexical types of an unknown word;
- the use of a maximum entropy classifier to classify the output of the parsing method, in order to find the actual lexical types.


## 3 Methodology

### 3.1 Error Mining

The error mining technique assumes that a large corpus of sentences is available. Each sentence is a sequence of tokens (including words as well as punctuation marks, etc.). The parser is run on all sentences, and we note for which sentences the parser is successful ${ }^{1}$. The parsability of a word is defined as the ratio of the number of times the word occurs in a sentence with a successful parse and the total number of sentences that this word occurs in. Thus, if a word only occurs in sentences that cannot be parsed successfully, the parsability of that word is 0 . On the other hand, if a word only occurs in sentences with a successful parse, its parsability is 1 . If there is no reason to believe that a word is particularly easy or difficult, then we expect its parsability to be equal to the coverage of the parser (the proportion of sentences with a successful parse). If its parsability is (much) lower, then this indicates that something is wrong. Normally, the coverage of the parser lies between $91 \%$ and $95 \%$. Yet, for many words, parsability values were found much lower than that, including quite a number of words with parsability 0 .

Words or word sequences that are found by this technique are considered problematic. This problem might be due to missing grammatical constructions, or due to missing or incomplete lexical entries. This paper focuses on the latter.

### 3.2 Automatic Lexical Acquisition Algorithm

## Parsing with Universal Tagset

Words that have been found problematic by the error mining technique may then be fed to the lexical acquisition algorithm. For each problematic word, a large number of sentences (in our experiments, we used 100) containing the word is extracted from large corpora, or taken from the Internet. Those sentences are parsed with a different version of Alpino: a parsing method has been used in which all possible 'universal tags' are assigned to the unknown or problematic word. By universal tags, we mean all tags that belong to an open part of speech class. ${ }^{2}$ Infrequent tags and function word tags are not taken into consideration. This boils down to a universal tagset of 340 tags. Note that 'tag' in this sense does not indicate the part of speech tag, but the atomic lexical type used by

[^42]a parser. These lexical types include grammatical information such as subcategorization frames. ${ }^{3}$ The parser assigns all universal tags to the unknown word, and each parse that is formed this way receives a probability score by the parser's disambiguation component. ${ }^{4}$

During the parsing process, Alpino's POS-tagger (Prins and van Noord 2001) keeps filtering implausible tag combinations. For example, if a determiner appears in front of the unknown word, the extensive range of verb tags should not be taken into consideration. The POS-tagger makes sure these tags will be filtered out. This process heavily reduces the computational overload, and makes the parsing method computationally feasible.

The parse that is considered the best parse by the disambiguation model is preserved. The tag that has been assigned in the best parse is our best guess given the actual sentence. When all sentences have been parsed, a list can be drawn up with the tags that have been used successfully, and their frequency. The correct tag(s) can then be determined statistically.

Take the following example sentences, in which the unknown word sneup appears (tags have been somewhat simplified to improve readability):
(1) Ik heb zin in sneup.

I feel like sneup
I would like some sneup.
(2) De sneup ligt in de kast.

The sneup lies in the cupboard
The sneup is in the cupboard.
(3) Ik wil sneup!

I want sneup
I want sneup!

- ik/pronoun(1st,sg,nom) heb/verb(hebben,sg1,transitive) zin/noun(de,sg) in/preposition(in) sneup/noun(de,sg)
- de/determiner(de) sneup/noun(de,sg) ligt/verb('hebben/zijn',sg3,ld_pp)
in/preposition(in) de/determiner(de) kast/noun(de,sg)
- ik/pronoun(1st,sg,nom) wil/verb(hebben,modal,intransitive)


## sneup/adverb

In these examples, the parser assigns twice the tag 'noun(de,sg)' to the word sneup and once the tag 'adverb'. This makes the first tag the most probable.

[^43]
## Adapting the Disambiguation Component

For the parsing method to work properly, the statistical disambiguation model of the parser had to be adapted. The parser's disambiguation model heavily relies on the lexicon. Based on training data, the disambiguation model has, for example, a preference to parse prepositional phrases as a prepositional complement to the verb, if such subcategorization frame exists. But this doesn't make any sense when parsing with a universal tagset, as every prepositional phrase would get classified as a complement to the verb. This problem is overcome by weighting each universal tag by its a priori probability (the actual frequency with which the tag appears in the training data). This way, the probability that a subcategorization frame with a prepositional complement is selected for an unknown verb, is scaled to the overall probability that such a subcategorization frame actually appears. In general, when the tag of a certain word in the sentence is unknown, we want the disambiguation component to take into account the a priori probability of all tags in the tagset when assigning a certain tag to the unknown word.

## Word Classification

A maximum entropy (maxent) classifier (Le 2004) has been trained to classify the unknown words, taking as features the outcomes of the procedure described (i.e. the tags that were successful). To enable the maxent classifier to make broader generalizations, each subattribute of the tags (e.g. singular/plural difference with nouns, adjective attributes, subcategorization frame of verbs, ...) has also been handed to the classifier separately.

Next to the information yielded by our parsing method, the morphology of the unknown word is taken into account. The morphology features that are used are:

- the word ending (three last letters, two last letters, last letter)
- +/- past participle, e.g. gefietst
- +/- word starts with particle, e.g. rondfietsen

A finite state automaton has been designed to determine whether a word has the characteristics of a past participle.

## Training the Classifier

To be able to work with decent training data, Alpino has been 'untrained' for a small part of the lexicon (about 1500 words, of which 1000 words have been used for training). Those words were considered unknown by Alpino, yielding a list of possible tags assigned by the procedure described above. The correct tags, to be used for training, could be extracted from the original lexicon.

As in many natural language processing applications, ambiguity is an important problem. This is no different when trying to deduce a word's possible tags. Part of this ambiguity can be handled by the algorithm, as some words are consistently ambiguous
(i.e. the kind of ambiguity that would be described by lexical rules). An example of such ambiguity is to be found in the Dutch verb system: the infinitive verb form in Dutch will also always be the plural form of the verb. This kind of systematic ambiguity is handled by the algorithm: if a structurally ambiguous tag has been found by the algorithm, the other one is automatically added by a number of manually constructed rules.

But the majority of ambiguity is not that straightforward. Such ambiguity is a potential problem for the algorithm. Consider the following sentences:
(4) 's Zomers fiets ik graag.
in summer bike.V I like
In summer, I like to bike.
(5) Ik rijd graag rond op mijn fiets. I ride like around on my bike.N I like to ride around on my bike.

Sentences 4 and 5 are an example of the latter kind of ambiguity: in 4 , fiets is used as the first person singular of the verb fietsen. In 5 , fiets is a noun. Note that there is quite some ambiguity of this kind, as it does not only depend on homonymous words, but also on internal ambiguity such as different subcategorization frames. In training, this kind of ambiguity is tackled in the following way: if a certain word had more than one lexical type in the lexicon, the classifier has been trained with all the lexical types available for the same attributes. In other words, if a word is ambiguous (as stipulated by the lexicon), both tags are taken into account for each context.

## 4 Results \& Evaluation

### 4.1 Results

The classifier yields a probability score for each tag. Only the best tags are kept, which are the tags with a probability higher than or equal to $6.5 \%$. The best tags are also determined relatively. If the probability score of a certain tag divided by the score of the next best tag on the list is higher than 8 , the next tags are not taken into consideration. These values are not chosen randomly: they yield the highest f-score for the development data.

To evaluate the classifier, the same procedure has been used as for training: words have been parsed with a version of Alpino that has been untrained for these words, and the results have been compared with the tags that are available in the original lexicon. As one word might have several lexical types, we have evaluated the results in terms of precision and recall. Precision indicates how many of the lexical types that have been found by our algorithm are correct. Recall indicates how many of the lexical types of a certain word are actually found. The results given are the average precision and recall for the $\pm 500$ test words. ${ }^{5}$

[^44]Two kinds of baseline have been used:

- a naive baseline in which every unknown word is assigned the overall most frequent tag, namely [noun(de,sg)];
- a more elaborate baseline, in which the most frequent tag for the part of speech of the unknown word is assigned. ${ }^{6}$ The most frequent tag for each part of speech is shown in table 1.1. These POS tags have been deduced from Alpino's lexicon.

Past participles (psp) have been considered as a separate word class, as they show a lot of structural ambiguity in Dutch: past participles can systematically be used as verb forms as well as adjectives. By taking them as a separate class, the performance of the classifier on this specific task can be evaluated.

As the rest category contains only 13 examples, we will not pay any attention to it in the evaluation. To properly evaluate the other part of speech classes (such as Dutch adverbs), more training and test data is needed.

| POS | n | most frequent tag |
| :--- | :---: | ---: |
| noun | 226 | noun(de,sg) |
| adjective | 101 | adjective(e) |
| past participle (psp) | 53 | adjective(no_e(adv)) |
| verb | 85 | verb(hebben,pl,transitive) |
| rest | 13 | tmp_noun(tmp_de,sg) |

Table 1.1: Most frequent tag for each part of speech
Table 1.2 shows the overall results of our algorithm (morphology \& parse results), compared to the baselines, and compared to the use of a maxent classifier with only morphological information and a maxent classifier with only the information yielded by our parsing method. In tables 1.3-1.5, these results are split out for each part of speech.

Table 1.2 shows that our algorithm is able to reach a precision of $77.55 \%$ and a recall of $72.15 \%$, yielding an f -score of $74.75 \%$. The algorithm beats the naive baseline by $50 \%$, and the more sophisticated baseline by about $35 \%$. Our parsing method without the combination with morphology achieves already quite good results (yielding an f-measure of $\pm 71 \%$ ). Combining this method with morphological information further improves these results, although recall decreases slightly.

Table 1.3 shows the results for the maxent classifier only trained on the morphological information of the word.

[^45]|  | precision <br> $(\%)$ | recall <br> $(\%)$ | f-measure <br> $(\%)$ |
| :--- | :---: | :---: | :---: |
| naive baseline | 23.01 | 20.80 | 21.85 |
| POS-based baseline | 44.14 | 35.22 | 39.18 |
| morphology | 53.93 | 59.73 | 56.68 |
| parse results | 69.79 | 72.47 | 71.11 |
| morphology \& parse results | 77.55 | 72.15 | 74.75 |

Table 1.2: Overall results

| Morphology | precision <br> $(\%)$ | recall <br> $(\%)$ | f-measure <br> $(\%)$ |
| :--- | :---: | :---: | :---: |
| noun | 67.60 | 67.69 | 67.64 |
| adjective | 53.94 | 68.07 | 60.19 |
| psp | 44.11 | 56.94 | 49.71 |
| verb | 28.10 | 35.98 | 31.56 |
| rest | 25.00 | 23.08 | 24.00 |

Table 1.3: Evaluation of morphological information results

The results of table 1.3 seem to indicate that morphology is already quite a good indicator for noun and adjectives, but more complex syntactic features (such as the subcategorization frame of verbs) are evidently not found. In order to find these features, we need the results yielded by our universal parsing method. Those results are given in table 1.4.

| Parse results | precision <br> $(\%)$ | recall <br> $(\%)$ | f-measure <br> $(\%)$ |
| :--- | :---: | :---: | :---: |
| noun | 88.53 | 84.56 | 86.50 |
| adjective | 66.37 | 82.43 | 73.53 |
| psp | 47.71 | 53.27 | 50.34 |
| verb | 41.96 | 43.33 | 42.63 |
| rest | 42.56 | 53.85 | 47.54 |

Table 1.4: Evaluation of parsing with universal tags

Table 1.4 shows that our parsing method scores better than the morphology classifier, with nouns reaching up to $86.5 \%$ and adjectives reaching up to $73.5 \%$. This is an increase of respectively about $20 \%$ and $10 \%$ compared to morphology. Verbs, on the other hand, still have unsatisfactory results: the results of the past participle have not increased, and also the other verbs have low results ( $42.5 \% \mathrm{f}$-measure). The information conveyed by morphology is an important feature for verbs. In table 1.5, the results of the combination
of both our parsing method and morphology are shown.

| Morphology + <br> parse results | precision <br> $(\%)$ | recall <br> $(\%)$ | f-measure <br> $(\%)$ |
| :--- | :---: | :---: | :---: |
| noun | 88.42 | 83.73 | 86.01 |
| adjective | 74.42 | 78.71 | 76.50 |
| psp | 71.98 | 54.37 | 61.95 |
| verb | 61.80 | 51.57 | 56.22 |
| rest | 38.46 | 26.92 | 31.67 |

Table 1.5: Evaluation of the combination of the universal parsing method with morphology
Combining the parsing method with morphology boosts the results for past participles and verbs with more than $10 \%$ ( f -measure $\pm 62 \%$ and $\pm 56 \%$ ). Morphology is indeed beneficiary for the verb results. Also, the adjective results increase slightly. The noun results, on the other hand, slightly decrease when combining with morphology, although this decrease does not outweigh the advantages for verb classification.

Common errors are found in the acquisition of adjective tags. The Alpino grammar contains a rather complicated adjective system. Different distinctions exist for adjectives that can be used predicatively, attributively, ... The algorithm is not always able to capture the correct subfeatures. Still, adjectives reach both precision and recall of about $\pm 75 \%$.

Also, the acquisition of infrequent subcategorization frames, such as:

- ditransitive verbs
- verbs with prepositional complement
- verbs with locative complement
- verbs with sbar-complement
poses some problems. To classify verbs with these subcategorization frames properly, we probably need more data, and perhaps a small adaptation of our parsing method (see below).


## 5 Conclusion and Future Work

The evaluation of our lexical acquisition algorithm gives quite good results. The algorithm beats the naive baseline by $50 \%$, and the POS-based baseline by $35 \% .52 \%$ of the words get all (and only) the correct lexical type(s) (as specified in Alpino's lexicon). This approach shows that automatic lexical acquisition is certainly feasible. There is, however, still room for improvement, especially with regard to verbs.

Which brings us to some further research issues. First of all, it might be interesting to investigate the usefulness of a cascading classifier, i.e. a separate classifier for each
part of speech. The results with regard to morphology (verb classification improves but noun classification gets worse) seem to indicate that this should indeed improve the overall result. Next, it might be useful to generalize over word paradigms when classifying words. Especially with regard to verbs, this generalization might improve the classification of subfeatures, such as the subcategorization frame of verbs.

It also remains to be investigated how to cope with unknown multi-word expressions. Coping with such expressions is likely to be a more difficult task, as the parsing method we use heavily relies on the context of words; if one of the context words is also an unknown word (as is the case with multi-word expressions), the method might have difficulties finding appropriate tags.

Next, it might be better to make use of the top-n parses (e.g. the 5 best parses) that are yielded by our parsing method. At the moment, only the best one is used, while the next best parses might contain quite useful information. This information should also be taken into account. This adaptation might give better results for infrequent subcategorization frames, as those frames might not make to the top-n parse due to their low a priori probability.

One last, important issue is to test our classification algorithm for unknown words on a real test set of unknown words. This will be done by testing Alpino's performance on a hand annotated test set that is not part of Alpino's treebank. Next, the lexical acquisition algorithm is applied to unknown words, the acquired tags are added to Alpino's lexicon, and the test set is parsed again, to see whether the results have improved. This way, it will become clear to what extent our classification algorithm might improve Alpino's coverage and accuracy.

## Bibliography

Baldwin, T., E. M. Bender, D. Flickinger, et al. (2004). Road-testing the English Resource Grammar over the British National Corpus. In Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004), Lisbon, Portugal.
Barg, P. and M. Walther (1998). Processing Unknown Words in HPSG. In Proceedings of the 17th International Conference on Computational Linguistics, Volume 1, Montreal, Canada, pp. 91-95.
Bouma, G., G. van Noord, and R. Malouf (2001). Wide coverage computational analysis of Dutch. In W. Daelemans, K. Sima'an, J. Veenstra, and J. Zavrel (Eds.), Computational Linguistics in the Netherlands 2000.

Brent, M. R. (1993). From grammar to lexicon: unsupervised learning of lexical syntax. Comput. Linguist. 19(2), 243-262.

Erbach, G. (1990). Syntactic processing of unknown words. In P. Jorrand and V. Sgurev (Eds.), Artificial Intelligence IV - methodology, systems,applications, pp. 371-382. Amsterdam: North-Holland.
Fouvry, F. (2003, April 12-17). Lexicon acquisition with a large-coverage unificationbased grammar. In EACL 2003. $10^{\text {th }}$ Conference of The European Chapter. Conference Companion, Budapest, Hungary, pp. 87-90. ACL.
Le, Z. (2004). Maximum Entropy Modeling Toolkit for Python and $C++$. http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html.
Prins, R. and G. van Noord (2001). Unsupervised POS-tagging Improves Parsing Accuracy and Parsing Efficiency. In Proceedings of the Seventh International Workshop on Parsing Technologies (IWPT 2001), Beijing, China, pp. 154-165.

Schulte im Walde, S. (2002). Evaluating Verb Subcategorisation Frames learned by a German Statistical Grammar against Manual Definitions in the Duden Dictionary. In Proceedings of the 10th EURALEX International Congress, Copenhagen, Denmark, pp. 187-197.
van der Beek, L., G. Bouma, and G. van Noord (2002). Een brede computationele grammatica voor het Nederlands. Nederlandse Taalkunde 7(4), 353-374. in Dutch.
van Noord, G. (2004). Error Mining for Wide-Coverage Grammar Engineering. In Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Barcelona, Spain, pp. 446-453.
van Noord, G. (2006). At Last Parsing Is Now Operational. In P. Mertens, C. Fairon, A. Dister, and P. Watrin (Eds.), TALN06. Verbum Ex Machina. Actes de la $13 e$ conference sur le traitement automatique des langues naturelles, Leuven, pp. 20-42.

# Bootstrapping Multilingual Geographical Gazetteers from Corpora 

Marieke van Erp<br>ILK/Language and Information Science, Tilburg University<br>M.G.J.vanErp@uvt.nl


#### Abstract

In this paper an approach to automatically generating multilingual geographical name gazetteers via two bootstrapping loops on different corpora is presented. First, a small seed-list of geographical names is matched to an unannotated dataset in one language, and training data for a memory-based classifier is generated. Memory-based learning is applied to extend the gazetteer. Then a cross-over to a different language is made by matching this extended gazetteer to a corpus in a different language. Again, training data for a classifier is generated and the bootstrapping process is repeated in order to extend the gazetteer further. This process is quite similar to co-training, in which information from other sources is introduced to enhance classification. To estimate the difference between the initial seed-list and the final gazetteer and thereby to evaluate the performance of the algorithm, they were matched to three datasets with manually annotated geographical entities.


## 1 Introduction

In corpus-based natural language processing one hardly ever finds clean data. A problem with real data can be that it is made up of data in several languages; this occurs for instance on webpages. Multilingual data may pose problems to natural language processing tools if one wants to extract information from these sources. One of the first tasks in information extraction is Named Entity Recognition (NER), the branch of information extraction that is concerned with identifying and classifying expressions that refer to named entities, such as people, organisations and companies, and geographical locations. It discerns the who, where, and what elements in a text, which can be seen as one of the first step towards further natural language processing tasks such as text summarisation, question answering and ultimately machine translation. At the seventh Message Understanding Conference (MUC) in 1998 a special track was devoted to NER. Since then much work has been done in the field of NER, demonstrating a wide variety of approaches. For comparisons of NER systems several competitions at conferences and workshops such as LREC ${ }^{1}$ and CoNLL (Tjong Kim Sang 2002; Tjong Kim Sang and De Meulder 2003) have been organised.

One approach that is fairly simple and seems to be particularly well-suited for monolingual geographical NER is the use of gazetteers (Mikheev et al. 1999). One can start

[^46]off with a small seed-list of geographical names and extend that by applying machine learning techniques to increase recall. Bootstrapping gazetteers is fairly common practice nowadays, see for instance: Jones et al. (1999), Niu et al. (2003), Pratim Talukdar et al. (2006), Riloff and Jones (1999), and Uryupina (2003). However, NER systems are generally language-dependent and thus not suitable for the abundance of multilingual data nowadays found on, for instance, the World Wide Web. The ability to deal with multilingualism is needed for tasks such as cross-lingual information extraction (Riloff et al. 2002) or for Natural Language Processing (NLP) tasks on multilingual textual databases. ${ }^{2}$ In this paper a language-independent approach to the problem of recognising geographical entities is proposed. To the author's knowledge, bootstrapping gazetteers has not been done cross-lingually in order to create a gazetteer that can deal with multilingual text. The aim of this work is to investigate the possibility of inducing a multilingual gazetteer by a bootstrapping process from unannotated data in English, Dutch and German, starting with a small English seed-list. Apart from providing a possible approach to dealing with named entities from multilingual sources, bootstrapping from multilingual source data may be beneficial for monolingual NER as well, as a corpus in a different language may contain useful information that is not present in the corpus in the first language. This assumption has been found useful for various NLP tasks such as automatic verb classification (Merlo et al. 2002), machine translation (Callison-Burch and Osborne 2003) and word sense disambiguation (Diab and Resnik 2002). Using different languages to aid classification can also be compared to co-training, as proposed by Blum and Mitchell (1998), who classified webpages using the content and the hyperlinks pointing to that page as different input feature spaces for the same classification task. In the present paper, the information different information sources are the different languages.

The choice for recognition of geographical named entities was made because gazetteers were found to be particularly useful for this NE-class (Mikheev et al. 1999).

## 2 Approach

### 2.1 Memory-based learning

Memory-based Learning (MBL) is an approach that is based on the idea that the direct use of examples is a better method to learn a solution to certain tasks than learning from rules deduced from examples. In the first phase labelled training examples are presented to the classifier. This set is treated as a collection of points in a multi-dimensional feature space, which is stored in the memory as an instance base. In the second phase, unseen and unlabelled test examples are classified by matching them to every instance in the instance base, calculating the matching distance between the new instance and every instance in the memory using a distance function. A class label is then assigned to the new instance according to the distance (Daelemans and Van den Bosch 2005). The approach in this work

[^47]uses a $k$-Nearest Neighbour classifier ( $k$-NN) (Dasarathy 1991). A class label is assigned by selecting the $k$ examples with the smallest distance to an instance. In this experiment the nearness is calculated via the overlap metric (equations 1.1 and 1.2) but various metrics are applicable depending on the nature of the data. $\Delta(Y, Y)$ is the distance between instances $X$ and $Y$, both represented by $n$ features, with $\delta$ the distance per feature. The distance between two instances is the sum of the differences between the features.
\[

$$
\begin{equation*}
\Delta(X, Y)=\sum_{i=1}^{n} \delta\left(x_{i}, y_{i}\right) \tag{1.1}
\end{equation*}
$$

\]

where:

$$
\delta\left(x_{i}, y_{i}\right)= \begin{cases}a b s & \text { if numeric, else }  \tag{1.2}\\ 0 & \text { if } x_{i}=y_{i} \\ 1 & \text { if } x_{i} \neq y_{i}\end{cases}
$$

The implementation used in this paper is the TiMBL package (Daelemans et al. 2004).

### 2.2 Data

Three corpora of closely related languages were used for this experiment: an English, a Dutch, and a German corpus. In order to keep the experiments as language-independent as possible no preprocessing steps other than tokenising the data were undertaken. In the remainder of this section the characteristics of each corpus are described.

The Reuters Corpus Volume 1 (RCV1) For English we used RCV1 ${ }^{3}$ which contains approximately 810,000 English news articles from the Reuters press agency. It contains newswire stories from August 20, 1996 until August 19, 1997, covering a wide variety of topics such as corporate or industrial news, economics, war, and sports.

The ILK corpus The Dutch data used in the work described in this paper comes from the ILK Corpus. ${ }^{4}$ This corpus was gathered by the Tilburg Induction of Linguistic Knowledge research group from various southern Dutch regional newspapers between 1985 and 1998. It consists of about 230,000 articles that together contain approximately 120 million words. The corpus has been partly annotated with prosody markers, named entities and it is NP-chunked, although these annotation layers were not included in this experiment because the other corpora were not annotated with this information.

The Frankfurter Rundschau corpus The last corpus used for this work is the part of the Frankfurter Rundschau Corpus that was made available for the Elsnet European Corpus Initiative. ${ }^{5}$ It consists of German newspaper texts from the Frankfurter Rundschau from July 1992 until March 1993. It contains about 34 million words.

[^48]
### 2.3 Bootstrapping Gazetteers

The approach undertaken in this work consists of two parts: a language internal bootstrapping loop and a cross-lingual bootstrapping loop. The language internal bootstrapping loop is no different from previous monolingual geographical entity recognition bootstrapping work: a classifier is trained on data automatically labelled with the help of a small gazetteer. The harvest set, containing unseen and unlabelled instances, is then presented to the classifier for which the classifier needs to predict class labels. The items classified as geographical entities are added to the gazetteer and the whole process can be rerun to further expand the gazetteer.

This initial gazetteer, or seed-list, contains 25 items which have been selected manually on the basis of their perceived frequency in the global news. It containes the following geographical names:

| New York | United States of America | U.S. | Mexico | Chile |
| :--- | :--- | :--- | :--- | :--- |
| Paris | Rio de Janeiro | Brazil | The Hague | Great Britain |
| Tokyo | China | Taiwan | Taipei | Bejing |
| Rome | Santiago de Compostela | Afghanistan | Barcelona | Iraq |
| Sydney | Los Angeles | Washington | Pakistan | Buenos Aires |

To decide whether a word is a geographical named entity every item in the gazetteer is checked against every word in the dataset. Upon a match the word is assigned the label 'GN' for 'geographic name', else 'O' for 'other'. Apart from the word and its label, the feature vector for each word also contained contextual and orthographic information. The context was encoded in a 2-1-2 context window, meaning that for each instance the two words before and the two words after the word that is to be classified are given. The other feature that was included here is a marker for whether or not the focus word is capitalised. A small part of the generated training instances is shown in Example 1.1.

$$
\begin{align*}
& \text {-_,-Emerging,evidence,that,+,,O } \\
& \text { _,Emerging,evidence,that,Mexico,-,,O } \\
& \text { Emerging,evidence,that,Mexico,'s,-,O } \\
& \text { evidence,that,Mexico,'s,economy,+,GN }  \tag{1.1}\\
& \text { that,Mexico,'s,economy,was,-,,O } \\
& \text { Mexico,'s,economy,was,bac,,-,O } \\
& \text { 's,economy,was,back,on,-,,O }
\end{align*}
$$

The capitalisation feature was ignored for German because for this language capitalisation is not an informative cue for the presence of a named entity, due to the fact that all nouns are capitalised. Other features, such as whether words surrounding the focus word were capitalised, did not aid classification and were therefore abandoned.

The training instances were generated from $85 \%$ of the data for all three languages, the harvest sets from the remaining $15 \%$. To speed up the experiments and to reduce the influence of false negatives, the ratio positive to negative examples was set to $1: 5$. The false negatives in the labelled data occur because not all geographical entities are
recognised in the first labelling round due to the small gazetteer. From a phrase like "between representatives of England, Wales, Scotland and Ireland" no positive instances will be generated because the seed-list does not contain the items "England", "Wales", "Scotland" or "Ireland".

As well as adjusting the proportion of positive and negative examples, only sentences that contained a geographical entity recognised through the gazetteer were included, as with these a sufficient amount of negative instances could already be generated. In order to not introduce too much noise in the form of false positives in the automatically extracted gazetteer, precision was valued higher than recall.

The classifier was trained on the automatically labelled data and then applied to unseen and unlabelled data. Terms which were labelled as geographical named entities by the classifier were considered candidates to be added to the gazetteer. Since the classifier is not perfect we sought means to extract the items that were classified as geographical names with a high degree of certainty. To this end 4 filters were developed. The first filter checked whether a token that was labelled as a geographical entity had also been labelled as a non-geographical entity, if this was the case the token was discarded. The word "City", for instance, occurs quite frequently as suffix in for example "Atlanta City" or "New York City". However, if this word occurs on its own it is not a geographical named entity and must therefore be discarded. The second filter performed the following check: if a capitalised token classified as a geographical entity also occured non-capitalised in the unlabelled set it was discarded; this also implies that a geographical name needs to be capitalised. Then, completely capitalised items as well as items of three letters and shorter were excluded. Finally, a threshold was put up to exclude items that occurred fewer than 5 times in the harvest set.

For the cross-lingual bootstrapping loop, the gazetteer from the first language (English) is matched to data in another language, in this experiment Dutch or German, to label data to train a classifier on. This yields instances labelled as geographical names because many geographical names, such as 'Amsterdam' are the same across different languages.

Ultimately one would like to perform fuzzy matching of geographical names in the crosslingual bootstrapping loop, e.g. exploiting measures such as Levenshtein distance (Levenshtein 1965) to match "England" to its Dutch form "Engeland". However, this would also yield many false positives such as "France" and "Francs". Hence for the present paper we restricted the cross-lingual bootstrapping loop to strict matching.

## 3 Experiments and Results

### 3.1 Monolingual Experiments

In the first series of experiments the original seed-list was applied to all three data sets in three separate experiments to get an idea of how well the small English oriented seed list works for bootstrapping on the three languages. It also serves as a baseline on which the cross-lingual bootstrapping should improve. Table 1.1 shows how many candidates for

|  | run 1 | new | run 2 | new | run3 | new | total | TP (\%) |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Reuters | 47 | 47 | 47 | 2 | 47 | 0 | 74 | $53(77.0 \%)$ |
| ILK | 117 | 116 | 147 | 51 | 166 | 24 | 216 | $156(76.9 \%)$ |
| Frankfurter R. | 16 | 16 | 17 | 3 | 17 | 0 | 44 | $41(93.2 \%)$ |

Table 1.1: Results of the monolingual bootstrapping runs
extending the gazetteer there were per run ("run 1", "run 2", "run 3"), as well as the number of unique and new items found in each run ("new").

In the first run the classifier is trained on the initial seed, in the second run on the gazetteer that was created in the first run and in the third run the classifier is trained on the gazetteer that was created in the second run. The penultimate column in Table 1.1 shows the total number of items the gazetteer contains after the three runs, i.e., the initial seed-list plus the new items from the first, second and third run. The last column ("TP") shows the percentage of true geographical entities or parts of geographical entities in the gazetteer. The items in the gazetteers were checked manually against atlases to determine whether they are true geographical names or not.

Although we attempted to include multiword entities this has only worked to some degree. Often parts like "River" in for instance "Yangtze River" have not been classified as geographical entities because they occur more often not as a named entity. The unlabelled sets ("harvest sets") from which the new items were harvested consist of $1,000,000$ instances for each language. As can be seen in Table 1.1, the number of new gazetteer items decreases sharply in each run, indicating that if no more data is added, language internal bootstrapping quickly leads to a dead end. Adding more instances to the harvest set could have been a solution here had it not been for the unacceptable number of false positives this yields in this experiment. This is due to the post-processing filter that removes items that have not been classified as geographical names often enough. Raising the threshold proportionally to the increase in instances yields similar results as the first internal bootstrapping experiment in which the harvest set has been kept the same throughout the runs.

### 3.2 Bilingual Experiments

A series of bilingual experiments was conducted to investigate the influence of one other language on the English gazetteer. To this purpose the gazetteer that was created in the first run of the language internal experiment on the English corpus was applied to Dutch, the labelled data was used to train a classifier which was then applied to the Dutch harvest set. After applying the same filters as in the language internal experiments the results were added to the gazetteer. This gazetteer was then applied to the English corpus and the training, classification and filtering were repeated. These three experiments were also carried out for English/German instead of English/Dutch. The results are shown in Table 1.2, where E stands for English, D for Dutch and G for German. The column "run 1" gives the results for the first parts of the experiments ( $\mathrm{E} \rightarrow \mathrm{D}$ and $\mathrm{E} \rightarrow \mathrm{G}$ ), "run

|  | run 1 | new | run 2 | new | total | TP (\%) |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\mathrm{E} \rightarrow \mathrm{D} \rightarrow \mathrm{E}$ | 124 | 122 | 60 | 22 | 216 | $143(66.2 \%)$ |
| $\mathrm{E} \rightarrow \mathrm{G} \rightarrow \mathrm{E}$ | 17 | 17 | 47 | 2 | 91 | $73(80.2 \%)$ |

Table 1.2: Results of the bilingual bootstrapping runs

|  | run 1 | new | run 2 | new | run 3 | new | total | TP (\%) |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\mathrm{E} \rightarrow \mathrm{D} \rightarrow \mathrm{G} \rightarrow \mathrm{E}$ | 124 | 122 | 9 | 9 | 48 | 3 | 206 | $149(72.3 \%)$ |
| $\mathrm{E} \rightarrow \mathrm{G} \rightarrow \mathrm{D} \rightarrow \mathrm{E}$ | 17 | 17 | 129 | 127 | 49 | 3 | 219 | $162(73.1 \%)$ |

Table 1.3: Results of the trilingual bootstrapping runs

2 " gives the results for the second parts $(\mathrm{D} \rightarrow \mathrm{E}$ and $\mathrm{G} \rightarrow \mathrm{E})$. In the column "total" the number of items after these bootstrapping loops is given, i.e. the gazetteer from the English monolingual bootstrapping experiment plus the new items from the bilingual bootstrapping runs. The column "TP" again gives the number of true geographical named entities per final gazetteer, measured manually. In both cases the number of items added to the English gazetteer is greater than in the monolingual experiments. However, more false positives, often person names such as "Adriaanse" and "Cathy", are added to the gazetteer. Especially the English/Dutch gazetteer gets particularly corrupted although the number of true positives is still over twice as much as in the monolingual English gazetteer (143 against 53).

### 3.3 Trilingual Experiments

The third series of experiments concern a trilingual bootstrapping loop. Two different loops have been investigated: English $\rightarrow$ Dutch $\rightarrow$ German $\rightarrow$ English and English $\rightarrow$ German $\rightarrow$ Dutch $\rightarrow$ English. The setup of the experiments is similar to the bilingual experiments, with the addition of an extra experiment on a third language. The results are shown in Table 1.3. Compared to the English/Dutch gazetteer the precision has increased, which is probably due to the conservative behaviour of the German classifier. Also the number of true positives has increased to 162 for the English-to-German-to-Dutch loop, indicating that conservativeness especially in the early runs seems to pay off.

### 3.4 Recall

In order to estimate the recall of lookup with the different gazetteers, the seed list and gazetteers from the experiments were matched to the Dutch CoNLL shared task 2002 test set and the English and German CoNLL 2003 test sets (Tjong Kim Sang 2002; Tjong Kim Sang and De Meulder 2003). The results are presented in Table 1.4. The first number in each column is the recall, the second (in brackets) the precision. The numbers in brackets behind the names of the test sets is the number of geographical names present in that test set. As can be expected with the still small gazetteers, also after bootstrapping, it comes

|  | English (1660) | Dutch (772) | German (1285) |
| :--- | :---: | :---: | :---: |
| seed list | $9.4 \%(100 \%)$ | $2.8 \%(100 \%)$ | $0.5 \%(100 \%)$ |
| English | $9.8 \%(100 \%)$ | $3.1 \%(100 \%)$ | $0.5 \%(100 \%)$ |
| Dutch | $11.3 \%(99.5 \%)$ | $12.7 \%(100 \%)$ | $2.7 \%(65.7 \%)$ |
| German | $9.5 \%(100 \%)$ | $3.1 \%(100 \%)$ | $0.5 \%(100 \%)$ |
| $\mathrm{E} \rightarrow \mathrm{D} \rightarrow \mathrm{E}$ | $12.3 \%(99.5 \%)$ | $6.9 \%(100 \%)$ | $2.3 \%(80.0 \%)$ |
| $\mathrm{E} \rightarrow \mathrm{G} \rightarrow \mathrm{E}$ | $11.3 \%(99.5 \%)$ | $6.6 \%(100 \%)$ | $2.3 \%(79.3 \%)$ |
| $\mathrm{E} \rightarrow \mathrm{D} \rightarrow \mathrm{G} \rightarrow \mathrm{E}$ | $11.7 \%(99.5 \%)$ | $6.8 \%(100 \%)$ | $2.3 \%(79.3 \%)$ |
| $\mathrm{E} \rightarrow \mathrm{G} \rightarrow \mathrm{D} \rightarrow \mathrm{E}$ | $11.3 \%(100 \%)$ | $8.4 \%(100 \%)$ | $2.3 \%(100 \%)$ |

Table 1.4: Recall and precision (in brackets) on CoNLL shared task test sets
as no surprise that recall on the CoNLL datasets is very low. The German test set proves to contain most unknown geographical entities, although some of its geographical named entities such as "Anne-Frank-Schule" ("Anne Frank School") do not occur in atlases and thus do not fall within our notion of geographical named entities. The goal to focus on precision rather than recall has been reached as in most cases no non-geographical names were flagged, indicating that the false positives in the gazetteers are not words that occur very frequently.

## 4 Conclusions and Future Work

Previous work has shown that bootstrapping is a suitable technique to label unseen data, when an iterative labelling scheme is used that feeds back to a classifier. The results in this work have shown that bootstrapping geographical entities with a memory-based learner can also be used in a cross-linguistic setting. Where precision decreases in monolingual bootstrapping if too much data is added and where the bootstrapping reaches a dead end if no more data is added, cross-lingual bootstrapping provides a way out, with the additional property of being portable to another language. Although the gazetteer does not only contain English geographical entities after the cross-lingual bootstrapping loop such as the German form "Genf" for "Geneva", it shows only a minor increase in recall when applied to an English text - but with a high precision. Moreover, the multilingual gazetteers also seem to work for Dutch and German texts. To make these gazetteers more useful for for instance cross-lingual information extraction a means to link the different names for entities in different languages, such as "Vienna" to "Wenen" and "Wien" needs to be found. For some entity pairs like "Ireland", "Ierland" and "Irland", this might be relatively easy as the words are very similar but for other pairs such as "Geneva" and the German form "Genf", or for "Germany" and its native name "Deutschland", contextual information is needed.

Since the focus of this work has been on precision one aim for future work is to explore this technique further to come to better results on recall. In order to do this we will further experiment with different filters and different cross-overs. As more sophisticated filters are
applied it might also be possible to experiment with fuzzy matching, which may also help link up the entities in different languages. Another interesting avenue of research is to investigate how portable this approach is to other languages. Intuitively this approach should also work for other sets of related languages, such as Spanish, Italian and French, or Norwegian, Swedish and Danish. It is also interesting to find out what effect adding or substituting one language with a slightly more distant language such as French would have on this approach.

## Acknowledgements

This research is funded by NWO (Netherlands Organisation for Scientific Research) and carried out at the Naturalis Research Labs in Leiden. The author would like to thank Antal van den Bosch and Caroline Sporleder for discussions and advice, and the anonymous reviewers for their helpful comments.

## Bibliography

Blum, A. and T. Mitchell (1998). Combining labeled and unlabeled data with co-training.
In Proceedings of the 11th Annual Conference on Computational Learning Theory, pp. 92-100.
Callison-Burch, C. and M. Osborne (2003). Co-training for statistical machine translation. In Proceedings of the 6th Annual CLUK Research Colloquium, Edinburgh, Scotland.
Daelemans, W. and A. Van den Bosch (2005). Memory-based language processing. Cambridge: Cambridge University Press.
Daelemans, W., J. Zavrel, K. Van der Sloot, and A. Van den Bosch (2004). TiMBL: Tilburg Memory Based Learner version 5.1, reference guide. Technical Report ILK Technical Report Series 04-02, ILK/Tilburg University, Tilburg, The Netherlands.
Dasarathy, B. V. (1991). Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques. Los Alamitos, CA: IEEE Computer Society Press.
Diab, M. and P. Resnik (2002). An unsupervised method for word sense tagging using parallel corpora. In Proceedings of the 40 th Annual Meeting of the Association of Computational Linguistics (ACL'02), Philadelphia, PA, pp. 255-262.
Jones, R., A. McCallum, K. Nigam, and E. Riloff (1999). Bootstrapping for text learning tasks. In IJCAI-99 Workshop on Text Mining: Foundations, Techniques and Applications, pp. 52-63.
Levenshtein, V. I. (1965). Binary codes capable of correcting deletions, insertions, and reversals. Cybernetics and Control Theory 10(8), 707-710. Originally in: Doklady Nauk SSSR 163(4): 845-848 (1965).
Merlo, P., S. Stevenson, V. Tsang, and G. Allaria (2002). A multilingual paradigm for automatic verb classification. In Proceedings of the 40 th Annual Meeting of the Association of Computational Linguistics (ACL'02), Philadelphia, PA, pp. 207-214.
Mikheev, A., M. Moens, and C. Grover (1999). Named entity recognition without gazetteers. In Proceedings of EACL, pp. 1-8.
Niu, C., W. Li, J. Ding, and R. K. Srihari (2003). A bootstrapping approach to named entity classification using successive learners. In Proceedings of the Association for Computational Linguistics (ACL), pp. 335-342.

Pratim Talukdar, P., T. Brants, M. Liberman, and F. Pereira (2006). A context pattern induction method for named entity extraction. In Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X), New York, NY.

Riloff, E. and R. Jones (1999). Learning dictionaries for information extraction by multilevel bootstrapping. In Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99), pp. 474-479.
Riloff, E., C. Schafer, and D. Yarowsky (2002). Inducing information extraction systems for new languages via cross-language projection. In Proceedings of the 19th International Conference on Computational Linguistics (COLING'02), Taipei, Taiwan.
Tjong Kim Sang, E. F. (2002). Introduction to the CoNLL-2002 shared task: Languageindependent named entity recognition. In Proceedings of CoNLL-2002, pp. 155-158.
Tjong Kim Sang, E. F. and F. De Meulder (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In Proceedings of CoNLL-2003, pp. 142-147.
Uryupina, O. (2003). Semi-supervised learning of geographical gazetteers from the internet. In Proceedings of the HLT-NAACL Workshop on the Analysis of Geographic References, pp. 18-25.

# Web Question Answering by Exploiting Wide-Coverage Lexical Resources 

Michael Kaisser<br>School of Informatics, University of Edinburgh<br>M.Kaisser@sms.ed.ac.uk


#### Abstract

This paper describes an approach to Question Answering (QA) that uses the linguistic information available in lexical resources like FrameNet, PropBank or VerbNet to find on the web, answers to natural language questions. The approach is realized in a system I am currently developing. The paper gives a preliminary evaluation and reports on initial experiences that reveal the challenges still ahead.


## 1 Introduction

Techniques used in Question Answering differ depending on where answers are to be found. If we restrict ourselves to natural language data we can distinguish two kinds of corpora for which different strategies are appropriate:

1. If the data is in a static, off-line corpus, it can be preprocessed, annotating or even restructuring the documents in a way that makes the actual querying more efficient. Syntactic and semantic information can automatically be added to the surface structure, and indices can be created that allow querying based on such categories.
2. If the data is in a dynamic, on-line corpus, such a strategy is difficult to pursue. The corpus would need to be constantly checked for updates and the new information would need to be processed and integrated. In the case of the web, its huge size creates even more problems: It would be extremely time consuming to annotate the complete web, and as annotation significantly increases the data volume, much more storage capacity is needed.

The approach proposed in this paper focuses on queries made to the second kind of corpus, but also exploits syntactic and semantic information, albeit in a way that differs from the first strategy. Here such information, extracted from wide-coverage lexical resources, is used to create alternative queries (paraphrases) that can lead to alternative sentences being found in the corpus. Such sentences, if found, are likely to contain an answer. Only those sentences will be further analyzed and used to construct an answer to the question.

For reasons of clarity, I will restrict my remarks in this paper to FrameNet (Fillmore and Lowe 1998), although I am exploring the use of PropBank (Palmer et al. 2005) and VerbNet (Schuler 2005) as well. Although the general idea shared between all these resources in basically the same, details differ considerably. (See Ellsworth et al. for a comparison between PropBank, FrameNet and the German FrameNet-based SALSA.)

## 2 FrameNet

FrameNet is a lexical database resource based on frame semantics and supported by corpus evidence. It documents the range of semantic and syntactic combinatory possibilities (valences) of target words (lexical units). In order to do this, it contains human-annotated sentences (currently more than 135,000 ), which exemplify the use of more than 6,100 lexical units organized into 625 semantic frames. As FrameNet is still in ongoing development, not all lexical units contain annotated sentences yet.

In FrameNet, words with similar semantics receive descriptions with identical role labels. The lexical units for "invent" and "design" for example both describe the relation between the roles (frame elements in FrameNet's terminology) Cognizer and Invention. Similarly, the frames for the verbs "buy" and "sell" both list the frame elements Buyer and Seller. The annotated sentences show that the position of these frame elements differs across different syntactic realizations of the same verb and across different verbs with similar role sets.

The purpose of FrameNet is to create a sample selection of how Natural Language works. Many applications employing FrameNet make use of the information the annotated sentences provide and apply it to sentences from other sources. Thus, FrameNet can help in getting closer to understanding Natural Language sentences or even texts. In the following I describe how I use FrameNet to

1. understand questions;
2. create a set of exact, alternative search engine queries;
3. analyze the text in the snippets returned by the search engine in order to find exact answers to the question.

## 3 Motivation

The acquisition and use of paraphrases or patterns for QA has so far been explored by various researchers: Lin and Pantel (2001), for example, use dependency path transformations to discover paraphrases. Ravichandran and Hovy (2001) use a machine learning method which is fed with a few hand-crafted examples to find surface text patterns.

The strength of the current approach is that a manually annotated corpus provides a much more reliable basis for paraphrase generation than the automatic approaches described above. With this approach, in theory, all retrieved paraphrases are valid, and-
because of the size of the lexicon-it can be expected that for a lot of questions paraphrases can be found. (Problems that arise in practice are described in the following sections of this paper.)

Another important advantage is that FrameNet does not only contain syntactic, but also semantic information. This can be exploited in many ways, e.g. for answer type checking. Furthermore the output of a QA system based on resources like FrameNet can not only provide an answer as an result, but also a semantic analysis describing the relation between constituents from the question and the answer. In other words, the paraphrases retrieved are neither surface patterns, nor pure syntactic transformations. Instead, they represent possibilities of how to express one and the same fact in different syntactic ways and they contain descriptions of the semantic roles for each of the constituents.

Please note that the work described in this paper builds largely on the system with which I participated in TREC 2004 (Kaisser 2004). The QA system described there also makes use of paraphrases, but they come from a hand-crafted pattern set. This paper describes extensions made to that system and extensions planned for the future, it will not repeat a complete system description. For issues like answer candidate processing etc., I would like to point the interested reader to the aforementioned paper.

## 4 Walking through an Example

In this section I will give a short explanation of how the system processes the question "When was the telegraph invented?"

First, the incoming question is parsed using MiniPar (Lin 1998), and the resulting dependency tree is simplified to the following structure:

```
head: invented(V)
subj: Who
whn: Who
obj: the telegraph
```

head indicates that the head of the question is the verb invented, subj indicates that the deep subject is who (which whn marks as also being a question word) and obj indicates that the deep object is the telegraph.

This provides enough information to look up the head verb in the FrameNet dictionary, where two lexical units for invent.v can be found. ${ }^{1}$ One of the entries contains annotated sentences including the following:

| Du Pont FE:Cognizer | in the USA | had | INVENTED lexical unit | nylon <br> FE:Invention | in the late 1930s |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: |

Parts of the sentences are annotated with frame elements, here Cognizer and Invention. The system parses all such annotated sentences to find out which semantic roles are assigned

[^49]to which syntactic roles. It shows that usually, in active sentences, the Cognizer role is realized as an NP at subject position, while Invention is an NP at object position.

As mentioned earlier, the analysis of the question showed that, in a potential (active) answer sentence, the answer should be in subject relation to the verb "invent". Furthermore "the telegraph" needs to be in object relation to the verb. From this it can be concluded that the filler for the Invention frame element is "the telegraph", and that the question asks for a Cognizer. The system can now give a pseudo-semantic formula for the question

```
invent_272(Cognizer=X, Invention="the telegraph")
```

and replace the frame elements Cognizer and Invention in each annotated sentence with their values from the question. For the above sentence the outcome would be:

```
ANSWER(NP) in the USA had invented the telegraph, in the late 1930s ...
```

The PPs "in the USA" and "in the late 1930s" are recognized as additional information, most likely specific to the topic of the initial annotated sentence, but not transferable to the new domain, so they are-in the current version of the system-simply removed:

ANSWER(NP) had invented the telegraph
This can straightforwardly be translated into the kind of pattern used in my TREC 2004 system (Kaisser 2004). These patterns describe reformulation rules stating (for example) that for a question like "Who invented the telegraph?" a proper answer might look like "NP had invented the telegraph".

From here the strategy is the same as in TREC 2004:

1. The system generates Google queries from the patterns, in this case:
"had invented the telegraph".
2. It extracts sentences from the Google snippets.
3. It parses these sentences and checks whether they have the required syntactic structure.
4. If a sentence has the correct syntax, the potential answer can be extracted, because the system knows from the FrameNet data where in that sentence the answer is located. For example in "By 1832 Samuel FB Morse had invented the telegraph." it must be the NP preceding "had", thus: "Samuel FB Morse".

For the given example, the system was able to find the correct, exact answer and the open proposition shown above can now be completed:

```
invent_272(Cognizer="Samuel FB Morse", Invention="the telegraph")
```


## 5 Coverage and Preliminary System Evaluation

I have evaluated the system with the 500 questions from the TREC 2002 test collection. The system returned a head verb for all 500 questions. In 246 cases the head verb was simply be. In FrameNet, this usually means that the lexical unit to look up is not a verb but a noun or an adjective, e.g. in "Who is the governor of Colorado?" the system would have to look up "governor" while in "How high is Mount Kinabalu?" it should go for "high". As the system currently only deals with heads that are verbs, these other cases have been excluded from the evaluation.

Ignoring "be", 254 look-ups were performed, in 189 cases ( $74 \%$ ) at least one lexical unit containing annotated sentences was found, while in 65 cases no lexical unit existed for the word or an existing lexical unit contained no annotated sentences. This means that in $26 \%$ of all cases, the current FrameNet release (2.1, at the time of writing) does not provide annotated sentences for the verb that was looked up.

For the 189 cases in which the system theoretically was able to properly process the question it returned 70 answers, of which 44 were correct, 6 inexact and 20 wrong.

The 129 cases where the system did not return an answer fall into four categories:

1. No mapping from question parts to frame elements could be achieved. (45\%)
2. The mapping from question parts to frame elements was not correct or incomplete. (34\%)
3. The generated queries produced zero hits on Google. (12\%)
4. The retrieved Google snippets were analyzed by the system to not contain an answer. (9\%)

The 20 wrong answers were due to incomplete mappings (30\%), wrong mappings ( $40 \%$ ) or other reasons ( $30 \%$ ), e.g. assignment of a wrong answer type or incorrect recognition of phrase boundaries for the answers.

As mentioned, the system is still under development. I think, the figures suggest that it is worth following the described idea further, but that mapping from question parts to frame elements has to be significantly improved.

## 6 Further Work: Answer Types

An important component of nearly every QA system is concerned with checking that answers are of the correct semantic type: QA systems usually know a question like "When was Franz Kafka born?" should be answered with a date, while "Who invented the telegraph?" asks for a person.

Currently the system borrows its answer types from the hand-written rephrasing rules used in the TREC 2004 system. This is, however, not a proper solution: I plan to replace this component by an approach based on an analysis of frame element fillers. The basic

Fillers for FE Cognizer

| Count | Name |
| :--- | :--- |

Pronouns:

| 861 | Pronoun sing. |
| :---: | :---: |
| 266 | Pronoun pl. |

Named Entities.

| 198 | Person |
| :---: | :---: |
| 22 | Organization |
| 16 | Location |

WordNet:

| 2205 | entity (id: 1740) |
| :---: | :---: |
| 1232 | object (id: 16236) |
| 794 | living thing (id: 3009) |
| 794 | organism (id: 3226) |
| 776 | causal agent (id: 5598) |
| 758 | person (id: 6026) |
| 459 | group (id: 26769) |
| 323 | social group (id: 7470450) |
| 217 | organization (id: 7523126) |
| 203 | artifact (id: 19244 ) |

Fillers for FE Invention

| Count | Name |
| :--- | :--- |
| Pronouns: |  |

Pronouns:

| 0 | Pronoun sing. |
| :---: | :---: |
| 4 | Pronoun pl. |

Named Entities:

| 0 | Person |
| :---: | :---: |
| 0 | Organization |
| 0 | Location |

WordNet:

| 287 | entity (id: 1740) |
| :---: | :---: |
| 226 | object (id: 16236) |
| 162 | abstraction (id: 16236) |
| 110 | relation (id: 27929) |
| 107 | psychol. feature (id: 20333) |
| 103 | cognition (id: 20729) |
| 98 | whole (id: 2645) |
| 98 | artifact (id: 19244) |
| 97 | social relation (id: 28549) |
| 97 | communication (id: 28764 ) |

Figure 1.1: These tables give information about the semantic content of the fillers for the frame elements Cognizer (2370 occurrences in the FrameNet data) and Invention (176 occurrences).
assumption is that most frame elements have dedicated semantic classes that their fillers can come from.

Figure 1.1 shows the results of an experiment done to test this assumption. Every string from the sentences in the FrameNet data that was annotated as either Cognizer or Invention was checked whether it was one of the words "I", "you", "he" or "she" (listed as Pronoun sing.) or "we" or "they" (Pronoun pl.). If it was not, the string was passed on to ANNIE (Cunningham et al. 2002), a Named Entity recognition system. If that produced no result, the system tries to find the matching WordNet (Miller et al. 1993) entry. When checking WordNet, word sense disambiguation was ignored: All hypernyms of all senses of the head noun of the phrase in question were added. The idea was that if all senses of a filler are taken into account, then one of them must always be the intended one. As not only one word is checked, but many, an accumulation for the correct sense can be expected. The wrong hits should be evenly distributed over other synsets and thus represent mere statistical noise.

The results seem to match one's intuition. The fillers for Cognizer are personal pronouns, persons (as recognized by ANNIE) or persons, groups or organizations in WordNet. Only artifact is misleading: A lot of fillers for Cognizer were rather general names for
organizations, e.g. "the bank", "the media" or "the court", all these expressions actually denote artifacts, but metonymy allows them to be used for organizations or groups of people as well.

As expected, not many personal pronouns or Named Entities can be found for the Invention frame element. The WordNet entries can be understood when taking a look at the different lexical units that make use of frame element Invention. People formulate "policies", "ideas" and "rules" (all cognitions), they design "a product", "the house" or "clothes" (artifacts), whereas "speeches" or "screenplays" (communications) tend to be invented.

It is planned to use the described data to check whether an answer candidate that is found by the system fits in the semantic constraints that the answer frame element describes. Furthermore, these filler lists can be useful when tackling the challenges of automatic role assignment and word-sense disambiguation, as will be described in the next two sections of this paper.

## 7 Further Work: Role Assignment

The correct interpretation of the question-i.e. the detection of the lexical unit to look up and the correct assignment of parts of the question to frame elements-is crucial for the sketched approach. (See the evaluation given in section 5.) If errors occur at this processing step, all following steps will be affected in a way that usually does not allow the correct answer to be retrieved. Unfortunately, experiences so far show that this step is also the most difficult one to implement.

In recent years, a lot of work has been done on automatic labeling of semantic roles. Most proposed solutions use statistical means to solve the problem. In Gildea and Jurafsky (2000) the authors train a classifier with the following features: phrase type, grammatical function, position (Is the constituent before or after the predicate defining the frame?), voice and the head word of the constituents. When running the classifiers on a unlabeled test set the authors report an accuracy between $79.6 \%$ and $80.4 \%$ depending on the method how evidence from the features is combined. ${ }^{2}$

Most work following the Gildea and Jurafsky experiments, sticks to the general idea described there, but modifies the feature set and/or the classifier used. Xue and Palmer (2004) show that a more careful feature selection-especially by taking more information from the target sentence's parse trees into account-leads to an overall better performance. Furthermore they argue that the argument identification and the argument classification subtask require the use of different features. Chen and Rambow (2003) add even more "deep linguistic features" to their feature set.

Although in the work described in this paper a correct labeling of semantic roles is crucial, there are several differences in the problem description when comparing it with the above cited works. In my approach, it is solely necessary to annotated questions with

[^50]semantic roles. ${ }^{3}$ This difference is crucial for various reasons:

1. Questions have a different syntax than declarative sentences. If one of the described approaches would be used one-to-one, one would either need to start with a set of hand-annotated questions as the training set or expect a drop in accuracy, because of the different nature of the training set and the test set.
2. There always is one semantic role that has to be annotated which is not mentioned in the question, but which is very important, because it represents the answer to the question.
3. Questions (especially TREC-style ones), tend to be shorter and show a smaller range of syntactic variants than declarative sentences (especially those in news paper articles, as found in the corpora used in the above described experiments).

As has been explained above, the role assignment module in the current system is not yet satisfactory. But because of the important nature of this module, I plan to experiment with a wide range of possible solutions in order to solve this problem. Mainly because the syntax of questions tends to be simpler, I hope to be able to develop accurate, nonstatistical methods, based on a syntactic analysis of the question. At the moment, only a few syntactic constraints are taken into account here (see section 4). The algorithm needs to be extended, so that it can deal with more complex syntactic structures. It is also planned to use the collected knowledge about semantic classes of frame elements (as described in section 6) to assist in the role assignment process.

For further help with separating good role assignments from bad ones web counts could be used: Imagine that in the example from above the fillers for the frame element are mistakenly switched:

```
invent_272(Cognizer="the telegraph", Invention=X)
```

In such a case the web will be searched for a rather odd construct like:
the telegraph had invented ANSWER
But the query "the telegraph had invented" produces zero hits on Google, while the query resulting from the correct assignment ("had invented the telegraph") produces 149. So, whenever the algorithm is unsure which role assignment to prefer, it is planed to try out all variants, and take relative web counts into account when making the final decision.

[^51]
## 8 Further Work: Word-Sense Disambiguation

Word-sense disambiguation in this context mainly means deciding between different lexical units that might exist for the word in the question that needs to be looked up. There are for example two entries for invent.v in FrameNet. In the described setup, three ways to disambiguate a question can be differentiated:

1. The syntactic analysis of the question might help to make the decision (e.g. if intransitive and transitive versions of a head verb exist).
2. It can be checked whether the frame element fillers from the question match the semantic class of the frame element they are assigned to. (See 6).
3. Sometimes even an analysis of the answer candidates becomes necessary: In the question "What did Samuel Morse invent?", for both senses of invent.v, "Samuel Morse" would be mapped to Cognizer, because both entries contain a Cognizer frame element, usually at subject position. The answer to the question, however, must be mapped either to frame element Invention or frame element New_idea (both usually found at object position), depending on the selected lexical unit. It is not possible to decide between the two meanings of invent.v until an answer was retrieved. Only then it becomes possible to use selectional restrictions to make this decision.

## 9 Conclusion

I have described the basic working principles behind my Question Answering system, which uses lexical resources as FrameNet to identify possible paraphrases of potential answer sentences. Furthermore, a linkage to FrameNet provides a step towards a better understanding of the question and the answer.

Although the described system is still under development results to date are promising. They also show that there are challenging tasks still ahead, especially in the areas of automatic role assignment and word-sense disambiguation. Nevertheless, there are promising solutions, for example the idea to exploit the observation that questions tend to show a simpler syntax that declarative sentences and the idea of semantically analyzing the fillers for frame elements.

Furthermore, in this paper I restricted myself to describe ways how FrameNet can be used for QA purposes. However, I am-as already mentioned-also taking a closer look at other wide-coverage lexical resources, i.e. PropBank and VerbNet. It should be interesting to see how the corpora's design issues affect the QA system's design and, of course eventually-its performance.

## Bibliography

Chen, J. and O. Rambow (2003). Use of deep linguistic features for the recognition and labeling of semantic arguments. In M. Collins and M. Steedman (Eds.), Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, pp. 41-48.

Cunningham, H., D. Maynard, K. Bontcheva, V. Tablan, and C. Ursu. (2002). Gate user guide. http://gate.ac.uk/.
Ellsworth, M., K. Erk, P. Kingsbury, and S. Pado (2004). Propbank, salsa, and framenet: How design determines product. In Proceedings of the LREC 2004 Workshop on Building Lexical Resources from Semantically Annotated Corpora, Lisbon.
Fillmore, C. F. B. C. J. and J. B. Lowe (1998). The berkeley framenet project. In Proceedings of COLING-ACL.

Gildea, D. and D. Jurafsky (2000). Automatic labeling of semantic roles. In ACL '00: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, Morristown, NJ, USA, pp. 512-520. Association for Computational Linguistics.

Kaisser, M. (2004). Question answering by searching large corpora with linguistic methods. In The Proceedings of the 2004 Edition of the Text REtrieval Conference, TREC 2004.

Lin, D. (1998). Dependency-based evaluation of minipar. In Workshop on the Evaluation of Parsing Systems.

Lin, D. and P. Pantel (2001). Discovery of inference rules for question-answering. Natural Language Engineering 7(4), 343-360.
Miller, G. A., R. Beckwith, C. Fellbaum, D. Gross, and K. Miller (1993). Introduction to wordnet: An on-line lexical database.

Palmer, M., D. Gildea, and P. Kingsbury (2005). The Proposition Bank: An annotated corpus of semantic roles. Computational Linguistics 31(1), 71-106.
Ravichandran, D. and E. Hovy (2001). Learning surface text patterns for a question answering system. In ACL '02: Proceedings of the 40 th Annual Meeting on Association for Computational Linguistics, Morristown, NJ, USA, pp. 41-47. Association for Computational Linguistics.
Schuler, K. K. (2005). VerbNet: A broad-coverage, comprehensive verb lexicon. Ph. D. thesis.

Xue, N. and M. Palmer (2004, July). Calibrating features for semantic role labeling. In D. Lin and D. Wu (Eds.), Proceedings of EMNLP 2004, Barcelona, Spain, pp. 88-94. Association for Computational Linguistics.

# DepAnn - An Annotation Tool for Dependency Treebanks 

Tuomo Kakkonen<br>Department of Computer Science, University of Joensuu, Finland<br>tuomo.kakkonen@cs.joensuu.fi


#### Abstract

DepAnn is an interactive annotation tool for dependency treebanks, providing both graphical and text-based annotation interfaces. The tool is aimed for semi-automatic creation of treebanks. It aids the manual inspection and correction of automatically created parses, making the annotation process faster and less error-prone. A novel feature of the tool is that it enables the user to view outputs from several parsers as the basis for creating the final tree to be saved to the treebank. DepAnn uses TIGER-XML, an XML-based general encoding format for both, representing the parser outputs and saving the annotated treebank. The tool includes an automatic consistency checker for sentence structures. In addition, the tool enables users to build structures manually, add comments on the annotations, modify the tagsets, and mark sentences for further revision.


## 1 Introduction

Treebanks, collections of syntactically annotated sentences, are needed for developing and evaluating natural language processing (NLP) applications, as well as for research in empirical linguistics. The earliest treebanks, constructed in 1970's, were annotated manually (Abeillé 2003). As treebank construction is labor-intensive, methods are needed for automating part of the work. The reason that treebanks are not constructed fully automatically is obviously the fact there are no parsers of free text capable of producing error-free parses. In semi-automatic treebank building, the work of an annotator is transformed from a tree builder to a checker and corrector of automatically created structures. Constructing a treebank semi-automatically calls for a range of tools, such as a part-of-speech (POS) tagger, a syntactic parser and an annotation tool.

In recent years, there has been a wide interest towards dependency-based annotation of treebanks. Dependency grammar formalisms stem from the work of Tesniére (Tesniére 1959). Most often the motivation for basing the treebank format on dependency is the fact that the language for which the treebank is developed for has a relatively free word order. In such languages, due to their rich morphology, there is more freedom in word order for expressing syntactic functions. In dependency-based grammars, only the lexical nodes are recognized, and the phrasal ones are omitted. The lexical nodes are linked with directed binary relations. The dependency structure of a sentence thus consists of a number of
nodes which is equal to the number of words in the sentence, a root node and the relations (dependency links) between the nodes.

Although more collaboration has emerged between treebank projects in recent years, the main problem with current treebanks in regards to their use and distribution is the fact that instead of reusing existing annotation and encoding schemes, new ones have been developed. Furthermore, the schemes that have been developed are often designed from theory and even application-specific viewpoints, and consequently, undermine the possibility for reuse. In addition to the difficulties for reuse, creating a treebank-specific representation format requires developing a new set of tools for creating, maintaining and searching the treebank.

The main motivation for designing and implementing DepAnn (Dependency Annotator), an annotation tool for dependency treebanks, stems from the need to construct a treebank for Finnish. As Finnish is a language with relatively free word order, dependencybased annotation format is a straight-forward choice as the basis for the annotation. Although DepAnn is customized to be used for creating the Finnish treebank, the choices made in the architecture and design of the system allow it to be modified to the needs of other treebank projects. Most importantly, DepAnn uses a XML-based abstract annotation format, TIGER-XML (Mengel and Lezius 2000) as both input and output formats.

This paper represents the main design principles and functionality of DepAnn. In addition, we describe how the system interacts with the other treebanking tools (POS taggers, morphological analyzers, and parsers). Section 2 shortly describes the principles of treebank construction. Section 3 represents the requirements defined for DepAnn based on an analysis of existing annotation tools, and describes the tool. Finally, in Section 4 we give concluding remarks and underline some future possibilities.

## 2 Background

Speed, consistency, and accuracy are the three key issues in treebank annotation. The most commonly used method for constructing a treebank is a combination of automatic and manual processing. Constructing a treebank, even with a semi-automatic method, is a labor-intensive effort. Efficient tools play a key role in lowering the costs of treebank development and enable larger, higher quality treebanks to be created. Both goals are crucial. The estimated costs of the Prague Dependency Treebank, the largest of the existing dependency treebanks, are USD 600,000 (Böhmová, Hajič, Hajičová, and Hladká 2003). A treebank has to be large enough to have any practical use, for example for grammar induction. The size of the existing dependency treebanks is quite limited, ranging from few hundreds to 90,000 sentences. Self-evidently, a treebank has to be also consistent and have a low error frequency to be useful.

A morphological analyzer and a parser should be applied in order to lower the burden of the annotators. The typical procedure is to use a parser that leaves at least part of ambiguities unresolved and dependencies unspecified, and let human annotators to do the inspection and correction of the parses. Thus, an annotator is correcting the POS and
morphosyntactic tags, resolving the remaining ambiguities and adding and correcting any missing or erroneous dependencies. A crucial component in this type of semi-automatic treebank creation is the annotation tool. A well-designed and well-implemented tool can aid the work of annotators considerably. With an annotation tool, the user can browse, check, and correct the parser's output as well as create structures from scratch. In some of the existing tools the annotations are automatically checked against inconsistencies before saving them to the treebank. In addition, the user is able to add comments to the structures or mark them as doubtful.

Dependency treebanks have been built for several languages, e.g. Czech (Böhmová et al. 2003), English (Rambow et al. 2002), Danish (Bick 2003; Kromann 2003), Italian (Lesmo, Lombardo, and Bosco 2002), and Dutch (van der Beek, Bouma, Malouf, and van Noord 2002). The TIGER Treebank of German is an example of a treebank with both phrase structure and dependency annotations (Brants, Dipper, Hansen, Lezius, and Smith 2002). The current direction in the thinking in the dependency vs. constituency discussion in general is on integration and cooperation (Schneider 1998). While dependency grammars are superior in handling free word order, on one hand some elements of constituency grammars are better for handling certain phenomena (e.g. coordination), and on the other hand, constituency-based grammars also need dependency relations, at least for verb valency. Furthermore, dependency structures can be automatically converted into phrase structures (Xia and Palmer 2000) and vice versa (Daum, Foth, and Menzel 2004), although not always with $100 \%$ accuracy.

We started designing a treebank for Finnish by analyzing the methods and tools used by other dependency treebank projects. The producers of the dependency treebanks have in most cases aimed at creating a multipurpose resource for research on NLP systems and theoretical linguistics. Some, e.g. the Alpino Treebank of Dutch (van der Beek et al. 2002), are built for a specific purpose. Most of the dependency treebanks consist of newspaper text and are annotated on POS, morphological and syntactic levels. An interested reader is referred to (Kakkonen 2005) for further details on the analysis of dependency treebanks.

After a throughout study of existing annotation methods and tools (such as GRAPH (Böhmová, Hajič, Hajičová, and Hladká 2003), Abar-Hitz (Díaz de Ilarraza, Garmendia, and Oronoz 2004), Annotate (Plaehen and Brants 2000), DTAG (Kromann 2003)), CDG SENtence annotaTOR (SENATOR) (White 2000), it was found that none of the available annotation tools satisfied all our requirements. Some tools were not suitable for dependency annotation, some were not compatible with any common XML-based annotation formats, the user-interface was not considered suitable or the tool didn't have all the functions we required. In addition, to our knowledge there aren't any annotation tools available capable of showing or merging outputs from several parsers for aiding the annotator's choices. Thus, the decision was made to design and implement an annotation tool with all the desired characteristics.

## 3 The Annotation Tool

### 3.1 Design Principles

The analysis of existing annotation tools was crucial in defining the requirements for the system to be developed. The following key features were recognized:

- Support for an existing XML encoding scheme

Building a treebank is such a labor-intensive effort that promoting co-operation between treebank projects and reuse of formats and tools is an important and widely accepted goal in treebanking community (e.g. (Ide and Romary 2003)). Using an existing encoding format will make the system reusable. In addition, existing tools supporting the same scheme can be used for browsing, manipulating and searching the annotated treebanks.

- Both textual and graphical display and manipulation of parse trees

For any annotation tool the capability to visualize the sentence structures is a necessity. In addition, the graphical view should preferably be interactive, so that the user can manipulate the structures. On the other hand, for some annotation tasks or for some user's needs textual view of the structure may be more suitable.

- An interface to morphological analyzers and parsers for constructing the initial trees In order to generate the initial trees for human inspection and modification, the annotation tool must have an interface to a morphological parser, a POS tagger and a syntactic parser. The tool should be able to use simultaneously outputs from several tools to guide the annotator's decisions.
- An inconsistency checker for both structures and encoding

The annotated sentences to be saved to the treebank should be checked against tagging inconsistencies. In addition to XML-based validation of encoding, the inconsistency checker should inform the annotator about several other types of mistakes, such as mismatching combinations of POS and morphological tags, missing main verb, and fragmented, incomplete parses.

- Menu-based tagging

In order to make the annotation process faster, setting the tags should be done by means of selecting the most suitable tag from a pre-defined set of tags, instead of requiring the annotator to type the tag label. In addition to being efficient, menubased tagging lowers the number of errors as there will be no errors cost by typos in the labels. On the other hand, keyboard shortcuts for selecting appropriate tags should be provided for more advanced users.

- A commenting tool

For easing the later revisions, possibly performed by other annotators, the user should be able to add comments on the annotated structures. In addition, user should be
able to mark a sentence as ready or unfinished to make it easier to locate sentences needing further revision.

The foremost design principle, apart from making the annotation process faster and less error-prone, was that the tool must be reusable and modifiable. The system was designed in way that the modules for processing the treebank output and input are kept separate from the structure viewing and manipulation modules, thus making the tool more easy to modify. The support for an existing encoding scheme is a crucial reusability feature of any treebanking software. The selection of the format was first narrowed down by the decision that the format should be XML-based, as XML offers a set of validation capabilities, in order to automatically check for encoding inconsistencies.

The aim of an abstract annotation model is to provide a general framework for linguistic annotation. Existing abstract annotation formats share the common goal of offering an intermediate level between the actual data (encoding scheme) and the conceptual level of annotation (annotation scheme). An advantage of such an approach is to enable a common set of tools to be used for creating and manipulating treebanks in several formats. From the set of possible option, including e.g. XCES (Ide and Romary 2003), TIGER-XML (Mengel and Lezius 2000) was selected to be used in DepAnn. TIGER-XML is an exchange format for corpora and treebanks, providing an XML-based representation format which is general enough for representing diverse types of corpus and treebank annotations (Mengel and Lezius 2000). The format is based on encoding of directed acyclic graphs (DAGs). Each DAG represents a sentence as terminal (i.e. words) and nonterminal (dependencies) nodes. The syntactic categories, POS, lemma and other information is represented as attributes in the nodes. The edges encode labeled links between terminals and nonterminals.

TIGER-XML has several desirable characteristics: First, it is flexible and extensible enough to accommodate different treebank annotation types, both dependency and consistency based. Second, it has been shown to be suitable for dependency annotation in several treebank projects (e.g. TIGER Treebank (Brants et al. 2002), Arboretum (Bick 2003)). Third, there are explicit specifications available how to encode dependency structures in the scheme (Kromann 2004). And finally, there exists a set of well-implemented tools supporting the format, such as TIGERSearch viewing/query tool and TIGERRegistry indexing tool (König, Wolfgang, and Voormann 2003), capable of transforming some wellknown corpus and treebank formats, such as the SUSANNE (Sampson 1995) and Penn Treebank (Marcus, Santorini, and Marcinkiewicz 1993) into TIGER-XML.

As TIGER-XML is a general model of treebank encoding, it would be possible to show and manipulate constituency structures with DepAnn. However, the decision was made that the tool was not going to be designed for both constituent and dependency structures in a suspicion that too general design would hamper the efficiency of dependency annotation. Thus, the visualization functions and the user interface are tuned for manipulating dependency structures.


Figure 1.1: The inputs and outputs of the tool.

### 3.2 Main Functionality

In DepAnn tool, the structure to be annotated is represented to the user in textual and graphical formats in order to offer the best option for each user's needs. The textual and graphical views are fully integrated, thus the changes applied in the graphical view immediately affect the textual one and vice versa. The user interface is customizable to suit the task and the annotator's preferences. The user can add comments on annotations, reminding on problematic parts on the sentence structures. Completed trees can be marked as ready, indicating that no further inspection and modifications are needed.

Outputs of several parsers and POS taggers can be applied in parallel to offer the annotator a possibility to compare the outputs in order to guide the annotation decisions. To be able to use the output of an parser in DepAnn, a converter must be implemented to transform the output from the parser or tagger-specific format to the format used by DepAnn. TIGER-XML (Mengel and Lezius 2000) is used as the input format for the structures obtained from the automatic tools, as well as the output format for the annotated treebank. For internal data representation the TIGER-XML structures are transformed into Java objects. Figure 1.1 illustrates the input and output processes of DepAnn.

The annotation process using DepAnn starts with processing the treebank texts with one or more parsers and taggers. Next, a converter is applied to the outputs in order to transform the tool-specific format into TIGER-XML. After the conversion, the annotator can view the parsed structures and build the annotated structure to be added to the treebank. The user can select the parser output to be used for creating the initial trees. Figure 1.2 illustrates the main frame of DepAnn's user interface.

The main groups of functions are indicated in Figure 1.2 by boxes A...E. The text field in the area bordered with box A shows the sentence being annotated in raw text format. Area B is a toolbar with controls for treebank browsing (buttons for showing the next and the previous sentence and a slidebar for browsing), checking and saving the sentence, and modifying the tag sets. In C, the user can graphically manipulate the structure by changing the values on nodes representing the words and dependency links and by removing, adding and rerouting the links between the nodes. Area D consists of the revision functions. User


Figure 1.2: The main frame of DepAnn tool.
can mark the sentence as ready, indicating that further revision is not needed. In addition, user can use the comment field to write notes concerning the sentence structure. Box E frames the tables for text-based structure manipulation and viewing.

The parser and tagger outputs for aiding the annotation decisions are shown in a separate resizable, customizable dialog. For example, in a computer system with multiple monitors, the dialog can be placed in to a separate desktop. In the current version, the user can select which parser's output is used as the initial tree for correction and modification. We are working on an extension to the system, in which the initial trees would be created by semi-automatically combining the parsers' and taggers' outputs by the aid of the annotator.

When the user decides to stop editing a sentence, an automatic consistency checking is performed to validate the sentence structure, the annotation, and encoding. First, a series of checks are run to verify that the sentence has a main verb, a root, all the words have word form and lemma information and morphosyntactic tags, and that the sentence is not fragmented etc. Second, if the first series of checks was passed, the sentence is transformed into TIGER-XML and validated against the XML schema to find any errors in encoding. The problems found are indicated to the user. The user can select which checks are run by modifying the system set-up.

### 3.3 Implementation Details

The annotation tool is implemented in Java. As Java is platform-independent, the system can be used in any environment for which Java is available. The system consists of three main components: the interface to parsers and taggers, the annotation tool itself, and the output module. Two freely available open source packages, OpenJGraph (Salvo 2006) and TIGER API (Demir et al. 2006), were used for developing the system, although both had to be modified considerably to be suitable to be used as a part of DepAnn. TIGER API, a Java API for TIGER-XML, is used for input and output processing. The graphical annotation manipulation functionality was build on top of OpenJGraph. The annotation tool uses Java Database Connectivity (JDBC) for storing the outputs from the parsing and tagging tools, as well as for the user comments and information on ready sentences. Thus, the MySQL database currently in use can be replaced by any other JDBC-compatible database.

## 4 Conclusion

The semi-automatic annotation tool for dependency structures discussed in the paper provides graphical and text-based annotation functions, possibility to use outputs from several parsers to aid the annotation decisions, tools for commenting the annotated structures, automatic consistency checking, and support for TIGER-XML format. In its first application, DepAnn will be used for creating a treebank for Finnish, aimed for evaluation of syntactic parsers. Outputs from two parsers/morphological analyzers, Functional Dependency Grammar parser (FI-FDG) (Tapanainen and Järvinen 1997) and Constraint Grammar parser (FINCG) (Karlsson 1990) is transformed to TIGER-XML and represented to the annotator as the basis for creating the correct structure. The tool is implemented in a way that it is adjustable for other treebank projects' needs. As the annotation format is based on TIGER-XML, the tool is not restricted to a particular set of POS, morphological or dependency tags. The modules for processing the treebank output and input are separate from the graphical and textual annotation modules, thus the tool could be modified to use any other annotation format. DepAnn will be made publicly available as an open source distribution.

As mentioned above, the issues related to reuse of tools and formats is one of the major issues in treebanking. Thus, few words on development costs of the annotation tool is in order. The work was conducted by a researcher with a degree in Software Engineering and few years of practical experience in programming and software designing. No exact data was recorded, but the amount of work to design and implement the system to its current state is around a half of a man-year. The work was considerably eased by using open source APIs for treebank manipulation and graph visualization. These observations underline the importance of reusing existing annotation schemes and software components for treebank development.

As discussed earlier, an improvement to the system that we are currently working on is
the semi-automatic creation of initial trees. The algorithm would automatically combine as many words and dependency links of the taggers' and the parsers' outputs as possible, and ask the annotator the make decisions on the rest. Such method would improve the quality of the initial trees, thus lowering the number of modifications needed to come up with the correct structure. Other future enhancements to the system could include even more strict and detailed checking algorithms for the annotated structures and an improved interface between DepAnn and the parsers which would allow the annotator to interact with the parsers in a case of problematic sentences. The approach has been successfully applied by some annotation tools, such as Annotate (Plaehen and Brants 2000) and the lexical analysis and constituency marking tools of the Alpino Treebank (van der Beek, Bouma, Malouf, and van Noord 2002). Often several annotators are working on the same sentences in order to ensure the consistency of the treebank. In such cases, it would be helpful if the tool would allow to manage multiple annotations and to perform inter-annotator agreement checks. Furthermore, the memory management of the tool could be improved in order to make it more efficient when working with large treebanks with tens of thousands of sentences.

## Acknowledgements

The author would like to thank the two anonymous reviewers for helpful comments on this paper and for some interesting suggestions for future work. The research reported in this paper has been supported by the European Union under a Marie Curie Host Fellowship for Early Stage Researcher Training at MULTILINGUA, University of Bergen, Norway, MirrorWolf project funded by the National Technology Agency of Finland (TEKES), and Automated Assessment Technologies for Free Text and Programming Assignments project funded by the Academy of Finland. The work was partly conducted while the author was working at the Human Language Technology Group at the Council for Scientific and Industrial Research (CSIR), Pretoria, South Africa and at the Faculty of Philosophy, University of Split, Croatia.

## Bibliography

Abeillé, A. (2003). Introduction. In A. Abeillé (Ed.), Treebanks: Building and Using Syntactically Annotated Corpora. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Bick, E. (2003). Arboretum, a hybrid treebank for Danish. In Proceedings of the 2nd Workshop on Treebanks and Linguistic Theories, Växjö, Sweden.

Böhmová, A., J. Hajič, E. Hajičová, and B. Hladká (2003). The Prague dependency treebank: Three-level annotation scenario. In A. Abeillé (Ed.), Treebanks: Building and Using Syntactically Annotated Corpora. Dordrecht, The Netherlands: Kluwer Academic Publishers.
Brants, S., S. Dipper, S. Hansen, W. Lezius, and G. Smith (2002). The Tiger Treebank. In Proceedings of the Workshop on Treebanks and Linguistic Theories, Sozopol, Bulgaria.

Daum, M., K. Foth, and W. Menzel (2004). Automatic transformation of phrase treebanks to dependency trees. In Proceedings of the 4 th International Conference on Language Resources and Evaluation, Lisbon, Portugal.
Demir, O., H. Keffer, V. Němčik, and S. B. Poggel (2006). TIGER API 1.8. - an interface to the TIGER corpus. http://www.tigerapi.org/ (Accessed April 21th, 2006).
Díaz de Ilarraza, A., A. Garmendia, and M. Oronoz (2004). Abar-hitz: An annotation tool for the basque dependency treebank. In Proceedings of the 4 th International Conference on Language Resources and Evaluation, Lisbon, Portugal.
Ide, N. and L. Romary (2003). Encoding syntactic annotation. In A. Abeillé (Ed.), Treebanks: Building and Using Syntactically Annotated Corpora. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Kakkonen, T. (2005). Dependency treebanks: Methods, annotation schemes and tools. In Proceedings of the 15 th Nordic Conference of Computational Linguistics, Joensuu, Finland, pp. 94-104.
Karlsson, F. (1990). Constraint grammar as a framework for parsing running text. In Proceedings of the 13th Conference on Computational Linguistics - Volume 3, Helsinki, Finland.
König, E., L. Wolfgang, and H. Voormann (2003). Tigersearch 2.1 user's manual. http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch (Accessed April 21th, 2006).

Kromann, M. T. (2003). The Danish dependency treebank and the DTAG treebank tool. In Proceedings of the 2nd Workshop on Treebanks and Linguistic Theories, Växjö, Sweden.

Kromann, M. T. (2004). Nordic treebank network tiger-xml: Proposals for extensions and conventions in tiger-xml within the nordic treebank network, septempber 1, 2004. http://www.id.cbs.dk/ ${ }^{\sim}$ mtk/ntn/tiger-xmlT.html (Accessed April 21th, 2006).
Lesmo, L., V. Lombardo, and C. Bosco (2002). Treebank development: the TUT approach. In Proceedings of the International Conference on Natural Language Processing, Mumbai, India.
Marcus, M., B. Santorini, and M. A. Marcinkiewicz (1993). Building a large annotated corpus of English: The Penn Treebank. Computational Linguistics 19(2), 313-330.
Mengel, A. and W. Lezius (2000). An XML-based representation format for syntactically annotated corpora. In Proceedings of the International Conference on Language Resources and Evaluation, Athens, Greece, pp. 121-126.
Plaehen, O. and T. Brants (2000). Annotate - an efficient interactive annotation tool. In Proceedings of the 6th Applied Natural Language Processing Conference, Seattle, Washington, USA.
Rambow, O., C. Creswell, R. Szekely, H. Taber, and M. Walker (2002). A dependency treebank for English. In Proceedings of the 3rd International Conference on Language Resources and Evaluation, Las Palmas, Gran Canaria, Spain.

Salvo, J. M. J. (2006). OpenJGraph - Java graph and graph drawing project. http: //openjgraph.sourceforge.net/ (Accessed January 18th, 2006).
Sampson, G. (1995). English for the Computer: The SUSANNE Corpus and Analytic Scheme. Oxford, UK: Oxford University Press.
Schneider, G. (1998). A Linguistic Comparison of Constituency, Dependency and Link Grammar. Zürich, Switzerland: Licentiate Thesis, University of Zürich.
Tapanainen, P. and T. Järvinen (1997). A non-projective dependency parser. In Proceedings of the 5th Conference on Applied Natural Language Processing, Washington, D.C., USA.

Tesniére, L. (1959). Éléments de Syntaxe Structurale. Paris, France: Klincksiek.
van der Beek, L., G. Bouma, R. Malouf, and G. van Noord (2002). The Alpino dependency treebank. In M. Theune, A. Nijholt, and H. Hondorp (Eds.), Computational Linguistics in the Netherlands 2001. Selected Papers from the Twelfth CLIN Meeting. Amsterdam, The Netherlands: Rodopi.
White, C. (2000). Rapid Grammar Development and Parsing: Constraint Dependency Grammar with Abstract Role Values. West Lafayette, Indiana, USA: PhD Thesis, Purdue University.

Xia, F. and M. Palmer (2000). Converting dependency structures to phrase structures. In Proceedings of the First International Conference on Human Language Technology Research, San Diego, California, USA.

# The Acquisition of Verb Semantics and Morphology in Italian as L2: Towards Computational Simulations 

Fabiana Rosi<br>Department of Linguistics, University of Pavia<br>rosi@unipv.it

## 1 Introduction

The study presents the preliminary computational simulations, which are carried out within a broader research (Rosi in preparation) about the acquisition of Tense-Aspect system in Second Language Acquisition. The research aims to contribute to the debate about the acquisition of Tense-Aspect system in Second Language Acquisition by comparing the development of Aspect morphology and verb semantics in Italian L2 between human learners and computational simulations as neural networks, namely Self-Organizing Maps (Kohonen 2001). The comparison intends to shed light on the cognitive principles and mechanisms that guide learners to acquire linguistic structures in a second language.

The literature on Tense-Aspect acquisition in First and Second Language, henceforth L1 and L2, (Andersen and Shirai 1994; Bardovi Harlig 2000) has documented the interaction between the morphological category of Aspect and the semantic category of Actionality in the development of the interlanguage of learners, who initially tend to restrict Aspect forms to specific Actionality classes and, then, gradually spread the Aspect markers following implicational developmental stages from most cognitively salient to least cognitively salient associations between Aspect and Actionality (fig.1.1). ${ }^{1}$

Recent surveys (Li and Shirai 2000; Li 2005) interpret the initial interaction between Aspect and Actionality in L1 Acquisition as the result of the children's analyses of the co-occurrence probabilities between morphological forms and semantic value of predicates in the linguistic input and verify this hypothesis in a connectionist model.

This study intends to test empirically if a neural network can display the acquisitional pattern of Aspect morphology and the interaction in this development between Aspect and Actionality also in L2 Acquisition, in order to help understanding, on the one hand, the

[^52]correlations between frequency effects and cognitive mechanisms, on the other, the role of the prior linguistic knowledge of L1 in L2 Acquisition.

## 2 Theoretical Background

### 2.1 The Acquisition of Italian Aspect Morphology

The category of Aspect expresses the point of view of speaker on the temporal phases of the event, that can be presented as completed and bounded, by means of perfective Aspect, or incomplete and ongoing, by means of imperfective Aspect, (Comrie 1976). Like other Romance languages, Italian verb system morphologically codifies the aspectual functions in the past, by means of two different past forms: perfective past (Passato Prossimo, Passato Remoto) and imperfective past (Imperfetto).

The category of Actionality encodes the inherent lexical-semantic information about the temporal phases of the event, ((Vendler 1957; Bertinetto 1986; Rothstein 2003). According to (Vendler 1957), on the basis of the semantic parameters of Durativity, Telicity and Dynamicity predicates are distinguished in four actional classes: States (ST), to be; Activities (ACT), to walk; Accomplishments (ACC), to grow up; Achievements (ACH), to die.

The studies on the acquisition of Italian as L1 and L2 (Antinucci and Miller 1976) for L1, (Giacalone Ramat 1995; Banfi and Bernini 2003) point out that perfective past is acquired before imperfective past, and that learners, at the beginning of acquisitional pattern, tend to associate perfective forms with Achievements and imperfective with States. Then, gradually, learners depart from these most cognitively congruent and salient associations and spread Aspect morphology to every predicates. The Aspect markers' diffusion is not casual, but it follows the implicational stages (fig.1.1) that emerge in acquisitional pattern of different L1 and L2, according with Aspect Hypothesis (Andersen and Shirai 1994).

Figure 1.1: The implicational stages of Aspect morphology acquisition.
Perfective: Achievements $>$ Accomplishments $>$ Activities $>$ States
Imperfective: States $>$ Activities $>$ Accomplishments $>$ Achievements

### 2.2 Self-Organizing Maps

Within the emergentist paradigm (MacWhinney 2001; Ellis 2003), recent studies (Li and Shirai 2000; Li 2005) interpret the interaction among the categories of Aspect and Actionality as the result of the learners' analysis of the probabilities of co-occurrences between Aspect morphology and Actionality semantics in the linguistic input. Children extract from the input the statistic frequencies of the combinations between Aspect forms and

Actionality classes. They, initially, strengthen the production of the most frequent associations, until the prolonged exposure to the input and the increasing account of data from input reduce the statistic difference between the most and the least frequent combinations.

Li \& Shirai (2000) test this hypothesis by means of computational simulations based on the Self-Organizing Maps (henceforth SOMs). The SOMs (Kohonen 2001) are unsupervised associative neural networks of "knots receptors" that classify input data translating relationships of similarity in topological relationships of proximity. Through an incremental exposure to an increasing account of data, the receptors are topologically organized on the network in such way that associated receptors have the tendency to recognize homogeneous classes of data. SOMs are biologically plausible models: human cerebral cortex can be conceived as essentially a multiple feature-map, where all neurons are initially coactivated and the associative strengths between neurons become more focused in parallel with the distributional increase of the corresponding co-occurrences in the input.

## 3 Method

### 3.1 Corpus

In order to test with computational simulations the acquisitional pattern found in literature (cfr 2.1.) and in recent researches on Italian L2 (Rosi 2006; Rosi in press), we collect an homogenous corpus of interlanguage and target language data that consists of the oral and written narratives of the same stories, produced by learners of Italian L2 and by Italian native speakers.

The interlanguage data are gathered by twenty-four Socrates students at University of Pisa, Italy, twelve German native speakers and twelve Spanish native speakers, who spend an year in Italy to learn Italian. They are requested to retell in Italian L2 and in their L1 three sequences of the silent film "Modern Times" by Charlie Chaplin, immediately after having watched them, each of them in a separate elicitation. The data collection is longitudinal, since the learners are interviewed three times, one every three months during the year spent in Italy, in order to gather information of three successive phases of the acquisitional development.

The Italian data are comprised of the oral and written narratives of the same three sequences of "Modern Times" produced in one interview by twenty-four Italian native speakers university students, as symmetric sample. These data provide a direct comparison between the native and non-native narrative production and are used as training corpus of the SOMs, so that the SOMs output analyses of the same data produced by the human learners. To these data will be added other corpora of Italian, both oral and written, in order to give the SOMs a significant representation of Italian language.

### 3.2 Procedure

As in the previous experiments that simulate by means of SOMs the acquisition of Aspect in correlation with Actionality (Li and Shirai 2000; Li 2005), the architecture of the SOMs is a multiple feature-map model (Mikkulainen 1997) connected by Hebbian learning principles (Hebb 1949). According to Hebb, the associative strength between two neurons is increased if both neurons are activated at the same time. In this model, two feature-maps are devoted to one specific type of linguistic information: one map is dedicated to semantic structures and classifies Actionality values, the other encodes morphological forms, the Aspect markers. Semantic receptors learn to form activated areas, or "bubbles", in correlation with the distributional behaviour of morphological inflections. This allows the SOMs to correlate semantic and morphological representations through the learning process, that develops by means of incremental training sets of Italian input, fed to the SOMs.

Morphological representations are inputted in the SOMs by selecting the two Italian past suffixes: both regular (for perfective aspect $-t o,-o$ ', for imperfective $-v a$ ) or irregular forms.

Semantic representations of predicates are more problematic. During the training, SOMs are expected to classify the predicates in most homogenous groups, which approach to the actional classes. In previous experiments, (Li and Shirai 2000; Li 2005), semantic representations consist in lexical co-occurrence analyses, extracted by the statistical frequencies of the co-occurrences between every predicates and the lexical items, which are attested within a predetermined window of words. We add to this methodology two innovative procedures, in order to notice the different results of everyone and to choose which model more appropriately accounts for the representation of such complex structure as verb semantics. The former consists in explicitly attributing semantic parameters relevant for actional classification, as definiteness of subject and object, to every predicate found in the input. The latter consists in calculating statistical frequency of co-occurrences between the predicate and specific elements of the context, as temporal expressions and negative form, hypothesized to influence the actional value of predicate (Rosi 2006). The second methodology aims to let emerge from distributional characteristic of data the semantic relations between lexical items, without an aprioristic supervision by researchers.

To observe effects of the interaction between semantic values and morphological forms in the learning pattern of the SOMs, we design different training sets and we input them incrementally, so that every stage includes new data and all previous stages.

After the training, the SOMs are requested to learn how to associate Aspect and Actionality like the Italian native use, following the same acquisitional pattern human learners go through. The assumption is that the strength of most frequent associations decreases, as more as the qualitative and quantitative account of data from input increases. In short, we expect that in first acquisitional phases the areas of morphological map activated for perfective form are co-activated with units of semantic map activated for Achievements, and the areas activated for imperfective form are co-activated with units activated for States. Then, since through the learning process the frequency of these associations decreases, the initial co-activations tend to relax gradually and also the less frequent associations
between Aspect and Actionality are acquired. Particular attention is paid to the dynamic evolution of the co-activations between morphological and semantic map during learning process, in order to enable a comparison with the interplay between Aspect morphology and Actionality semantics in the human learners' acquisitional development.

## 4 Human Learners' Acquisitional Development

First experiments consist in applying SOMs for modelling the learning curve of human learners, as basis for next simulations of SOMs' learning. In these experiments, SOMs are used in order to topologically represent the semantic similarity of predicates on the basis of statistical co-occurrences of predicates and Aspect morphological markers in actual production of learners and natives. These SOMs representations may be compared with the simulations that we are going to gather as output during the training of SOMs with a broader corpus of Italian standart.

In these preliminary experiments, SOMs have been fed with vectors which report the statistical distribution of co-occurrences between 40 predicates and the morphological marker of perfective, imperfective and present. The selected predicates are the 40 types predicates (about 10 for each Actionality class, see Appendix) that occur in the narration of every three scenes, so that is possible to compare the distribution with morphology of same predicates in different acquisitional stages. For that learners are requested to narrate in the three elicitations three different scenes, the learning space has to be considered on the comparison between natives and non natives' narration of the same scene, rather than on comparison among learners' production in each acquisitional stage.

The acquisitional pattern pointed out in literature (Andersen and Shirai 1994; Giacalone Ramat 1995) is confirmed since perfective and imperfective aspect gradually spread from most congruent Actionality classes to every predicates. Indeed, the statistical distribution of morphological forms with semantic classes is not constant through the three acquisitional stages we have analysed. At the end of the acquisitional process, learners approach much more natives' distribution of morphological markers through semantic classes. Furthermore, distribution of Aspect morphology through semantic classes doesn't evolve homogenously in Spanish-speakers and German-speakers learners.

It is significant that in SOMs representation of scene 1 (fig.1.2, fig.1.3, fig.1.4), that correspond to the initial acquisitional stage, telics (in italic) and atelics (in bold) predicates are grouped in same areas. This result displays that actually, at the beginning of learning, predicates belonging to same actional classes are marked with same morphological forms, according with the most prototypical associations predicted by Aspect Hypothesis.

The comparison between data of two samples of learners (fig.1.2 fig.1.3) underlines that Spanish-speakers rely on the actional value in selecting morphological forms, since telics and atelics are more concentrated in topological areas, whereas German-speakers' distribution is more blurred. A possible explication is that German-speakers tend to overextend perfective marker to every predicate, even in contexts where imperfective is more appropriate, because of influence of their L1, that codifies the past with only perfective forms (Preteritum and

Perfekt). In opposite, Spanish-speakers may be advantaged by their L1, because Spanish verb system morphologically encodes perfective and imperfective past, so learners are more competent to acquire aspectual functions of both morphological forms and to select the more appropriate Aspect for every predicate in different contexts. As example, notice that the State predicate potere, can, is included in atelic area in Spanish-speakers' and Italianspeakers' SOMs representation, in correlation of the high frequency of imperfective forms with state, but not in the German-speakers' one, since they don't associate imperfective Aspect neither to most congruent Actionality class, the States.

Therefore, it seems worth examining the SOMs representation of narration of scene 1in Spanish L1, in order to compare the use of morphological markers by Spanish-speakers in their L1 and in Italian L2. As shown in fig.5, predicate classification in Spanish L1 is quite similar to predicate representation in Italian L2 produced by Spanish-speakers and in Italian-speakers' narrative, e.g. the predicate poder, can, takes clearly part in atelics area.

In accordance with Aspect Hypothesis, during the acquisition, the most prototypical associations between Aspect and Actionality gradually decrease in parallel with the increase of the use of morphological forms with least congruent semantic class of predicates. SOMs representations of narration of the third scene ${ }^{2}$ (fig.1.6, fig.1.7, fig.1.8) confirm this acquisitional pattern both for Spanish-speakers and German-speakers, who approach more native use of Aspect markers than they do in first stage. It seems that German-speakers initially have to overcome a sort of bias determined by their L1, but, thanks to prolonged exposure to the input, they acquire both aspectual functions and start to select morphological forms with every actional type of predicate.

The SOMs representations carried out in first phases of the research demonstrate to be able to capture the different learning curve of German-speakers and Spanish-speakers learners of Italian. These initial experiments display that is possible to represent the acquisition of Aspect morphology by means of SOMs and encourage us to go on with the research. Additionally, the findings confirm that typological distance between L1 and L2 plays a role in acquisitional pattern and that the influence of L1 is stronger in first acquisitional phases. Consequently, results of application of SOMs representation to analysis of human learners' production provide several suggestions for application of SOMs to simulation of second language acquisition, as discussed in next paragraph.

## 5 Preliminary Conclusions and Future Work

On the basis of preliminary conclusions provided by first experiments we elaborate some procedures in order to address the challenging questions which arise in previous researches on computational simulations of First Language Acquisition (Li and Shirai 2000; Pirrelli et al. 2004; Dell'Orletta et al. 2005; Lenci et al. in press).

[^53]First of all, since we intend to simulate the acquisition of Italian as L2 rather than L1, we have to take into account the linguistic knowledge of the L1 that crucially distinguishes adult learners of a second language from children that are acquiring their mother tongue without an already formed linguistic competence. An attempt to deal with the linguistic representations provided by the L1 is to input in the SOMs the distributional biases hypothesized to be influenced by L1 verb system, such as the overextension of perfective forms found in modelling of German-speakers learners' acquisitional development. For simulating this bias, we propose to strengthen the connections' weights of the associative correlations between perfective bubbles and telics and atelics areas in semantic and morphological maps that simulate German-speakers' acquisition. This strengthening of connections' weights of perfective is reduced in advanced stages of training, in order to simulate the learning curve found in learners data and the assumption that L1 influence weakens through the acquisitional development of L2. An alternative method is to include in the training corpus of the SOMs also learners' L1 data, such as the German and Spanish narratives elicitated, eventually together with other German and Spanish corpora. With this procedure we try to simulate the acquisition of Italian L2 in SOMs which have already acquired L1.

Furthermore, an interesting question is whether gradual diversifications in the input distribution of morphological markers are a necessary requisite for the Tense-Aspect acquisition. Adult learners of L2 indeed, generally, are exposed from the beginning of learning to not simplified input, that is, in opposite, the usual input initially provided to children acquiring L1. We aim to test this hypothesis by feeding a corpus with constant distribution of Aspect inflections across actional classes into the SOMs, whereas Li \& Shirai (2000) use as training set the parental speech collected in CHILDES (MacWhinney 1995), where the distribution of morphological forms across predicates is incrementally diversified.

In addition, we compare the results of the different methodologies used to give the SOMs the semantic representations of predicates, in order to go beyond the distributional co-occurrence model of previous studies and to deepen the knowledge about the role played in Actionality classification by semantic features and by specific lexical elements in context.

## Acknowledgements

This research is supported by a PhD scholarship at University of Pavia, Italy. I'm very grateful to Alessandro Lenci for his constant and precious comments. I thank Boris Mangano for his kind help with Latex format. I would like to thank also all Socrates students and Italian native speakers who provide me linguistic data for my corpus and the Interdepartmental Linguistic Center of University of Pisa, Italy.

## Bibliography

Andersen, R.W. and Y. Shirai (1994) Discourse motivation for some cognitive acquisition principles. Studies in Second Language Acquisition 16, 133-156.
Antinucci, F. and R. Miller (1976). How children talk about what happened. Journal of Child Language 3, 167-189.

Banfi, E. and G. Bernini (2003). Il verbo Verso l'Italiano, Giacalone-Ramat, A., Roma, Carocci, 170-115.

Bardovi Harlig, K. (2000). Tense and Aspect in Second Language Acquisition: form, meaning and use. Oxford, Blackwell.
Bertinetto, P.M. (1986). Tempo, aspetto e azione nel verbo italiano. Il sistema dell'indicativo. Firenze, Accademia della Crusca.
Comrie, B. (1976). Aspect. Cambridge, Cambridge University Press.
Dell'Orletta, F., Lenci, A., Montemagni, S. and V., Pirrelli, (2005). Climbing the path to grammar: a maximum entropy model of subject/object learning. Proceedings of ACL-05, Psychocomputational Models of Human Language Acquisition, Michigan, Ann Arbor, 72-81.

Ellis, N.C. (2003). Constructions, Chunking and Connectionism : the Emergence of Second Language Structures. The Handbook of Second Language Acquisition, Doughty, C.J. and M.H. Long (eds), Oxford, Blackwell, 63-103.
Giacalone Ramat, A. (1995). Tense and Aspect in learner Italian. Temporal Reference. Aspect and Actionality, Bertinetto, P.M., Bianchi, V., Dahl, Ø. and M. Squartini (eds), Torino, Rosemberg\&Sellier, 289-307.
Hebb, D. (1949). The organization of behavior: A neuropsychological Theory. New York, N.Y., Wiley.

Kohonen, T. (2001). Self-Organizing Maps. 3 ediz., Springer\&Verlag.
Lenci, A., Montemagni, S. and V. Pirrelli (in press). Acquiring and Representing Meaning. Theoretical and Computational Perspectives. Acquisition and Representation of Word Meaning. Theoretical and Computational Perspectives., Lenci, A., Montemagni, S. and V. Pirrelli, Pisa/Roma, Istituti Editoriali e Poligrafici Internazionali, 19-66.

Li, P. (2005). Language acquisition in a self-organizing neural network model. Connectionist Models of Development., Quinlan, P. T. (ed) USA-Canada, Psychology Press, 115-149.

Li, P. and Y., Shirai (2000). The Acquisition of Lexical and Grammatical Aspect. Berlin, Mouton de Gruyter.

MacWhinney, B. (1995). The CHILDES project: Tools for Analizing Talk. Hillsdale, NJ., Erlbaum.

MacWhinney, B. (2001). Emergentist approaches to language. Frequency and the Emergence of Linguistic Structure., Bybee, J. and P., Hopper (eds), Amsterdam, Benjamins, 449-470.
Mikkulainen, R. (1997). Dyslexic and category specific aphasic impairments in a self-organizing feature map model of the lexicon. Brain and Language. 59

Pirrelli, V., Calderone, B., Herreros, I. and M., Virgilio, (2004). Non-locality all the way through: emergent global constraints in the Italian morphological lexicon. Proceedings of ACL/SIGPHON 2004.

Rosi, F. (2006). L'interazione fra aspetto e Aktionsart nelle interlingue di apprendenti avanzati di L2. Studi Italiani di Linguistica Teorica e Applicata. XXV 1
Rosi, F. (in press). Imparare il passato italiano in classe. Proceedings of VI AItLA Symposium.
Rosi, F. (in preparation). Aspect and Actionality in Italian L2: acquisitional strategies and computational simulations. PhD dissertation, University of Pavia, Italy.
Rothstein, S. (2003). Structuring events: a study in the semantics of lexical aspect. Oxford, Blackwell.

Vendler, Z. (1957). Verbs and times. Philosophical Review., 143-160.

## Appendix

Figure 1.2: SOMs representation of narration of scene 1 by German-speakers.

|  |  |  |  |  |  |  |  |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| portare <br> arrivare <br> dare <br> finire <br> liberare <br> potere <br> uscire <br> venire |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

Figure 1.3: SOMs representation of narration of scene 1 by Spanish-speakers.

| dare | camminare |  | fare |  | andare <br> chiamare |  |  |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
|  | guardare |  |  |  |  | avere | stare |
| credere <br> essere | sembrare |  |  |  |  |  | passare |
|  | volere | potere |  |  |  |  | portare |
|  |  |  |  |  | dire |  | cadere |$|$| parlare |
| :--- |

Figure 1.4: SOMs representation of narration of scene 1 by Italian-speakers.

| sembrare | volere |  | potere |  | sapere <br> dovere | andarevia <br> guardare <br> liberare <br> sentire |  |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| camminare |  | essere |  |  |  |  |  |
|  |  | esserci | credere |  | dire |  |  |
| lasciare | fare |  |  |  |  |  |  |
|  |  | pensare |  |  | cominciare |  | trovare |
| avere <br> parlare |  |  |  |  | passare |  | andare |
|  |  |  |  |  | arrivare | cadere |  |
| prendere |  |  |  |  |  |  |  |
| iniziare |  |  |  |  |  |  |  |$|$| arrestare |
| :--- |
| portare |

Figure 1.5: SOMs representation of narration of scene 1 in Spanish L1.

|  | créer |  | pensar |  | hacer <br> querer <br> caminar |  |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| dar |  |  |  |  |  | ser parecer |
| haber |  |  |  | estar |  | par |
|  |  |  | ir tener |  |  | poder hablar |
| pasar |  |  |  |  |  | \|levar |
|  |  |  |  |  |  |  |
| venir |  |  | salir |  |  | Irse Ilamar <br> tenerque mirar <br> iniciar dejar liberar <br> acordar <br> sentir |
|  |  |  |  |  |  |  |
| caer <br> Ilegar <br> poner <br> coger <br> suceder <br> acabar |  |  |  |  |  | saber |

Figure 1.6: SOMs representation of narration of scene 3 by German-speakers.

| camminare <br> credere <br> parlare <br> stare | esserci |  |  |  |  |  |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- |
|  | guardare |  |  |  |  | ricordarsi <br> sapere |
|  | volere | potere | avere |  | sembrare |  |
| dovere |  | essere |  |  |  | venire |
|  |  | chiamare |  |  | fare |  |
| cominciare |  | passare |  |  |  | arrivare <br> lasciare |
|  |  |  |  | prendere | andare |  |
| portare |  |  |  |  | andarevia <br> cadere <br> vedere |  |
|  |  |  |  |  |  |  |
| dire |  |  |  | finire <br> iniziare <br> liberare <br> mettere <br> sentire <br> succedere <br> uscire |  |  |

Figure 1.7: SOMs representation of narration of scene 3 by Spanish-speakers.

| arrestare <br> cadere <br> finire <br> iniziare <br> liberare <br> mettere <br> passare |  |  |  |  |  |  |  |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | sentire |  |
|  | arrivare |  | portare |  |  |  |  |
|  | vedere | dire |  |  | fare |  |  |
| fermare |  | prendere | chiamare |  | uscire | essere |  |
| lasciare <br> venire |  |  |  |  |  |  | succedere |
| cominciare |  |  | andare | trovare |  |  | guardare |
| pensare |  |  |  |  | camminare |  |  |
|  |  |  |  |  |  |  | volere |
|  |  |  |  | potere |  |  |  |
| credere <br> ricordare <br> sapere |  |  | avere |  | dovere | esserci | parlare <br> sembrare <br> stare |

Figure 1.8: SOMs representation of narration of scene 3 by Italian speakers.

| sembrare <br> sentire <br> ricordarsi |  |  | sapere |  | esserci |  | dovere <br> parlare <br> stare |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
|  |  | succedere |  |  | potere |  |  |
|  |  |  |  |  |  |  | avere <br> camminare |
| credere | pensare |  |  | dire |  |  | volere |
|  |  | cominciare |  | trovare |  | essere | passare |
|  |  |  |  |  |  |  |  |
| iiberare <br> mettere | iniziare | andare |  |  | chiamare |  | andarevia <br> guardare |
|  |  |  | fare | dare <br> arrivare |  |  |  |
| vedere |  | finire |  |  |  | venire |  |
| cadere | arrestare <br> uscire |  | fermare <br> prendere |  |  | portare |  |

Figure 1.9: Index of 40 verbs used in SOMs representations in Italian, English and Spanish, with actional value.

| Italian | English | Spanish | Actionality |
| :--- | :--- | :--- | :--- |
| andare | go | ir | ACT |
| andare via | go away | irse | ACH |
| arrestare | arrest | detener | ACC |
| arrivare | arrive | llegar | ACH |
| avere | have | tener | ST |
| cadere | fall | caer | ACH |
| camminare | walk | caminar | ACT |
| chiamare | call | Ilamar | ACT |
| cominciare | begin | empezar | ACH |
| credere | believe | créer | ACT |
| dare | give | dar | ACC |
| dire | say | decir | ACC |
| dovere | must | tenerque | ST |
| esserci | be there | haber | ST |
| essere | be | ser | ST |
| fare | do | hacer | ACT |
| fermare | stop | parar | ACC |
| finire | end | acabar | ACH |
| guardare | watch | mirar | ACT |
| iniziare | start | iniciar | ACH |
| lasciare | leave | dejar | ACC |


| Italian | English | Spanish | Actionality |
| :--- | :--- | :--- | :--- |
| liberare | free | liberar | ACC |
| mettere | put | poner | ACC |
| parlare | speak | hablar | ACT |
| passare | pass | pasar | ACC |
| pensare | think | pensar | ACT |
| portare | carry | Ilevar | ACT |
| potere | can | poder | ST |
| prendere | catch | coger | ACC |
| ricordarsi | remember | acordar | ST |
| sapere | know | saber | ST |
| sembrare | seem | parecer | ST |
| sentire | hear | sentir | ACT |
| stare | stay | estar | ST |
| succedere | happen | suceder | ACC |
| trovare | find | encontrar | ACH |
| uscire | go out | salir | ACH |
| vedere | see | ver | ACT |
| venire | come | venir | ACH |
| volere | want | querer | ST |
|  |  |  |  |

# What's the Difference between a Bat and a Mouse? A First Step towards Answering Comparison Questions in Open-domain QA 

Silke Scheible<br>School of Informatics, University of Edinburgh<br>S.Scheible@sms.ed.ac.uk


#### Abstract

In order to handle the question What's the difference between $X$ and $Y$ ? in open-domain Question Answering, a first crucial step is to understand which concepts are to be compared when X and $Y$ are polysemous. This paper reports on an investigation of the senses people choose to use in answering these questions and the way in which WordNet can be used in replicating their choices. Three approaches are tested and evaluated, based on the frequency, similarity or relatedness of the question terms. A relatedness measure proposed by Banerjee and Pedersen (2003) was found to achieve the highest accuracy.


## 1 Introduction

Comparisons have long been a central issue in academic areas such as philosophy, psychology or linguistics. The ability to see what makes objects, thoughts or words similar to each other is often considered to be the basis for human perception and cognition. In recent years there has been an increased interest in the computational aspects of comparisons, and various studies in the area of artificial intelligence have been put forward to formalise the intuitive notions of similarity and difference.

However, although comparison questions like What's the difference between $X$ and $Y$ ? or How are $X$ and $Y$ similar? are frequently asked ${ }^{1}$, they have not yet been addressed in open-domain question answering (QA). In closed-domain QA systems ground-breaking work has been done by McKeown (1985), but the task there is more straightforward due to a structured data model from which similarities and differences can be extracted.

In order to handle comparison questions in open-domain QA, the first important step is to understand what items are being compared. Unlike in closed-domain QA, where the database contains well-defined entities, even this first step is far from trivial, if an answer to a difference question is not explicitly given in the corpus. Consider for example:

[^54]What's the difference between a bat and a mouse?
A native speaker of English would answer this question by explaining the differences between a small nocturnal mammal and a small animal that belongs to the family of rodents. However, in the minimally different question

What's the difference between a bat and a racquet?
it is clear that the word bat now refers to a club used in ball games. While humans are generally very efficient at disambiguating polysemous words, this poses a major problem for computers. As Miller (1995) observes, "polysemy is a major barrier for many systems that accept natural language input". In particular, he notes that "in information retrieval, a query intended to elicit material relevant to one sense of a polysemous word may elicit unwanted material relevant to other senses of that word". Likewise, if we ask a QA system for the difference between a bat and a mouse, we do not want results that compare the club to a little grey rodent.

In order to answer difference questions automatically, it is therefore crucial to "make sense" of the question asked. This paper discusses both the way in which and the extent to which WordNet (Fellbaum 1998) can be used in identifying the same senses to be compared as people do in answering "What's the difference" questions.

## 2 Human Disambiguation of Difference Questions: An Experiment

To understand what question people take to be posed when asked a difference question involving ambiguous terms, I designed and carried out an experiment that involved the judgement of three human participants. Each of them was presented with a set of 100 "What's the difference between X and Y ?" questions, where X and Y were polysemous nouns listed in WordNet (version 2.0). ${ }^{2}$ Half of the questions were real user questions taken from question logs on the Internet, while the other half were generated to maximise polysemy. This ensured that there were both realistic questions on the one hand, but also questions with a high degree of polysemy. Each judge was then asked to assign exactly one sense to each of the question terms X and Y by choosing from the appropriate set of WordNet senses. The judges were further required to provide a confidence value for each decision between A ( $=$ confident) and D (= very unsure).

The judges agreed on what sense was intended in $82 \%$ of the questions, which shows that the task of disambiguating difference questions is no trivial task, even for humans. Judges 1 and 2 achieved an agreement of $86 \%$, while Judges 2 and 3 agreed on $84 \%$ of the questions. The best agreement was achieved by Judges 1 and 3 with $92 \%$. These values provide an upper limit for the task, since an automated approach should not be expected to outperform human annotators dealing with the same task. A closer look at

[^55]|  | word1 (\# senses) | word2 (\# senses) | J1 | J2 | J3 |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | disciple (1) | apostle (3) | 11 | 11 | 12 |
| 2 | sales (1) | marketing (3) | 11 | 12 | 12 |
| 3 | science (2) | technology (2) | 12 | 11 | 11 |
| 4 | consonance (2) | dissonance (3) | 23 | 22 | 23 |
| 5 | democrat (2) | republican (3) | 22 | 22 | 11 |
| 6 | legend (2) | fable (3) | 13 | 13 | 12 |
| 7 | porcelain (1) | glass (7) | 11 | 17 | 11 |
| 8 | original (2) | copy (4) | 12 | 22 | 12 |
| 9 | probation (3) | parole (3) | 33 | 23 | 33 |
| 10 | lease (3) | licence (3) | 23 | 33 | 23 |
| 11 | project (2) | product (6) | 12 | 13 | 12 |
| 12 | pea (3) | bean (4) | 11 | 21 | 11 |
| 13 | metre (3) | rhythm (5) | 24 | 31 | 24 |
| 14 | license (4) | certification (4) | 13 | 13 | 44 |
| 15 | pause (2) | break (15) | 17 | 11 | 17 |
| 16 | complement (6) | union (11) | 68 | 611 | 44 |
| 17 | rock (7) | stone (12) | 11 | 23 | 11 |
| 18 | death (8) | life (14) | 54 | 22 | 113 |

Table 1.1: Noun pairs disagreed upon by Judges J1, J2 and J3
the 18 questions where the judges disagreed shows that their average confidence level lies between B and C, while the average level of the 82 questions agreed upon is just slightly below A, indicating that disagreement is accompanied by lower confidence in the decision.

Table 1.1 displays the 18 noun pairs disagreed upon by the three judges, alongside with their number of senses in WordNet and the senses chosen by the individual judges. About half of them involve only a minor disagreement, where one of the judges labelled one of the question terms differently. The pairs legend/fable and probation/parole are a case in point. While Judges 1 and 2 chose sense 1 for legend and sense 3 for fable (both having the same gloss, a story about mythical or supernatural beings or events), Judge 3 agreed with sense 1 for legend, but chose sense 2 for fable (a short moral story (often with animal characters)). For parole, all judges chose the gloss ((law) a conditional release from imprisonment that entitiles [sic] the person to serve the remainder of the sentence outside the prison as long as the terms of release are complied with). However, for probation, Judges 1 and 3 picked ( (law) a way of dealing with offenders without imprisoning them; a defendant found guilty of a crime is released by the court without imprisonment subject to conditions imposed by the court; "probation is part of the sentencing process"), whereas Judge 2 thought it was (a trial period during which an offender has time to redeem himself or herself). It seems that the decision may sometimes be influenced by the way information is presented in WordNet's glosses, e.g. by a preceding domain
specification (as in "(law)"), or by identical glosses (as in the case of legend/fable).
There is only one noun pair in the question set where the judges disagree completely, which is life/death. This may be due to the philosophical nature of this question on the one hand, but it may also be a result of the large number of senses listed in WordNet for each of the words ( 8 for death, and 14 for life).

## 3 Modelling Human Difference Question Disambiguation

The question of interest is how a system could emulate the results described in the previous section. Intuitively, it is clear that the decisions made by the judges are not random. However, what technique humans actually employ to disambiguate difference questions is a psycholinguistic question, and may differ for different people in different situations. The question is whether it is possible to find a property that makes it possible for a system to imitate the human decision process. The following three alternatives are proposed, and will be discussed in turn:
I) Frequency - choose the most frequent sense of each question term
II) Similarity - choose the sense of each term that is most similar to the other
III) Relatedness - choose the sense of each term that is most closely related to the other

The first alternative is fairly straightforward. If the assumption in I) holds true, the frequency information provided in WordNet for each sense (derived from the sense-tagged SemCor corpus ${ }^{3}$ ) could be taken as an indicator for this method.

The second and third alternatives are more sophisticated. Unlike the first, they take into consideration that the two question terms are not disambiguated independently, but that the underlying process is one that could be described as co-disambiguation, where one question term is used to disambiguate the other (and vice versa). Option II suggests that this co-disambiguation is based on the similarity of the two question terms. This is motivated by the observation that humans often ask difference questions in order to distinguish between two concepts that they perceive as similar (Milosavljevic 1999). Generally, one could say that the more similar the question terms X and Y are, the more likely it is that people are confused about their differences. Therefore, option II suggests that for polysemous terms in difference questions, the intended senses are likely to be the ones that are the most similar. In order to imitate this process computationally, a similarity measure is required that can calculate a similarity score for any two given word senses. The software package WordNet::Similarity ${ }^{4}$ seems most suitable for this purpose, as it provides several such measures.

The third alternative is based on the relatedness of concepts, which is a notion often confused with similarity. However, it is a much broader concept: While similarity implies

[^56]relatedness, the inverse relationship does not hold true. Consider, for example, that toast and toasters are related, but by no means similar. The advantage of considering relatedness rather than similarity as underlying mechanism is that it may account for cases where the question terms are related rather than similar. This may happen when the person who poses the query is uncertain about the entities he or she is asking about. However, it is also possible that the broadness of the relatedness approach causes it to be less accurate than the similarity approach. In addition to the measures of similarity, WordNet::Similarity also provides relatedness measures, which will be used to choose the sense pairs that are most related.

## 4 Experiments and Results

### 4.1 Random Baseline

A first experiment was carried out to provide a baseline against which the results of the frequency, similarity and relatedness experiments can be compared. For this purpose, a random sense pair was generated for each difference question. An evaluation was then performed by comparing the results against the gold standard labelling of each of the three judges (in terms of accuracy). Questions were counted as correct only when both question terms were disambiguated correctly. The column labelled "Random" in Table 1.2 shows that the random baseline only achieves very poor results, $13 \%$ for Judge 2 and $14 \%$ for Judges 1 and 3.

### 4.2 The Frequency Approach

In order to test the frequency approach, senses were selected according to the frequency information provided in WordNet. What this amounts to is a so-called "first sense heuristic" (McCarthy et al. 2005), where the first listed sense is taken to be the most frequent sense, even when no frequency information is available (due to non-occurrence of the senses in the SemCor corpus). This is based on the general assumption that lexicographers list the primary senses of a word first. The chosen senses were then compared against the gold standard annotation of each of the judges (see column labelled "Frequency" in Table 1.2).

|  | Judges |  |  |  |  | Similarity |  | Relatedness |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | J1 | J2 | J3 | Random | Frequency | WUP | LIN | LESK |
| J1 | - | $86 \%$ | $92 \%$ | $14 \%$ | $48 \%$ | $65 \%$ | $67 \%$ | $76 \%$ |
| J2 | $86 \%$ | - | $84 \%$ | $13 \%$ | $46 \%$ | $58 \%$ | $59 \%$ | $68 \%$ |
| J3 | $92 \%$ | $84 \%$ | - | $14 \%$ | $48 \%$ | $65 \%$ | $64 \%$ | $74 \%$ |

Table 1.2: Results of experiment

The results are a considerable improvement to the random baseline results. However, compared to the results achieved by the human annotators (cf. columns labelled J1, J2 and J3 in Table 1.2), $46 \%-48 \%$ is quite poor. Treating Judge 1 as benchmark, the accuracies achieved by Judge 2 and Judge 3 are $86 \%$ and $92 \%$ respectively. Thus, the frequency method only reaches about $56 \%(48 / 86)$ of inter-human agreement.

### 4.3 The Similarity Approach

I then assessed the accuracy of the similarity approach to disambiguating difference questions, using WordNet::Similarity. ${ }^{5}$ Two measures, WUP (Wu and Palmer 1994) and LIN (Lin 1998), are representative of the alternative approaches towards measuring similarity in WordNet::Similarity: A path-based approach (WUP) and an information-content approach (LIN). WUP relies on path lengths between concepts in the WordNet hierarchy. It calculates the depth of the lowest common subsumer (lcs) of the concepts from the root of the hierarchy, and scales this value by the sum of the depths of the concepts themselves. In contrast, LIN uses the notion of information content (IC). The idea is that the similarity of two concepts can be measured by the extent to which they share information in common, which is indicated by the IC of their lowest common subsumer (and of the concepts themselves). In WordNet::Similarity, information content is by default derived from the sense-tagged SemCor corpus. However, there are options available that also allow using untagged corpora such as the BNC. For the present purpose the BNC-based approach turned out to be more suitable, as SemCor produced too many zero counts.

Given a question "What's the difference between X and Y?", I paired each sense of term X with each sense of term Y and calculated the similarity value of the pair according to both WUP and LIN. For both measures, the sense pair with the highest score was then selected as the result. Partial credit was given in case of a tie, i.e. where the top sense pairs received the same similarity score.

The results, displayed in the columns labelled WUP and LIN in Table 1.2, show that the similarity approach works much better than the frequency approach. The WUP and LIN measures perform almost equally well with $65 \%$ and $67 \%$ agreement with Judge 1, $58 \%$ and $59 \%$ with Judge 2, and $65 \%$ and $64 \%$ with Judge 3. The results of the measures are considerably worse when treating Judge 2 as the benchmark.

### 4.4 The Relatedness Approach

The relatedness approach was tested by applying the same procedure as described in the similarity approach, with the sense pair with the highest relatedness score selected as the winner. Of the three relatedness measures available in WordNet::Similarity, the Adapted LESK measure (Banerjee and Pedersen 2003) achieved the best results. In contrast to the similarity measures described above, LESK does not base its decisions on the structure

[^57]of the WordNet hierarchy or on information content, but on the glosses for each concept. It uses an algorithm for word sense disambiguation first proposed by Lesk (1986), based on gloss overlaps. While the original Lesk algorithm only considered the glosses of the concepts themselves, the extended gloss overlap measure (LESK) proposed by Banerjee and Pedersen (2003), also takes into account overlaps in glosses related to these words. Banerjee and Pedersen have furthermore implemented a scoring mechanism which takes into account the number of words that overlap, as opposed to assigning the same score to single and multiple word overlaps.

The last column in Table 1.2 shows that LESK outperforms both the frequency and the similarity measures, with $76 \%$ agreement with Judge 1, $68 \%$ with Judge 2, and $74 \%$ with Judge 3. The Lesk-Judge 1 agreement is thus only $8 \%$ worse than the lowest human agreement (between Judges 2 and 3; 84\%).

## 5 Discussion

The results in Table 1.2 show that all of the three techniques introduced in Section 3 perform substantially better than the random baseline. Here I discuss the results in more detail in order to show up the strengths and weaknesses of the three approaches.

### 5.1 Problems of the Frequency Approach

The frequency method assumes that for each question term, humans choose its most frequent sense, with frequency taken to correlate with either frequency in SemCor or just the order in which senses are listed in WordNet. However, while the most frequent sense may be the first sense that comes to mind when disambiguating a question term, it is bound to be rejected if inappropriate to the context. Here, the other term in the question appears to be sufficient context for rejecting the most frequent sense as inappropriate. The poor results in the experiment support the conclusion that the frequency method is too rigid a measure to be used for disambiguating difference questions. However, while interpreting these results one has to bear in mind that half of the question set was designed to maximise polysemy. It is possible that the selection of these items was biased towards low-frequency senses of the question terms, while in naturally occurring data most words have only one commonly occurring sense. In order to see whether the results described in this section under-represent the true performance of the frequency approach, I tested it on the subset of real user questions in the original test set ( 50 questions altogether). Treating Judge 1 as a benchmark, the accuracy achieved by the frequency approach is $44 \%$, i.e. $4 \%$ worse than on the full set. It is therefore safe to conclude that the frequency approach is not suitable for the purpose of disambiguating difference questions.

### 5.2 Problems of the Similarity and Relatedness Approaches

Compared to the frequency method, both the similarity and the relatedness approaches produce better results and may be adequate for identifying the same disambiguated senses reached by human annotators. In order to investigate the performance of the measures in more detail, I analysed the noun pairs where the judges agreed ( 82 pairs), but WUP, LIN and LESK produced different results. ${ }^{6}$ The analysis revealed several interesting issues, which I discuss below.

In WordNet, highly similar concepts are often included in the same synonym set (synset). As a consequence, both the similarity and the relatedness measures will almost always choose these senses: The path-based measure WUP chooses them because they are connected by the shortest path, the IC-based measure LIN because the lcs is a direct ancestor to both (and the concepts have the same IC value). The gloss-based measure LESK chooses them because they have identical glosses. This causes two different problems. First, as soon as the words share senses that occur in the same synset, results are inevitably wrong if these are not the intended ones. The most striking error in the question set results from this kind of problem. Atom and molecule are both included in the synset with the gloss ((nontechnical usage) a tiny piece of anything), whereas the intended senses are individual "physics and chemistry" glosses. But even when the senses included in the same synset are the intended ones, there is a second problem. Given that the purpose of disambiguating difference questions is to distinguish two similar concepts, the next step in answering difference questions requires identifying some differences between them. As this is only possible if the intended senses belong to different synsets, such questions cannot be answered by using WordNet alone as a knowledge source.

While one could argue that the previous problem is down to WordNet not being finegrained enough, researchers have also complained about the opposite: Often WordNet senses are too fine-grained, and even human judges have difficulty assigning the correct senses. As discussed above, the annotators who provided the gold standard for the test set could only agree on 82 out of 100 questions. The noun pair pea/bean is a case in point. According to Judges 1 and 3, the correct senses are 1. pea -- (seed of a pea plant) and 1. edible bean -- (any of various edible seeds of plants of the family Leguminosae). However, this is debatable, since the sense pair 2. pea -(the fruit or seed of a pea plant) and 2. bean -- (any of various seeds or fruits suggestive of beans) appears very similar to the first sense pair. An answer based on this incorrect pair would probably be as good as that based on the first one. (This will be part of a later investigation).

While the problems discussed so far affect both the similarity and the relatedness approach, some are particular to similarity. One problem lies in the hierarchical structure of WordNet: Although similar, some senses may only have a dummy root node as their lowest common subsumer. This affects both WUP and the LIN, resulting in similarity scores of 0 for the senses concerned. The resulting mistakes can be quite grave, since distant senses

[^58]that have an lcs may receive a higher score than the correct senses. This problem does not affect LESK, as it is purely gloss-based, which is reflected in the fact that it performs much better at disambiguating noun pairs such as science/art, enlightenment/romanticism and temperature / heat.

Another problem that affects the similarity measures only concerns the network density of WordNet, which is often reported as a problem. Jiang and Conrath (1997) note that "it can be observed that the densities in different parts of the hierarchy are higher than others", such as in the plant/flora section. A consequence of this is that decisions based on the closeness of concepts can become highly unreliable once high-density areas are involved. Consider for example the pair pork/pig, where pig is disambiguated incorrectly by both WUP and LIN. The problem here is that pig is in a very dense area in WordNet, and has the hypernyms => swine => even-toed ungulate => ungulate => placental mammal => mammal => vertebrate, craniate => chordate => animal => ... Because of this, both the shortest path and the IC methods fail. For pig, WUP chooses sense 6, (a crude block of metal (lead or iron) poured from a smelting furnace), while LIN selects sense 4, (bull, cop, copper, fuzz, pig -- uncomplimentary terms for a policeman). This latter sense of pig happens to have a higher IC in the BNC (and also in SemCor).

The main weakness of the relatedness approach is that LESK depends heavily on the format of the glosses, which are subject to idiosyncratic decisions made by WordNet lexicographers. This is illustrated by the following glosses of the pair mug/cup: (mug -- with handle and usually cylindrical) and (cup -- a small open container usually used for drinking; usually has a handle; 'he put the cup back in the saucer"; 'the handle of the cup was missing"). Had the lexicographer working on the gloss of mug considered the format of the gloss for cup, the LESK measure would most likely have produced the right result.

It appears that most problems are due to the hierarchical structure of WordNet, and consequently mainly affect the similarity measures. We can conclude that the best contribution that WordNet can make to disambiguating difference questions is by the relatedness approach. But $76 \%$ accuracy is not sufficient for WordNet relatedness to be used on its own.

## 6 Conclusion and Future Work

This paper has investigated how questions of the form What's the difference between $X$ and $Y$ ? can be disambiguated automatically as a first step towards answering comparison questions in open-domain QA. The goal was to find an automatic way of identifying how humans disambiguate such questions. Three approaches were proposed, based on frequency, similarity and relatedness. The relatedness approach produced the best results, thus suggesting that humans choose the senses that are the most related when they process difference questions, and not the most frequent or similar ones. However, while the frequency approach can easily be dismissed as unsuitable, an analysis of the data revealed
that the problem with the similarity approach may not lie in the theoretical idea itself, but in the fact that WordNet is not an adequate representation of similarity. Two concepts can be similar in many different ways, and often this is not captured in a taxonomy such as WordNet. Milosavljevic (1999) observes that "although echidnas, porcupines and hedgehogs are extremely similar in appearance, they are not closely related under the Linnaean taxonomy [...], because appearance is not an attribute which is used to classify animals under this system". We can conclude that similarity may well be the underlying mechanism, but that the similarity leading to the question is not one that is captured in WordNet. The broader notion of relatedness is better suited for taxonomies such as WordNet, as it also captures sense pairs that are not classified as similar within the hierarchy.

Future work includes investigating alternative possibilities for this task. As the individual results obtained for measures such as LIN and LESK are quite different, it may be of interest to investigate whether polling would produce improved results. I also want to consider whether there are more suitable knowledge bases available, since the disambiguation of the question terms is only the first step towards an implementation of a QA system that provides satisfying answers to difference questions. The descriptions of concepts in WordNet glosses are often too short to construct a satisfying answer. A possible alternative is an encyclopedia such as the Wikipedia. ${ }^{7}$ Its advantages over WordNet are not only longer definitions, but also a broader coverage of concepts. Its possible disadvantage is, of course, the significant variability in its entries. However, in initial analysis, the disambiguation pages provided for polysemous entries look particularly promising. Considering that the Lesk algorithm is independent from the use in WordNet, its application to Wikipedia entries constitutes an interesting experiment for future work.

[^59]
## Bibliography

Banerjee, S. and T. Pedersen (2003). Extended gloss overlaps as a measure of semantic relatedness. In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence.

Fellbaum, C. (Ed.) (1998). WordNet: An electronic lexical database. MIT Press, Cambridge, MA.
Jiang, J. J. and D. W. Conrath (1997). Semantic similarity based on corpus statistics and lexical taxonomy. In Proceedings of International Conference on Research in Computational Linguistics.
Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from a ice cream cone. In Proceedings of SIGDOC ' 86.
Lin, D. (1998). An information-theoretic definition of similarity. In Proceedings of the 15th International Conference on Machine Learning.
McCarthy, D., R. Koeling, and J. Weeds (2005). Ranking WordNet senses automatically. Technical report, Department of Informatics, University of Sussex.

McKeown, K. R. (1985). Text generation: using discourse strategies and focus constraints to generate natural language text. New York: CUP.
Miller, G. (1995). WordNet: A lexical database for English. Communications of the ACM 38(11).
Milosavljevic, M. (1999). The automatic generation of comparisons in descriptions of entities. Ph. D. thesis, Microsoft Research Institute, Macquarie University, Sydney, Australia.

Pedersen, T., S. Banerjee, and S. Patwardhan (2005). Maximizing semantic relatedness to perform word sense disambiguation. Technical report, University of Minnesota Supercomputing Institute Research Report UMSI 2005/25.

Wu, Z. and M. Palmer (1994). Verb semantics and lexical selection. 32nd Annual Meeting of the Association for Computational Linguistics.

# Automatic Verb Valency Frames Disambiguation for Czech 

Jiří Semecký<br>Institute of Formal and Applied Linguistics, Charles University, Prague, Czech Republic<br>semecky@ufal.mff.cuni.cz


#### Abstract

This paper deals with automatic disambiguation of verb valency frames on Czech data. Main contribution lies in determining of the most useful features for valency frame disambiguation. We experimented with diverse types of features, including morphological, syntax-based, idiomatic, animacy and WordNet-based. The considered features were classified using decision trees, rule-based learning and Naïve Bayes classifier.

On a set of 7778 sentences we achieved accuracy of $79.86 \%$ against baseline $68.27 \%$ obtained by assigning the most frequent frame.


## 1 Introduction

Many recent NLP applications, including machine translation, information retrieval, and others, aiming at higher quality results need semantic analysis of language data on the sentence level. As verbs are understood as central elements of sentences, the key aspect in determination of the sentence meaning is estimation of meaning of the verb. Valency frames of verbs usually partially correspond to their meanings.

Choosing the appropriate verb frame with respect to a given frames definition could be described as a special case of word sense disambiguation. First results of verb frame disambiguation were already reported by (Erk 2005) for German and (Lopatková, Bojar, Semecký, Benešová, and Žabokrtský 2005) for Czech.

For our task we used VALEVAL (Bojar, Semecký, and Benešová 2005), a human annotated corpus of valency frames containing data selected from the Czech National Corpus (Kocek, Kopřivová, and Kučera 2000). VALEVAL contains frames assigned according to definitions in the VALLEX lexicon (Žabokrtský and Lopatková 2004).

We generated a vector of features describing the contexts of a verb for each verb in our dataset. Later, we trained machine learning methods on a part of the data, and tested it on the rest. For lack of data, we employed 10 -fold cross-validation.

We used three different methods, Naïve Bayes classifier, decision trees and rule-based learning. We tested five different types of features describing verb occurrences based on a context within one sentence.

This paper is divided as follows: in Section 2, we give an overview of data which we worked with, in Section 3 we describe methods which we employed in the frame disambiguation and features which we used for describing verbs in their context. In Section 4, we evaluate our results using two different metrics. In the last section, we conclude and suggest further development.

## 2 Data Resources

### 2.1 Valency Lexicon

For automatic assignment of valency frames we need a valency lexicon consisting of formal definitions of frames. In our experiments we used VALLEX, a manually created valency lexicon of Czech verbs, which is based on the framework of Functional Generative Description (FGD) (Sgall, Hajičová, and Panevová 1986).

VALLEX is being built since 2001 and the work is still in progress. The VALLEX version 1.0 (autumn 2003), which we used in our task, defines valency for over 1,400 Czech verbs and contains over 3,800 frames. 6000 valency frames.

The VALLEX lexicon consists of verb entries corresponding to particular verb lexemes, i.e. complex units consisting of the verb base lemma and its possible reflexive particle se or si. For example, the verb lexeme dodat si consists of a base lemma dodat and a reflexive particle si. There is also the verb dodat with no reflexive particle, which has other meaning.

Each verb entry consists of definitions of one or more frames, which roughly correspond to meanings of the verb. The average number of frames per verb lexeme in VALLEX is 2.7 and the average number of frames per base lemma is 3.9 .

Each valency frame consists of a set of frame slots corresponding to complements of the verb. Each frame slot is described by functor, expressing the type of relation between the verb and the complement (e.g. Actor, Patient, Addressee), list of possible morphological forms in which the frame slot might be expressed, and type of the slot (obligatory, optional or typical).

Moreover, each frame in the lexicon is accompanied by an explanation of the meaning (using synonyms or glosses), an example sentence or phrase, and its aspectual counterpart if it exists. Some frames are assigned to semantic classes. A frame could also be marked as "idiom" if it is used idiomatically.

Figure 1.1 shows an example of a VALLEX entry for the verb lexeme dodat, containing five frames for its different senses, namely supply, ship, mention, add, and encourage. Each frame is described by list of frame slots (e.g. ACT, ADDR, PAT, DIR for the first frame). The superscript specify the type of the slot, and the subsript represents its surface representation (the preposition, if applicable, and the case).

```
dodat pf.
1) dodat }\mp@subsup{}{1}{}\approx\mathrm{ dopravit
-frame: ACT [ 
-example: dodat někomu zboží do domu
-asp.counterparts: dodávat }\mp@subsup{}{1}{}\mathrm{ impf.
-class: transport / exchange
2) dodat }\mp@subsup{2}{2}{}\approx\mathrm{ dopravit
-frame: ACT }\mp@subsup{1}{1}{obl}\mp@subsup{\mathbf{PAT}}{4}{obl}\uparrow\mathbf{DIR3}\mp@subsup{}{}{obl}\mp@subsup{\mathbf{BEN}}{3,pro+4}{typ
-example: dodat někomu / pro někoho do domu zboží
-asp.counterparts: dodávat }\mp@subsup{2}{2}{}\mathrm{ impf.
-class: transport
3) \mp@subsup{dodat }{3}{}\approx\mathrm{ říci; podotknout}
-frame: ACT ol
-example: dodal k tomu své připomínky / vše, co věděl
-asp.counterparts: dodávat }\mp@subsup{}{3}{}\mathrm{ impf.
-class: communication
4) dodat }\mp@subsup{4}{4}{}\approx\mathrm{ doplnit; připojit
-frame: ACT }\mp@subsup{|}{1}{obl}\mp@subsup{\mathbf{PAT}}{4}{obl}\mp@subsup{\mathbf{EFF}}{k+3}{obl
-example: dodal ke starému zbožá nové
-asp.counterparts: dodávat }\mp@subsup{4}{4}{}\mathrm{ impf.
-class: combining
5 dodat }\mp@subsup{}{5}{}\approx\mathrm{ povzbudit (idiom)
-frame: ACT [1 \aDDR }\mp@subsup{}{3}{obl}\mp@subsup{\mathbf{PAT}}{2,4}{obl
-example: dodat někomu odvahy / odvahu
-asp.counterparts: dodávat }\mp@subsup{}{5}{}\mathrm{ impf.
-class: exchange
```

Figure 1.1: Example of VALLEX entry for verb lexeme dodat (meanings: supply, ship, mention, add, and encourage).

### 2.2 Training and Testing Data

For training and testing of disambiguation methods, we need data annotated according to the chosen frame definitions. There is a manually annotated corpus of frame annotations VALEVAL (Bojar, Semecký, and Benešová 2005) developed as a lexical sampling experiment using VALLEX frame definitions. It contains 109 selected base lemmas. For each base lemma, 100 sentences from the Czech National Corpus ${ }^{1}$ (Kocek, Kopřivová, and Kučera 2000) were randomly selected.

[^60]For purpose of the VALEVAL corpus, reflexivity of verbs (expressed by a separate reflexive particle) was disregarded, as there is no automatic procedure to determine it. For all verbs selected to be present in the VALEVAL, their aspectual counterparts including iterative forms were added too. In order to cover both "easy" and "difficult" cases, verbs were selected randomly from both ends of the difficulty spectrum. Moreover, some verbs were added on purpose to cover specific cases too.

The VALEVAL was concurrently annotated by three annotators looking at the sentence containing the verb and three preceding sentences. Annotators had also the option of selecting no frame if the corresponding frame was missing or if the decision could not been done due to wrong morphological analysis. The inter-annotator agreement of all three annotators was $66.8 \%$, the average pairwise match was $74.8 \%$.

### 2.3 Data Preparation

As for input data for the frame disambiguation task, we used VALEVAL sentences where all three annotators agreed. Moreover, sentences on which annotators did not agree were rechecked by another annotator, and sentences with a clear mistake were corrected and added too. This resulted in a set of 8066 sentences.

Then, we automatically parsed the sentences using Charniak's syntactic parser (Charniak 2000), which was trained on the Prague Dependency Treebank (Hajič 1998). Some sentences could not have been parsed because of their length (the corpus contains sentences from fiction with length over 400 words). After excluding unparsed sentences, 7778 sentences remained, which served as input for disambiguation methods. There were 72.0 sentences per base lemma in average, ranging from a single sentence to 100 sentences (the original amount in the VALEVAL). Figure 1.2 shows the distribution of number of sentences per base lemma.


Figure 1.2: Distribution of the number of sentences per base lemma

## 3 Method

### 3.1 Machine Learning Methods

For automatic frame disambiguation, we generated a vector of features for each instance of a verb. A detailed description of these vectors is given in Section 3.2.

Later, we trained machine learning methods for each verb separately on a part of the data, and tested it on the rest. Due to lack of annotated data, we employed 10 -fold crossvalidation: we divided the data into 10 parts, for each tenth we trained the algorithm on the remaining data and tested it on the selected tenth. Finally, we counted the accuracy as the average of accuracies over the ten runs.

We tested three different classification methods, namely Naïve Bayes classifier, decision trees and rule-based learning, the later two implemented in the the machine learning toolkit C5.0 (Quinlan 2005).

Naïve Bayes classifier computes the probability that an instance belongs to a given class separately for each feature and computes the overall probability as if the features were independent.

The decision trees algorithm finds the most discriminative feature, i.e. the one that suits best for dividing the training data into two parts belonging to different classes. After the first decision, the process continues recursively in all branches resulting in a tree of decisions which indicates the features to use for division of the feature space, i.e. a decision tree.

The ruleset algorithm creates a set of independent rules defined as a conjunction of conditions for feature values. Conditions of individual rules may overlap, in which case the rules' predictions are aggregated using their confidence (proposed by the algorithm) to reach a verdict.

Decision trees and the rulesets are equally expressive.

### 3.2 Feature Selection

We experimented with several types of features containing different information about the context of the verb within one sentence. The following list describes five different types of features we used.

- Morphological: purely morphological information about lemmas in a small window centered around the verb.
- Syntax-based: information resulting from the result of an automatic syntactic parser (including mainly morphological and lexicographical characteristics).
- Idiomatic: occurrence of idiomatic expressions in the sentence according to the VALLEX lexicon.
- Animacy: information about animacy of nouns and pronouns both dependent on the verb and occurring anywhere in the sentence.

| Feature type | \#Features | \#Used features | Relative weight |
| :--- | :---: | :---: | :---: |
| Morphological | 60 | 21 | $24.28 \%$ |
| Syntax-based | 103 | 22 | $58.40 \%$ |
| Idiomatic | 118 | 1 | $0.82 \%$ |
| Animacy | 14 | 9 | $5.76 \%$ |
| WordNet | 128 | 25 | $10.74 \%$ |
| Total | $\mathbf{4 2 3}$ | $\mathbf{7 8}$ | $\mathbf{1 0 0 . 0 0 \%}$ |

The column "\#Used features" indicates the number of features used in the decision trees. The column "Relative weight" indicates the weight based on the feature occurrences in the decision trees.

Table 1.1: Types of features.

- WordNet: information based on the WordNet top-ontology classes of the lemmas both dependent on the verb and occurring anywhere in the sentence.

The first two columns of Table 1.1 show the number of features belonging to each of the groups. In the following section we give a detailed description of each group of the features.

## Morphological Features

Czech positional morphology (Hajič 2000) uses morphological tags consisting of 12 actively used positions, each stating value of one morphological category. The morphological categories are: part of speech, detailed part of speech, gender, number, case, possessor's gender, possessor's number, person, tense, grade, negation and voice. Categories which are not relevant for a given lemma (e.g. tense for nouns) are assigned a special value.

For lemmas within a five-word window centered around the verb (two preceding lemmas, the verb itself, and two following lemmas) we used each position as a single feature. Hence we obtained 60 morphological features ( 5 lemmas, 12 features for each).

## Syntax-based Features

Based on the result of an automatic syntactic parser we extracted the following features:

- Two boolean features stating whether there is a pronoun se or si dependent on the verb.
- One boolean feature stating whether the verb depends on another verb.
- One boolean feature stating whether there is a subordinate verb dependent on the verb.
- Six boolean features, one for each subordinating conjunction defined in the VALLEX lexicon ( $a b y, a \check{t}, a \check{z}, j a k$, $\check{z} e$ and $z d a$ ), stating whether this subordinating conjunction occurs dependently on the verb.
- Seven boolean features, one for each case, stating whether there is a noun or a substantive pronoun in the given case directly dependent on the verb.
- Seven boolean features, one for each case, stating whether there is an adjective or an adjective pronoun in the given case directly dependent on the verb.
- Three boolean features, one for each degree of comparison (positive, comparative, superlative), stating whether there is a lemma in the given degree directly dependent on the verb.
- Seven boolean features, one for each case, stating whether there is a prepositional phrase in this case dependent on the verb.
- 69 boolean features, one for each possible combination of preposition and case, stating whether there is the given preposition in the given case directly dependent on the verb.

Together, we used 103 syntax-based features.

## Idiomatic Features

We extracted a single boolean feature for each idiomatic expression defined in the VALLEX lexicon. We set the value of the corresponding feature to true if all words of the idiomatic expression occurred anywhere in the sentence contiguously. Features corresponding to not occurring idiomatic constructions were set to false.

Together, we obtained 118 idiomatic features.


#### Abstract

Animacy We partially determined animacy of nouns and pronouns in the whole sentence. Then, we introduced seven boolean features, one for each case, stating whether there is an animate noun or pronoun in this case syntactically dependent on the verb, and one integer feature stating the number of animate nouns and pronouns dependent on the verb. Moreover, we introduced another seven boolean features, one for each case, stating whether there is an animate noun or pronoun in this case anywhere in the sentence, and one integer feature stating the number of animate nouns and pronouns in the sentence. The later features can operate even in case of wrong result of syntactic parser. In cases where we could not decide, we set the feature to false.

Together we obtained 14 features for animacy. We determined the animacy using several techniques. As for nouns, the Czech lemmatizer created by Jan Hajič (Hajič 2000) gives additional information about some lemmas. These include among others identification of first names


and surnames. In cases where the lemmatizer marked a lemma as a name we set the animacy to true. We also used the fact that the morphological category gender distinguishes between masculine animate and masculine inanimate in some cases, as the masculines behave differently for animate and inanimate nouns. However, for common feminine and neutrum nouns we could not determine the animacy.

As for pronouns, the morphological category detailed part of speech gives us information about the type of the pronoun. Some types of pronoun imply animacy. Again, not all cases can be determined in this way.

### 3.3 WordNet Features

In some cases, dependency of a certain lemma or a certain type of lemma on a verb can imply its particular sense. However, as the machine learning methods which we used work with a fixed number of features, we could not have added information about individual lemmas easily. We described a lemma type in terms of belonging to WordNet (Fellbaum 1998) classes instead.

In the first step, we used the definition of WordNet top ontology made at University of Amsterdam (Vossen, Bloksma, Rodriguez, Climent, Calzolari, Roventini, Bertagna, Alonge, and Peters 1997) to obtain a tree-based hierarchy of 64 classes.

Then, for each lemma present in the definition of the top ontology, we used the WordNet Inter-Lingual-Index to map English lemmas to the Czech EuroWordNet (Pala and Smrž 2004), extracting all Czech lemmas belonging to the top level classes. After this step we ended up with 1564 Czech lemmas associated to the WordNet top-level classes. As we worked with lemmas, and not with synsets, one lemma could have been mapped to more top-level classes. Moreover, if a lemma is mapped to a class, it belongs also to all the predecessors of the class.

In the second step, we used the relation of hyperonymy in the Czech WordNet to determine the top-level class for other nouns as well. We followed the relation of hyperonymy transitively until we reached a lemma assigned in the first step. Again, as we worked with the lemmas instead of synsets, one lemma could have been mapped to more top-level classes.

For each top level class we created one feature telling whether a noun belonging to this class is directly dependent on the verb, and one feature telling whether such noun is present anywhere in the sentence.

This resulted into 128 WordNet class features.

## 4 Results

### 4.1 Baseline for Frame Disambiguation

As a baseline for each base lemma we took the relative frequency of its most frequent frame using 10 -fold cross validation. The baselines ranged from $24 \%$ (for base lemma vzít with

10 different annotated frames) to $100 \%$ for verbs with only one frame. Figure 1.3 shows distribution of the relative frequency of the most frequent frames.


Figure 1.3: Distribution of the relative frequency of the most frequent frames

We computed the overall baseline as weighted average of the individual baselines. The overall baseline was $68.27 \%$ when weighting by the number of sentences in our dataset and $60.64 \%$ when weighting by the relative frequency in the Czech National Corpus. The second one better predict behaviour on real data. The difficulty of the task can be seen in the Table 1.2.

### 4.2 Evaluation

In our experiments we tested performance of automatic disambiguation classifiers based on each presented type of features separately, as well as on different combinations of feature types. Then, based on the acquired decision trees, we observed which features were most frequently used for the decisions.

Table 1.3 states accuracy of the word sense disambiguation task for different combinations of features. Columns corespond to different disambiguation methods - Naïve Bayes classifier (NBC), decision trees (DT), and rule-based learning (RBL). The symbol $\emptyset_{\text {data }}$ indicates the average accuracy weighted by the number of sentences in the input data, whereas the symbol $\emptyset_{C N C}$ indicates the average accuracy weighted by the relative frequency in the Czech National Corpus (CNC).

|  | $\oslash_{\text {data }}$ | $\oslash_{C N C}$ |
| :--- | :---: | :---: |
| Average number of frames | 4.58 | 5.61 |
| 10-fold baseline | 68.27 | 60.64 |

$\oslash_{\text {data }}$ denotes average weighted by the number of sentences in the dataset. $\oslash_{C N C}$ denotes average weighted by the number of sentences in the Czech National Corpus.

Table 1.2: Difficulty of the frame disambiguation task

|  | $\oslash_{\text {data }}$ |  |  |  | $\oslash_{C N C}$ |  |  |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Type of features | NBC | DT | RBL | NBC | DT | RBT |  |
| Morphological | 71.88 | 73.83 | 74.25 | 62.06 | 66.26 | 65.33 |  |
| Syntax-based | 77.05 | 78.33 | 78.23 | 70.46 | 70.65 | 70.77 |  |
| Idiomatic | 68.31 | 68.37 | 68.31 | 60.97 | 60.93 | 60.73 |  |
| Animacy | 65.89 | 70.77 | 70.76 | 52.84 | 62.58 | 62.46 |  |
| WordNet | 63.01 | 70.64 | 70.59 | 45.4 | 60.21 | 60.04 |  |
| M + S | 73.51 | 78.9 | 78.7 | 63.98 | 69.48 | 68.97 |  |
| M + W | 72.69 | 73.85 | 73.9 | 62.08 | 66.07 | 66.47 |  |
| S + A | 73.51 | 78.58 | 78.48 | 63.51 | 70.69 | 71.19 |  |
| S + I | 77.14 | 78.29 | 78.32 | 69.87 | 70.69 | 71.06 |  |
| S + W | 73.8 | 78.49 | 78.86 | 59.87 | 71.15 | 71.28 |  |
| M + S + A | 74.52 | 78.76 | 79.22 | 63.5 | 69.77 | 68.63 |  |
| M + S + I | 73.48 | 78.8 | 78.86 | 63.99 | 68.74 | 69.2 |  |
| M + S + W | 74.32 | 79.16 | 79.47 | 64.94 | 77.25 | 77.41 |  |
| M + A + I | 72.76 | 74.61 | 74.88 | 61.75 | 63.52 | 64.35 |  |
| M + A + W | 73.23 | 74.23 | 74.29 | 62.26 | 61.16 | 63.84 |  |
| S + A + I | 73.52 | 78.62 | 78.5 | 63.38 | 70.88 | 70.8 |  |
| S + A + W | 72.96 | 78.89 | 79.16 | 60.81 | 70.71 | 70.9 |  |
| M + S + I + W | 74.19 | 79.43 | 79.36 | 64.91 | 77.38 | 77.55 |  |
| M + S + A + I | 74.51 | 79.05 | 79.27 | 63.5 | 68.6 | 70.6 |  |
| M + S + A + W | 74.63 | 79.81 | 79.41 | 64.69 | 76.94 | 77.04 |  |
| M + S + I + A + W | 74.59 | 79.6 | 79.86 | 64.68 | 76.97 | 77.05 |  |

Table 1.3: Accuracy [\%] of the frame disambiguation task

The table shows that, taken each group of features individually, the syntactic features performed best achieving accuracy $78.33 \%$ over the baseline $68.27 \%$ (using decition trees). Idiomatic features scored worst and even brought little improvement when combined with other types of features. This is mainly due to low number of idioms defined in the VALLEX lexicon, and therefore low number of idioms in the data.

Morphological features turned out to be the second best type when measured individually.

### 4.3 Importance of the Features

We summed the number of applications of individual features in decision trees weighted by 1 for the features used in the root of decision trees, by 0.5 for the features applying in the first level of decision trees, by 0.25 for features applying in the second level, etc.

Over the whole data (including all 10 runs of cross-validation), 78 features were used at least once, and 345 features were not used at all. Details can be seen in Table 3.2.

Table 1.4 shows the features which resulted as the most important ones, and their

| Feature type | Feature description | Weight |
| :--- | :--- | ---: |
| Syntax-based | Presence of reflexive particle se dependent on the verb | 51.5 |
| Syntax-based | Presence of preposition in accusative dependent on the verb | 26 |
| Morphological | Gender of the word following the verb | 17.5 |
| Syntax-based | Presence of a noun or a nom. pron. in dative dependent on the verb | 13.5 |
| Morphological | Part of speech of the word following the verb | 8 |
| Morphological | Gender of the verb | 7.5 |
| Syntax-based | Presence of preposition $z$ in genitive dependent on the verb | 7 |
| Morphological | Voice of the verb | 6.25 |
| Syntax-based | Presence of preposition in dative dependent on the verb | 6.125 |
| Syntax-based | Presence of a verb (in infinitive) dependent on the verb | 6 |
| Morphological | Case of the word two possitions after the verb | 6 |
| Syntax-based | Presence of preposition $z a$ in accusative dependent on the verb | 5.5 |
| Syntax-based | Presence of preposition in local dependent on the verb | 5.5 |
| Syntax-based | Presence of noun or a subst. pron. in instrumental dep. on the verb | 5.5 |
| Syntax-based | Presence of reflexive particle si dependent on the verb | 5 |

Table 1.4: Features most often chosen in the decision trees
respective relative weights. Syntax-based features were used most often for important decisions.

## 5 Conclusion

We have performed automatic disambiguation of verb valency frames using machine learning techniques. We have tried various types of features describing context of verbs. Syntaxbased features have shown to be most effective.

Currently we are working on applying the methods on larger lexical resources, namely the tectogrammatically annotated part of the Prague Dependency Treebank, which uses PDT-VALLEX as a frames definition, and PropBank.

We are also aiming at improving the feature set, by elaborating individual groups of features, for example by using a richer idiomatic lexicon, extending the coverage of semantic classes, or by adding other syntax-based characteristics.

## Acknowledgements

The research reported in this paper has been partially supported by the project of Information Society No 1ET101470416 and by the grant of the Grant Agency of the Charles University No. 372/2005/A-INF/MFF.

## Bibliography

Carletta, J., A. Isard, S. Isard, J. Kowtko, G. Doherty-Sneddon, and A. Anderson (1996). Herc dialogue structure coding manual. Technical Report HCRC/TR-82, HCRC.

Bojar, O., J. Semecký, and V. Benešová (2005). VALEVAL: Testing VALLEX Consistency and Experimenting with Word-Frame Disambiguation. Prague Bulletin of Mathematical Linguistics 83.

Charniak, E. (2000, April). A Maximum-Entropy-Inspired Parser. In Proceedings of NAACL-2000, Seattle, Washington, USA, pp. 132-139.
Erk, K. (2005). Frame assignment as word sense disambiguation. In Sixth International Workshop on Computational Semantics (IWCS), Tilburg.
Fellbaum, C. (1998). WordNet An Electronic Lexical Database. The MIT Press.
Hajič, J. (1998). Building a Syntactically Annotated Corpus: The Prague Dependency Treebank. Issues of Valency and Meaning, 106-132.
Hajič, J. (2000). Morphological Tagging: Data vs. Dictionaries. In Proceedings of ANLPNAACL Conference, Seattle, Washington, USA, pp. 94-101.
Kocek, J., M. Kopřivová, and K. Kučera (Eds.) (2000). Czech National Corpus - Introduction and User Handbook (in Czech). Prague: FF UK - ÚČNK.
Lopatková, M., O. Bojar, J. Semecký, V. Benešová, and Z. Žabokrtský (2005). Valency Lexicon of Czech Verbs VALLEX: Recent Experiments with Frame Disambiguation. In 8th International Conference on Text, Speech and Dialogue.
Pala, K. and P. Smrž (2004). Building Czech Wordnet. Romanian Journal of Information Science and Technology, 79-88.
Quinlan, J. R. (2005). Data Mining Tools See5 and C5.0. http://www.rulequest.com/see5-info.html.
Sgall, P., E. Hajičová, and J. Panevová (1986). The Meaning of the Sentence and Its Semantic and Pragmatic Aspects. Prague, Czech Republic/Dordrecht, Netherlands: Academia/Reidel Publishing Company.

Vossen, P., L. Bloksma, H. Rodriguez, S. Climent, N. Calzolari, A. Roventini, F. Bertagna, A. Alonge, and W. Peters (1997). The EuroWordNet Base Concepts and Top Ontology. Technical report.

Žabokrtský, Z. and M. Lopatková (2004, May 6, 2004). Valency Frames of Czech Verbs in VALLEX 1.0. In proceedings of the Workshop of the HLT/NAACL Conference, pp. 70-77.

# The Interplay of Syntax and Morphology in Building Parsing Models for Modern Hebrew 

Reut Tsarfaty<br>Institute for Logic, Language and Computation, University of Amsterdam<br>rtsarfat@science.uva.nl


#### Abstract

As of yet, there is no statistical parser for Modern Hebrew (MH). Current practice in building parsing models is not immediately applicable to languages that exhibit strong interaction between syntax and morphology, e.g. Modern Hebrew, Arabic and other Semitic languages. We suggest that incorporating morphological and morphosyntactic information into the parsing model is essential for parsing Semitic languages. Using a morphological analyzer, a part-of-speech tagger, and a PCFG-based general purpose parser, we segment and parse unseen MH sentences using a small annotated corpus. The Parseval scores obtained are not comparable to those of, e.g., state-of-the-art models for English, due to remaining syntactic ambiguity and limited morphological treatment. We conjecture that adequate morphological and syntactic processing of MH should be done in a unified framework in which morphology and syntax can freely interact and share information in both directions.


## 1 Introduction

The structure of Semitic languages poses clear challenges to the traditional view of Natural Language Processing, in which different processing layers ${ }^{1}$ are handled separately. Specifically, Semitic languages demonstrate strong interaction between morphological and syntactic processing, which limits or precludes the application of standard tools and techniques for parsing Semitic languages.

The problem, in essence, is as follows. Modern Hebrew (MH), Arabic, and other Semitic languages, have a rich morphology. Affixes that are appended to the stem of a word carry substantial information and serve different syntactic functions. Therefore, a first step towards utterance understanding is to extract the different constituents that exist at the word level to allow for further processing (e.g., parsing). However, because of the large-scale morphological ambiguity in Semitic languages already at the word level, and due to the lack of vocalization in written texts, each word-form may have multiple possible morphological analyses. Picking out the correct analysis is largely dependent on contextual information, which may be carried over syntactic structures. Therefore, a suitable treatment of morphological analysis in Semitic languages demands a treatment of syntactic analysis and vice versa.

[^61]This work focuses on MH and presents a baseline architecture for parsing that incorporates one level of morphological processing, namely morphological segmentation. The particular contribution of this work is to demonstrate that MH statistical parsing is feasible, even with a relatively small set of annotated data. Yet, in the current setting, our results fall behind those achieved for, e.g., English, which may be due to corpus size, annotation scheme, limited morphological treatment, and flexible sentence structure. In the future we intend to develop models that implement a closer interaction between morphological and syntactic processing, which are better suited for capturing linguistic phenomena in Semitic languages, and are expected to boost MH parsing accuracy.

## 2 Linguistic Data

### 2.1 Semitic Morphology

Morphological analysis of a MH word consists of, at least, the stem, prefixes, person, number and gender inflections, pronominal suffixes, and so on (Segal 2000; Bar-Haim 2005; Sima'an et al. 2001). The different morphological processes that take place in the formation of MH words can be roughly divided into (i) derivational morphology, (ii) inflectional morphology and (iii) concatenation.

Verbs, nouns, and adjectives in Semitic languages are derived from (tri-)consonantal roots plugged into templates of consonant/vowel skeletons. The lexical items in (1), for example, are all derived from the same root, $[i][l][d] .{ }^{2} \quad\left({ }^{〔} \ldots\right.$. indicates surface forms, $[c]$ indicates template's slots for root's consonants, $(c)$ indicates doubling of root's consonants.)
a. 'ild'
b. 'iild'
c. 'mwld'
$[i] e[l] e[d] \quad[i] i[l](l) e[d]$
a child (n)
deliver a child (v)
$m u[][l](l) a[d]$
innate (adj)

In addition, MH has a rich array of agreement features expressed at the word level. Features such as gender, number and person are expressed in the word's inflectional morphology. Verbs, adjectives, determiners and even numerals have to agree on the inflectional features with the noun they complement or modify. For example, in (2b) the suffix heh (h) alters the noun 'ild' (child) and its modifier 'gdwl' (big) to feminine gender.
a. ild gdwl
b. ildh gdwlh
child.MS big.MS
child.FS big.FS
a big boy
a big girl

Finally, many particles in MH, such as conjunctions, prepositions, complementizers and relativizers, are prefixed to the word. Such particles serve syntactic functions that are distinct from that of the stem, yet a multiplicity of them may be concatenated together with the stem to form a single (space-delimited) word. For example, the word form in (3) is formed from a conjunction, a relativizer, a preposition, and a definite noun phrase.

[^62]

Figure 1.1: Syntactic Structures of MH Phrases ('...' mark word boundaries)
a. 'wksmhbit'
w ks m h bit
and when from the house
Identifying such constituents within words is crucial for analyzing the syntactic structure of sentences, as they reveal structural dependencies such as subordinate clauses, adjuncts, and prepositional phrase attachment.

### 2.2 Syntactic Structures

Turning now to syntactic structures in MH, we first note that sentences in MH have a relatively free word order. ${ }^{3}$ In general, MH allows for both SV and VS, and in some circumstances for SVO permutations such as VSO and others (Shlonsky 1997). To illustrate, figures 1.1a-1.1b show two distinct syntactic structures that express the same grammatical relations.

Further, as a result of the concatenation process the constituents that are combined to form phrases and sentences in MH are not words, but rather, the morphological constituents that were concatenated together to form words. Figure 1.1c demonstrates that a MH wordform may coincide with a single constituent, as in 'ica' (leave, go out), it may overlap with an entire phrase, as in ' h ild' (the boy), or it may span across phrases as in 'w m h bit' (and from the house). Thus, it becomes clear that in order to perform syntactic analysis (parsing) of MH sentences we must first set the sequence of morphological constituents in place.

### 2.3 The Problem: Ambiguity

MH and other Semitic languages exhibit a large-scale ambiguity at the word level. This means that there are multiple ways in which a word can be broken down into its constituent morphemes. This is further complicated by the fact that most vocalization marks (diacritics) are omitted in MH texts. The word-form 'fmnh', for instance, has four readings, to

[^63]| Segmentation: | fmnh | fmnh | fmnh | fmnh | $\mathrm{f}+\mathrm{mnh}$ |
| :--- | :--- | :--- | :--- | :--- | :--- |
| Vocalization: | shmena | shamna | shimna | shimna | she + mana |
| Analysis: | fat.FS | grew-fat.FS | lubricate.FS | oil-of.FS | that + counted |
| Meaning: | fat (adj) | grew fat (v) | lubricate (v) | her oil (n) | that counted (rel) |

Table 1.1: Morphological Analyses of the Word Form 'fmnh'
a.

b.


Figure 1.2: Morphological Ambiguity Resolution in Different Syntactic Contexts
which (at least) five morphological analyses can be found, as shown in table 1.1. ${ }^{4}$ Moreover, the different morphological analyses of a word may give rise to different segmentation possibilities. In the case of the word-form 'fmnh' the five morphological analyses correspond to two distinct morphological segmentation possibilities, as observed in the table.

The morphological analysis of a word-form, and in particular its morphological segmentation, cannot be disambiguated without reference to context, i.e., an utterance. When context is available, various syntactic features of surrounding forms provide useful hints for choosing the correct analysis. Figures $1.2 \mathrm{a}-1.2 \mathrm{~b}$ show the correct analyses of the form 'fmnh' in the different syntactic contexts in which they appear. Note that the correct morphological analysis maintains agreement on gender (M/F) and number (S/P) between the noun and the verb or the adjective. In particular, the analysis 'that counted' is easily picked out for 1.2 b as it is the only one maintaining agreement with the modified noun.

Therefore, we would want to conclude that syntactic processing (parsing) must precede morphological analysis; however, this would be in apparent contradiction to our previous conclusion. For this reason, independent morphological and syntactic mechanisms for MH will not suffice. In what follows we describe a parsing architecture that incorporates one level of morphological processing, namely segmentation, as a first attempt to model the interaction between morphological and syntactic processing. We further observe that the morphosyntactic categories that are assigned to morphological segments must coincide with the lowest level of non-terminals in the syntactic parse tree. Therefore, we incorporate an intermediate level of processing, part-of-speech (POS) tagging, to ensure agreement between the morphological and the syntactic tasks.

[^64]
## 3 Formal Settings

Before describing our baseline architecture, we first develop a formal account of an integrated model for morphological and syntactic processing in a generative probabilistic framework.

Let $w_{1}^{m}$ be a sequence of words from a fixed vocabulary (i.e., a sequence of surface word-forms as they occur in the text), let $s_{1}^{n}$ be a sequence of segments of words from a (different) vocabulary, let $t_{1}^{n}$ be a sequence of morphosyntactic categories from a finite tag set, and let $\pi$ be a syntactic parse tree.

We define morphological segmentation as the task of identifying the sequence of morphological constituents that were concatenated to form a sequence of words. Formally, we define the task as (1.1), where $\operatorname{seg}\left(w_{1}^{m}\right)$ is the set of segmentations resulting from all possible morphological analyses of the words.

$$
\begin{equation*}
s_{1}^{n *}=\underset{s_{1}^{n} \in \operatorname{seg}\left(w_{1}^{n}\right)}{\operatorname{argmax}} P\left(s_{1}^{n} \mid w_{1}^{m}\right) \tag{1.1}
\end{equation*}
$$

Syntactic analysis, parsing, is the task of identifying the structures of phrases and sentences. In MH, such tree structures combine segments of words that serve different syntactic functions. Formally, we define it as (1.2), where yield $(\pi)$ is the ordered set of leaves of the syntactic parse tree.

$$
\begin{equation*}
\pi^{*}=\underset{\pi \in\left\{\pi^{\prime}: y \operatorname{yiel}\left(\left(\pi^{\prime}\right)=s_{1}^{n}\right\}\right.}{\operatorname{argmax}} P\left(\pi \mid s_{1}^{n}\right) \tag{1.2}
\end{equation*}
$$

The part-of-speech (POS) tagging task is concerned with assigning morphosyntactic categories to words. Following our theoretical exposition in section 2, it becomes clear that in MH categories are assigned to morphological segments rather than to words. So we define the task of POS tagging as (1.3), where analyses $\left(s_{1}^{n}\right)$ is the set of possible POS tags' assignments for a sequence of morphological segments.

$$
\begin{equation*}
t_{1}^{n *}=\underset{t_{1}^{n} \in \operatorname{analyses}\left(s_{1}^{n}\right)}{\operatorname{argmax}} P\left(t_{1}^{n} \mid s_{1}^{n}\right) \tag{1.3}
\end{equation*}
$$

The task of the integrated model for morphological and syntactic processing is to find the most probable morphological segmentation and syntactic parse tree given a sequence of word-forms, as in (1.4).

$$
\begin{equation*}
\left\langle\pi, s_{1}^{n}\right\rangle^{*}=\underset{\left\langle\pi, s_{1}^{n}\right\rangle}{\operatorname{argmax}} P\left(\pi, s_{1}^{n} \mid w_{1}^{m}\right) \tag{1.4}
\end{equation*}
$$

We can rewrite (1.4) using conditional probabilities, thus distinguishing the morphological and syntactic tasks, yet conditioning the latter on the former.

$$
\begin{equation*}
\left\langle\pi, s_{1}^{n}\right\rangle^{*}=\underset{\left\langle\pi, s_{1}^{n}\right\rangle}{\operatorname{argmax}} \underbrace{P\left(\pi \mid s_{1}^{n}, w_{1}^{m}\right)}_{\text {parsing }} \underbrace{P\left(s_{1}^{n} \mid w_{1}^{m}\right)}_{\text {segmentation }} \tag{1.5}
\end{equation*}
$$

In order to ensure agreement between the morphological and syntactic tasks, we incorporate an intermediate level of POS tagging into the model, which ensures that the
morphosyntactic categories assigned to the morphological segments coincide with the lowest level of non-terminals in the syntactic parse trees (cf. (Charniak et al.1996)). This results in (1.7).

$$
\begin{gather*}
\left\langle\pi, t_{1}^{n}, s_{1}^{n}\right\rangle^{*}=\underset{\left\langle\pi, t_{1}^{n}, s_{1}^{n}\right\rangle}{\operatorname{argmax}} P\left(\pi, t_{1}^{n}, s_{1}^{n} \mid w_{1}^{m}\right)  \tag{1.6}\\
=\underset{\left\langle\pi, t_{1}^{n}, s_{1}^{n}\right\rangle}{\operatorname{argmax}} \underbrace{P\left(\pi \mid t_{1}^{n}, s_{1}^{n}, w_{1}^{n}\right)}_{\text {parsing }} \underbrace{P\left(t_{1}^{n} \mid s_{1}^{n}, w_{1}^{n}\right)}_{\text {tagging }} \underbrace{P\left(s_{1}^{n} \mid w_{1}^{m}\right)}_{\text {segmentation }} \tag{1.7}
\end{gather*}
$$

Finally, we employ the assumption that $P\left(w_{1}^{m} \mid s_{1}^{n}\right) \approx 1$, since morphological segments can only be conjoined in a certain order. ${ }^{5}$ So, instead of (1.5) and (1.7) we end up with (1.8), (1.9) respectively.

$$
\begin{align*}
& \approx \underset{\left\langle\pi, s_{1}^{n}\right\rangle}{\operatorname{argmax}} \underbrace{P\left(\pi \mid s_{1}^{n}\right)}_{\text {parsing }} \underbrace{P\left(s_{1}^{n} \mid w_{1}^{m}\right)}_{\text {segmentation }}  \tag{1.8}\\
& \approx \underset{\left\langle\pi, t_{1}^{n}, s_{1}^{n}\right\rangle}{\operatorname{argmax}} \underbrace{P\left(\pi \mid t_{1}^{n}, s_{1}^{n}\right)}_{\text {parsing }} \underbrace{P\left(t_{1}^{n} \mid s_{1}^{n}\right)}_{\text {tagging }} \underbrace{P\left(s_{1}^{n} \mid w_{1}^{m}\right)}_{\text {segmentation }} \tag{1.9}
\end{align*}
$$

## 4 Evaluation Metrics

The intertwined nature of morphology and syntax in MH also challenges standard parsing evaluation metrics, as the proposed segmentation need not coincide with the gold segmentation for a given sentence. Therefore, we cannot use morphemes as the basic units for comparison. Since words are complex entities that can span across phrases, we cannot use them for comparison either. Therefore, we redefine precision and Recall by considering the spans of syntactic categories based on the (space-free) sequences of characters they correspond to. Formally, we define syntactic constituents as $\langle i, A, j\rangle$ where $i, j$ mark the location of characters, we define $T=\{\langle i, A, j\rangle \mid A$ spans from $i$ to $j\}$ and $G=\{\langle i, A, j\rangle \mid A$ spans from $i$ to $j\}$ as the test/gold parse trees respectively, and calculate as follows.

$$
\begin{gather*}
\text { labeled precision }=\frac{\#(G \cap T)}{\# T}  \tag{1.10}\\
\text { labeled recall }=\frac{\#(G \cap T)}{\# G} \tag{1.11}
\end{gather*}
$$

[^65]
## 5 Experimental Setup

### 5.1 The Baseline Architecture

Our departure point for the syntactic analysis of MH is that the basic units for processing are not words but the morphological segments that are concatenated together to form words. Therefore, we obtain a segment-based probabilistic grammar by training a probabilistic context-free grammar on a segmented and annotated MH corpus (Sima'an et al. 2001), in which segments are assigned morphosyntactic categories and are combined to form syntactic structures. Then, we use existing tools - i.e., a morphological analyzer (Segal 2000), a part-of-speech tagger (Bar-Haim 2005; Bar-Haim et al. 2005), and a general purpose parser (Schmid 2000) - in conjunction to segment and parse unseen sentences.

The Data The data set we use is taken from the MH treebank (Sima'an et al. 2001) which consists of 5001 sentences from the daily newspaper 'ha'aretz'. We employ the syntactic categories and POS tag sets developed in (Sima'an et al. 2001). We concentrate on segmentation information and ignore inflectional morphology altogether as it would lead to extreme data sparseness. The data set we use includes 3257 sentences of length greater than 1 and less than 21. The number of segments per sentence is $60 \%$ higher than the number of words per sentence. ${ }^{6}$ We conducted 8 experiments in which the data is split into training and test sets that are disjoint, and apply cross-fold validation to obtain robust averages.

The Morphological Analyzer A morphological analyzer helps to recover the segmentation of words by identifying their morphological constituents together with the corresponding morphosyntactic categories. Various analyses may be proposed for each word. A few standalone morphological analyzers for MH have been developed using different techniques and employing different tag sets ((Yona 2004), (Adler and Gabai 2005), (Segal 2000), (Bojan 2006)). In this work, we use Segal's morphological analyzer (Segal 2000) as it was shown to be robust and achieved the best coverage so far ( $96 \%$ ). Since the morphosyntactic categories employed by the analyzer differ from the POS tags in the treebank, we use an automatic translation of the analyzer's output to the treebank's annotation scheme. ${ }^{7}$

The Part-of-Speech (POS) Tagger The most comprehensive work on POS tagging for MH to date is MorphTagger (Bar-Haim 2005). This work uses Hidden-Markov-Models (HMMs) for POS tagging of Semitic languages. One of the tasks of MorphTagger is to pick out the segmentation of words to allow for correct POS tags' assignment. Therefore, MorphTagger uses a tri-gram model that provides short-contextual information to support disambiguation, and picks out the most probable segmentation and POS tags in context.

[^66]The Parser To keep our preliminary exploration formally and computationally simple, we start out with a general purpose PCFG parser to which simple Maximum Likelihood (ML) estimation methods can be applied. LoPar (Schmid 2000) is a general purpose parser for PCFGs which can be used for statistical viterbi-like parsing with any grammar or tag set. Therefore, we can use it in conjunction with the segment-based treebank grammar we obtained to parse sequences of morphological segments. Further, LoPar can parse both tagged and untagged sequences, which allows us to explore different architectural settings.

The Models We devise and implement two baseline models that are inspired by the formal account we developed in section 3 .

In the first model, henceforth Model I, we use the morphological analyzer and MorphTagger to find the most probable segmentation for a given sentence. This is done by providing MorphTagger with multiple morphological analyses per word and letting it find the segmentation that maximizes the sum $\sum_{t_{1}^{n}} P\left(t_{1}^{n}, s_{1}^{n} \mid w_{1}^{m}\right)$ (Bar-Haim 2005, section 8.2). Then, the parser is used to find the most probable parse tree for the selected sequence of morphological segments. Formally, this model is an approximation of equation (1.8) (albeit a crude one, as we perform a step-wise maximization rather than making a joint decision). ${ }^{8}$

In Model II we percolate the morphological ambiguity further, to the lowest level of non-terminals in the syntactic parse trees (i.e., the POS tags). Here we use the morphological analyzer and MorphTagger in conjunction to find the most probable segmentation and POS tag assignment by maximizing the joint probability $P\left(t_{1}^{n}, s_{1}^{n} \mid w_{1}^{m}\right)$ (Bar-Haim 2005, section 5.2). Then, the parser is used to find the most probable parse tree for a sequence of segments enriched with their morphosyntactic categories. Formally, this model attempts to approximate equation (1.9). (Note that here we couple a morphological and a morphosyntactic decision, as we are looking to maximize $P\left(s_{1}^{n}, t_{1}^{n} \mid w_{1}^{m}\right) \approx P\left(t_{1}^{n} \mid s_{1}^{n}\right) P\left(s_{1}^{n} \mid w_{1}^{m}\right)$ (cf. equation 1.9). Then we constrain the space of possible syntactic trees to those that confine with the result of the joint maximization. $)^{9}$

Smoothing Because of the relatively small size of our corpus (less then $10 \%$ of the WSJ portion of the Penn treebank), we encounter a sparse data problem in all levels of processing. In the current architecture, smoothing the estimated probabilities is delegated to each of the relevant subcomponents of the integrated architecture. Out of vocabulary

[^67](OOV) words are treated by the morphological analyzer, which proposes all possible segmentations assuming that the stem is a proper noun. The Tri-gram language model used by MorphTagger is smoothed using Good-Turing discounting (the so-called 'Katz backoff', see (Bar-Haim 2005, section 6.1)),,$^{10}$ and the parser uses a variant of absolute discounting, in which the discounted value is redistributed according to various backoff strategies to events with zero frequency encountered in the parsing process (Schmid 2000, section 4.4).

Evaluation We use seven measures to evaluate our integrated models. First, we present the percentage of sentences for which the model could propose a pair of corresponding morphological and syntactic analyses. This measure is referred to as string coverage. In order to capture tagging and parsing accuracy we refer to our redefined Parseval measures. We separate the evaluation of assigned morphosyntactic categories, i.e., POS tags precision and recall, and phrase-level syntactic categories, i.e., labeled precision and recall ${ }^{11}$ (where root nodes are discarded as usual, and empty trees are counted as zero). Finally, we report segmentation precision and recall, in order to give an impression of the morphological disambiguation capabilities of the integrated model. ${ }^{12}$

## 6 Results

Table 1.2 shows the evaluation scores for the models. Model I, in which the parser operated on segmented sequences of words, proposed compatible morphological and syntactic analyses for $99 \%$ of the unseen sentences. However, the accuracy results are much lower $-60.3 \%$ and $58.4 \%$ labeled precision and recall for parsing, and 82.4 and $82.6 \%$ precision and recall for POS tagging.

In model II, the input for the parser was enriched with morphosyntactic categories that were selected in tandem with the segmentation. This improved labeled precision and recall in $0.5 \%$ and $2.1 \%$ respectively, and POS tagging precision and recall in $2.1 \%$. However, together with the improved accuracy we observe a decrease of $3 \%$ in string coverage. This means that the capability of the model to provide compatible morphological and syntactic analyses has dropped. Also, we observe a decrease of $3 \%$ in our segmentation results, which is mainly due to the drop in string coverage.

[^68]|  | String <br> Coverage | Labeled <br> Precision | Labeled <br> Recall | POS tags <br> Precision | POS tags <br> Recall | Segment. <br> Precision | Segment. <br> Recall |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Model I | $99.2 \%$ | $60.3 \%$ | $58.4 \%$ | $82.4 \%$ | $82.6 \%$ | $94.4 \%$ | $94.7 \%$ |
| Model II | $96.0 \%$ | $60.8 \%$ | $60.5 \%$ | $84.5 \%$ | $84.7 \%$ | $91.3 \%$ | $91.6 \%$ |

Table 1.2: Evaluation Metrics, Models I and II

## 7 Analysis

This work presents a first set of statistical parsing standardized results for MH. The high string coverage score demonstrates that, in principle, models that incorporate morphological information can parse unseen sentences based on segmented and annotated corpora. Furthermore, comparison of the two models shows that coupling the morphological decision with a morphosyntactic one (currently only based on short context) improves parsing accuracy. Yet, the scores we report show that this is still insufficient for broad-coverage parsing with high accuracy comparable to other languages.

The reasons for the low parsing accuracy are several. First, the results were obtained using a relatively small set of training data, and a weak (unlexicalized) parser. ${ }^{13}$ Further, the low accuracy is partially due to the severe ambiguity of the resulting PCFG. Since word order in MH is relatively free, CFG rules can appear in various permutations, which in turn leads to major structural ambiguity. This indicates that bare phrase structures are not adequate for capturing regularities in MH , especially with limited training data. Since we included only limited amount of morphological information that hints on possible dependencies, the parser has very limited means to recover from that.

A comparison between the models shows that while POS tags' assignment helps to improve parsing accuracy, it has negative effects on string coverage. The reason for that is that a probable yet incorrect POS tag assignment constrains the parser in a way that makes it impossible for it to recover correct syntactic structures. A POS tagger that is optimized towards syntactic decisions based on short context may result in imperfect disambiguation, especially for a language such as MH , in which long distance dependencies (e.g., due to agreement) are likely to be found.

Thus, we conclude that POS tagging is perhaps insufficient for enforcing agreement between the morphological and syntactic tasks, and propose to include larger contexts for disambiguation. We conjecture that only more extensive information sharing between the two levels of processing, i.e., morphological patterns and inflections on the one hand, and syntactic dependencies on the other hand, will allow for successful syntactic and morphological disambiguation.

[^69]| Alphabet | aleph | bet | gimel | dalet | heh | vav | zayin | chet | tet | yod | khaf |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Transliteration | a | b | g | d | h | w | z | x | j | i | k |
| Pronounciation | , | $\mathrm{b}, \mathrm{v}$ | g | d | h | v | z | kh | t | y | $\mathrm{k}, \mathrm{kh}$ |
| Alphabet | lamed | mem | nun | samech | 'ayin | peh | tsadi | kof | reish | shin | tav |
| Transliteration | l | m | n | s | e | p | c | q | r | f | t |
| Pronounciation | l | m | n | s | , | $\mathrm{p}, \mathrm{ph}$ | ts | k | r | $\mathrm{sh}, \mathrm{s}$ | t |

Table 1.3: Transliteration

## 8 Conclusion

Traditional approaches for devising parsing models and defining evaluation metrics are not adequate for MH , as they presuppose a certain language structure and separate layers of processing. Parsing Semitic languages requires serious morphological consideration, and we have shown that incorporating morphological cues (most crucially segmentation) and morphosyntactic information (currently based on short context) helps to recover parses for MH sentences. However, the high variability of the phrase structure, severe structural ambiguity, and relatively small amount of annotated data make it insufficient for completing the parsing task successfully.

Different languages mark regularities in their surface structures in different ways. English encodes regularities in word order, while MH provides useful hints for grammatical relations in its derivational and inflectional morphology. Much more work is required to prove our thesis that exploiting such information to discriminate between syntactic structures helps to correctly recover structural dependencies. In the future, we intend to develop more sophisticated models, allowing for closer interaction between morphological and syntactic processing, in order to improve parsing accuracy and facilitate morphological disambiguation.

## Acknowledgements

This work is funded by the Netherlands Organization for Scientific Research (NWO), grant number 017.001.271. I would like to thank Khalil Sima'an for supervising this work, and to Remko Scha for detailed comments on an earlier draft. The Knowledge Center for Hebrew Processing at the Technion, Israel, provided corpora and processing tools, and Roy BarHaim provided much knowledge and technical support concerning morphological analysis and part-of-speech tagging of MH, without which this work would have been impossible.

## 9 Transliteration

Table 1.3 illustrates the transliteration scheme for the MH alphabet we adopt from (Sima'an et al. 2001).

## Bibliography

Meni Adler and Dudi Gabai. Morphological analyzer and disambiguator for Modern Hebrew. Knowledge Center for Processing Hebrew, 2005.

Roy Bar-Haim. Part-of-speech tagging for hebrew and other semitic languages. Master's thesis, Technion, Haifa, Israel, 2005.

Roy Bar-Haim, Khalil Sima'an, and Yoad Winter. Choosing an Optimal Architecture for Segmentation and POS-Tagging of Modern Hebrew. In Proceedings of $A C L$ 2005 Workshop on Computational Approaches to Semitic Languages, 2005.
Dalia Bojan. HAMSA: A Morphological Analyzer for Hebrew Texts. The Knowledge Center for Hebrew Processing, 2006. URL http://yeda.cs.technion.ac.il: 8088/analyzer1/analyzer1.html.

Eugene Charniak, Glenn Carroll, John Adcock, Anthony R. Cassandra, Yoshihiko Gotoh, Jeremy Katz, Michael L. Littman, and John McCann. Taggers for Parsers. AI, 85(1-2):45-57, 1996.

Helmut Schmid. LoPar: Design and Implementation. Institute for Computational Linguistics, University of Stuttgart, 2000.

Erel Segal. A Probabilistic Morphological Analyzer for Hebrew undotted texts. Master's thesis, Computer Science Department, Technion, Isreal, 2000.
Ur Shlonsky. Clause Structure and Word Order in Hebrew and Arabic: An Assay in Comparative Semitic Syntax. Oxford University Press, 1997.
Khalil Sima'an, Alon Itai, Yoad Winter, Alon Altman, and Noa Nativ. Building a Tree-Bank for Modern Hebrew Text. In Traitement Automatique des Langues, volume 42. 2001. URL http://mila.cs.technion.ac.il/website/ hebrew/resources/corpora/treebank/index.html.

Shlomo Yona. A Finite-state based Morphological Analyzer for Hebrew. Master's thesis, University of Haifa, Israel, November 2004.


[^0]:    ${ }^{1}$ The prefix al-becomes ' $l$ - when the noun occurs after a word ending with a vowel
    ${ }^{2}$ Available at www.informatik.uni-stuttgart.de/ifi/bs/research/arab_e.html

[^1]:    ${ }^{3}$ In all this paper, the atomic type s represents well-formed nominal sentences
    ${ }^{4}$ Each np is decorated by its definiteness feature ( $d e f$, ind) followed by its case feature

[^2]:    ${ }^{5}$ Recall that $\iota$, the description operator, is of type $(\mathrm{e} \rightarrow \mathrm{t}) \rightarrow \mathrm{e}$ and $(\iota \mathrm{P})$ returns the only individual that verifies the property P

[^3]:    ${ }^{6}$ The key $\diamond_{c s}$ allows to open $\square_{c s}$ lock since $\diamond_{c s} \square_{c s} \mathrm{~A} \vdash \mathrm{~A}$

[^4]:    ${ }^{7}$ To preserve the form-meaning correspondence, we will assume that the syntactic type np is lifted to $\left(\mathrm{s} /{ }_{0} \mathrm{np}\right) \backslash_{0} \mathrm{~s}$

[^5]:    ${ }^{1}$ Let $w \in \operatorname{Nec}^{\prime}{ }_{\mathrm{Sc}}(p, q)$ and assume that (3.4) is true but (3.5) is false. Hence there is a $q^{\prime} \in \operatorname{Alt}(q)$ with $q^{\prime} \leq^{\mathrm{E}} q$ and $w \in \operatorname{Enable}\left(p, q^{\prime}\right)$. As $Q$ is finite, let $q^{\prime}$ be $\leq^{\mathrm{E}}$-least with these properties. From $w \in \operatorname{Enable}\left(p, q^{\prime}\right)$ we know that there is a $u \in f(w)$ with $u \in q^{\prime}$ and $u \in p_{\delta}^{*}$. Since $q^{\prime}$ is least, we can conclude from (3.4) that $u \in q^{\downarrow}$ which contradicts $u \in p^{\prime}$, as $q^{\prime} \neq q$ and $q^{\prime} \leq^{\mathrm{E}} q$.
    Now let $w \in \operatorname{Nec}^{\prime}{ }_{S c}(p, q)$ and assume that (3.5) is true but (3.4) is false. Then there is a world $u \in f(w)$ such that $u \in p_{\delta}^{*}$, there is no $v \in f(w)$ with $v \in p_{\delta}^{*}$ and $v<^{\mathrm{E}} u$, but $u \notin q^{\downarrow}$. If $u \notin q^{\downarrow}$, then $u \in q_{u}$ with $q_{u} \in \operatorname{Alt}(q)$ and $q_{u} \leq^{\mathrm{E}} q$. From (3.5) we then derive that $w \notin \operatorname{Enable}\left(p, q_{u}\right)$. But this means that $u \notin p_{\delta}^{*}$, as $u$ is the only world in $f(w)$ which makes $q_{u}$ true.

[^6]:    ${ }^{2}$ It is here that the assumption of an 'at least'-reading pays off. Had I not assumed an 'at least'-reading, I would have had to invest more effort and ink into the justification of this relevance ordering.

[^7]:    ${ }^{1}$ The distinction between total and partial affectedness has been extensively discussed with reference to the partitive/accusative alternation in Finnish, cf. (Krifka 1992).
    ${ }^{2}$ See (Grimm 2005) for an empirical problems with the argument constellations of verbs in the middle voice that arise if 'causally affecting another participant' is taken as primitive.

[^8]:    ${ }^{3}$ This is certainly not the only sense in which "affectedness" has been appealed to. The heterogeneous range of senses which "affectedness" has accumulated include aspectual/holistic affectedness, as in the well-discussed spray/load alternation (see (Levin and Hovav 2005, p. 209) and references therein), and a sort of empathetic affectedness whereby affectedness is correlated with animacy, the degree of animacy purportedly matching the degree of affectedness for Differential Object Marking (Naess 2004). Note that these uses are dependent on properties of objects, spatial and animacy, respectively, and not strictly of events, which permits restricting the notion of affectedness used here to be the most basic one, i.e., being altered in some manner. The general approach advocated here is that such specific notions of affectedness should arise from the interaction of event-based and object properties, and restrict what is taken as primitives of agency to a minimum. See further argumentation in (Grimm 2005).

[^9]:    ${ }^{4}$ Recall that a partially ordered set is a lattice if every non-empty finite subset has a least upper bound and a greatest lower bound.
    ${ }^{5}$ More formally: A subset U of a partially ordered set is upwards closed if $x$ in U and $x \leq y$ implies that $y$ belongs to U and, conversely, U is downwards closed if $x \geq y$ implies that $y$ belongs to U .
    ${ }^{6}$ Of course, barring contradictions of entailments, i.e., the patient argument of 'kill' will typically not be satisfied by arguments below nodes containing sentience.

[^10]:    ${ }^{7}$ This gloss seems to contradict the dative's status as marking non-volitionality. Recall from section 3.1, one of the verb types that takes dative subject is 'wanting/needing'. Yet, (Platts 1884) gives examples of this predicate, named in (Masica 1991) as 'cahiye', to be: "Is necessary, is needful. . . ; should or ought ..." Meanings such as these accord both with a non-volitional interpretation and with the deontic sense under discussion here, standing in clear contrast to the more straight-forward volitive meaning of the ergative.

[^11]:    ${ }^{1}$ Objects can cause agreement errors in sentence production as well (cf. Hartsuiker et al., 2001; Hemforth \& Konieczny, 2003; Konieczny et al., 2004). While Hartsuiker and his colleagues offer a percolation explanation, Hemforth \& Konieczny (2003) propose a feature reactivation mechanism which I will not discuss here. Note, that feature reactivation cannot explain the attraction effect found with a preceding distractor in the relative clause construction. Furthermore it cannot explain the absence of an asymmetry and the ambiguity effect described below.

[^12]:    ${ }^{1}$ Plausible as this may look for 'John' in (1), note that it is not for 'everyone' in (2).

[^13]:    ${ }^{2}$ We will only be dealing with singular anaphora.
    ${ }^{3}$ This is expressed as $\hat{o}=\hat{s}$, which implies that $e_{\hat{o}}$ is the same type as $e_{\hat{s}}$, as a result of which the type rules allow $h=p$.

[^14]:    ${ }^{4}$ This is a role Glue can play to some extend in addition to its normal role of composing meanings. The work of Fry (1999) with Glue in negative polarity licensing also uses Glue for similar purposes.

[^15]:    ${ }^{1}$ Whether it actually is a presupposition, an implicature or something else has been under debate for a long time. For recent opinions consult van Rooij and Schulz (2005), Roberts (2005) or Geurts and van der Sandt (2004) and reactions to this paper in Theoretical Linguistics, in particular Beaver (2004). Roberts talks about the "prejacent" of only instead of its "presupposition", whereas van Rooij and Schulz use the terminology "positive contribution" as opposed to the asserted "negative contribution".

[^16]:    ${ }^{2}$ Landman (1989) describes in the second part of his article on groups how individuals can have different properties in different roles they play in society, e.g. John may have two jobs, as a judge and as a janitor, where John as a judge (denoted $j \upharpoonright J$ ) may have a different income than John as a janitor $\left(j \upharpoonright J^{\prime}\right)$. Although the intersections of quantifiers that I am using may have some similarities with Landman's restricted individuals they don't have the same closure conditions which Landman claims for the latter.
    ${ }^{3}$ Whether it is indeed always nested sets of progressively decreasing numbers of authorities on which the hierarchy between these people is legally founded is of course a matter of how things are in reality. Many

[^17]:    ${ }^{5}$ A quantifier $\mathcal{Q}$ is upward monotonic iff it holds that $\forall P \forall P^{\prime}\left[\left(\mathcal{Q}(P) \wedge P \subset P^{\prime}\right) \rightarrow \mathcal{Q}\left(P^{\prime}\right)\right]$.
    ${ }^{6}$ As far as I can see, the approach in van Rooij and Schulz (2005) interprets or as exclusive disjunction. Consider a world $w_{1}$ in which John kissed Jane and nobody else, and a world $w_{2}$ with John kissing both

[^18]:    Jane and Mary. A non-exclusive interpretation of or should make (16a) true in $w_{2}$. However as the background predicate is not minimal in $w_{2}$ (after all, there is also $w_{1}$ ), (17) will predict the exclusion of $w_{2}$. I admit that this is actually a way how the sentence (16a) can be understood; so after all the effect might not been unwanted. In any case, this problem does not immediately carry over to the cases of indefinites that I am discussing.

[^19]:    ${ }^{1}$ Note that this insensitivity to syntactic structure means that these theories predict that truthconditionally equivalent sentences differing in syntactic form should receive the same interpretation. For example, John came to the party is predicted to receive the same interpretation as (4), which is obviously incorrect when strengthening is involved.
    ${ }^{2}$ We use the term "strengthened meaning" without being committed to any particular procedure for strengthening meanings. The strengthening may come through the use of an exhaustive operator in the syntax/semantics, as an implicature using some form of Gricean reasoning, through default reasoning, or any other method of shrinking the meaning of $X$.

[^20]:    ${ }^{3}$ In Zimmermans's system, the members of $\mathcal{L}$ are epistemically modalized as "it is compatible with the speaker's knowledge that." This is irrelevant to the present discussion, however, where the concern is with symmetry.
    ${ }^{4}$ Fox's system is more explicit about the syntactic assumptions made, so I will restrict discussion to that work. Nonetheless, the conclusions I reach follow for Chierchia's system just the same.
    ${ }^{5}$ Allowing the system to strengthen the first disjunct does not help generate the correct meaning, as the reader can easily verify.

[^21]:    ${ }^{6}$ In this, I assume Groenendijk and Stokhof's partition semantics for questions. See (Groenendijk and Stokhof 1997) for a survey of theories of questions, including Groenendijk and Stokhof's own partition theory.
    ${ }^{7}$ This way of setting things up means that conversations follow "strong relevance," in the sense of (Spector 2003).

[^22]:    ${ }^{8}$ For more on "levels" in questions, see (Potts 2006).
    ${ }^{9}$ For formal definitions of how such partitions arise, see Section 4.
    ${ }^{10}$ However it is that strengthening is implemented

[^23]:    ${ }^{11}$ One can imagine more fine-grained ways of partitioning the space. All that we require is that this be a fine enough partitioning of the space of possibilities. I assume it is, though the issue is not at all trivial when quantified expressions are used. Here, $[\mathrm{a}]=\{\mathrm{w}$ in $c$ : John ate all of the cookies in w$\}$, $[\mathrm{sbna}]=\{\mathrm{w}$ in $c$ : John ate some but not all of the cookies in w$\},[\mathrm{n}]=\{\mathrm{w}$ in $c$ : John ate none of the cookies in w$\}$.
    ${ }^{12}$ For an illuminating discussion of such contexts, where a question has been raised whose presuppositions are not satisfied by the context, see (Collingwood 1940; Rescher 2000). These are admittedly highly artificial contexts, and probably not all the frequent in normal discourse. Indeed, Collingwood calls this the "fallacy of many questions," and Rescher calls such questions "problematic." They do create highly defective contexts, yet, as (12) shows, we can sometimes deal with them just fine, which is all that really matters here.

[^24]:    ${ }^{13}$ By $\left[X_{1}, \ldots, X_{n}\right]$ I simply mean that set of worlds where $X_{1} \ldots X_{n}$ all hold.
    ${ }^{14}$ See the cautious words in Fn.12. A further difficulty is that it is not entirely clear how such partitions arise, which may point to a difficulty for question-based approaches, such as the one taken here. Nonetheless, imagining such a context is not all that difficult either, and, given such a context, the correct prediction is made. For instance, imagine we see John holding his stomach, moaning in pain and obviously distraught. I ask, what's wrong with him?, to which you respond with (14). Here, $[Y, p]$ is a set of worlds where he ate all of the cookies and regrets having done so, $[\neg Y, p]$ is a set of worlds where he ate all the cookies but doesn't regret having done so, and $[X, \neg p]$ is a set of worlds where he ate some but not all of the cookies.

[^25]:    ${ }^{15}$ I say nothing about the source of presupposition in this paper. I am assuming that there is some function or other that assigns presuppositions to all atomic sentences of the language.

[^26]:    ${ }^{16}$ The notation " $X ; Y$ " means first do $X$, then do $Y$. See (van Benthem 1989) for details.

[^27]:    ${ }^{1}$ For a classical presentation of Kripke semantics, see (Blackburn, de Rijke, and Venema 2002).

[^28]:    ${ }^{2}$ The actual command-line parameters passed to SPASS are: -Auto=1 -PProblem=0 -PGiven=0 -PStatistic=1 -Sorts=0 -TimeLimit=100.
    ${ }^{3}$ We consider $@_{i}$ is as an additional modality; see Chapter 7 of (Blackburn, de Rijke, and Venema 2002).

[^29]:    ${ }^{4}$ A portion of HT's timeouts are caused by yet another unexpected issue: formulas translated with HT take longer to get translated into CNF (a conversion that SPASS performs on all input before starting the superposition algorithm). For example, in some cases SPASS may spend over 2 minutes in the CNF conversion of a formula translated via HT and only 40 seconds when translated with LHT (even though the later is structurally more complex). We reported this to SPASS's developers. Not knowing the exact details of the CNF translation used by SPASS, we cannot attempt to find an explanation.

[^30]:    ${ }^{1}$ It would have been simpler here to say $@_{\text {alarm }}[R]$ safe-locked but we will use this structure to express more interesting properties later. Also, for the moment we want to discuss only some intuitive ideas. We will provide a detailed definition of the syntax and semantics of $\mathcal{H} \mathcal{L}_{\mathcal{C}}(@, \downarrow)$ in Section 2.

[^31]:    ${ }^{2}$ For the case of propositions, $\llbracket \rrbracket$ is equivalent to a standard truth-value assignment function, as we can take $\}$ as the false value and $\{\}\}$ as the true value.

[^32]:    ${ }^{3}$ Because we are working with partial functions, equality is not necessarily an equivalence relation. When a term $t$ is not defined, $t=t$ is falsified.
    ${ }^{4}$ Safe replacement means, as usual, that no free occurrence of $t_{1}$ that is replaced, is within the scope of a $\downarrow$ operator binding a variable of $t_{2}$.

[^33]:    ${ }^{1}$ I.e., if the sequent $\Gamma \vdash \Delta$ is a theorem of $\mathbf{S}$ then it has a proof in which every formula is a subformula in $\Gamma$ or $\Delta$.

[^34]:    ${ }^{2}$ In this case the point of evaluation is the shared element of all formulas that are not :-prefixed.

[^35]:    ${ }^{1}$ Notice that + is not commutative. In particular, $(t+s): \varphi \supset(s+t): \varphi$ is not a theorem of LP.

[^36]:    ${ }^{2}$ An S4 Kripke model is a triple $(G, R, V)$, where $G$ is a nonempty set whose elements are referred to as worlds, $R$ is a reflexive and transitive binary relation on $G$, and $V$ is a map from worlds to sets of propositional letters (so $V(\Gamma)$ is the set of propositional letters taken to be true at world $\Gamma$ ).

[^37]:    ${ }^{3}$ The author's LP tableau system is essentially a reformulation of Artemov's Gentzen-style system for LP (Artemov 2001).

[^38]:    ${ }^{4}$ It will also be convenient to allow for the case that a tableau is for (or begins with) a set of formulas. In this case, the tableau is constructed from the single-branched tree consisting of that set of formulas (appearing in any order).

[^39]:    ${ }^{5}$ The subformula property is the property whereby each of the formulas below the line in a rule diagram is a subformula of the formula above the line.

[^40]:    ${ }^{6}$ This is Fitting's approach in (Fitting 1999).

[^41]:    ${ }^{1}$ The actual encoding needed for Blackbox is slightly more complicated because we have to deal explicitly with the Closed World Assumption (Blackbox does not do it automatically), but the main intuitions are clear from the example we described above.

[^42]:    ${ }^{1}$ In the scope of this paper, a successful parse means a parse that spans the whole sentence.
    ${ }^{2}$ If a certain tag appears with more than 15 different words in a large corpus, it is considered universal. The outcome has been slightly manually adapted to get a systematic tagset.

[^43]:    ${ }^{3}$ An example of such a lexical type tag is verb(hebben, inf,transitive). This tag indicates an infinitival form of a verb combining with the auxiliary hebben, used transitively.
    ${ }^{4}$ Alpino's disambiguation component uses a maximum entropy model. This component is explained in detail in van Noord (2006).

[^44]:    ${ }^{5}$ Note that this evaluation framework evaluates precision and recall quite strictly: if one of the subat-

[^45]:    tributes of a certain tag is wrong, the whole tag is considered wrong. It may well be the case, however, that the algorithm missed out on only one certain subattribute of the tag, and found all other characteristics correctly.
    ${ }^{6}$ This is the score one might hope to attain by using an ordinary POS tagger.

[^46]:    ${ }^{1}$ http://www.lrec-conf.org/

[^47]:    ${ }^{2}$ Such databases are found at the Dutch National Museum for Natural History and undoubtedly at many other places.

[^48]:    ${ }^{3}$ http://about.reuters.com/~researchandstandards/corpus/statistics/
    ${ }^{4}$ http://ilk.uvt.nl/ilkcorpus/
    ${ }^{5}$ http://www.elsnet.org/resources/~eciCorpus.html

[^49]:    ${ }^{1}$ I will return to word-sense-disambiguation problems in section 8 .

[^50]:    ${ }^{2}$ They could further increase accuracy to $81.2 \%$ by the automatic clustering of head words.

[^51]:    ${ }^{3}$ The reason for this is that the positions of the semantic roles present in the search engine query are already known, just as the position of the expected answer element. In order to verify that the sentences returned from the search engine actually match the underlying pattern, a syntactic analysis of these sentences is necessary. But if this analysis shows that the syntactic structure of the sentence indeed is the desired one, it is already known which constituent fills which semantic role.

[^52]:    ${ }^{1}$ All figures can be found in the Appendix.

[^53]:    ${ }^{2}$ Due to spatial limit of this paper, here we focalise the comparison between the beginning and final stages only, without providing information about second stage, that will be included in analysis in the broader research (Rosi in preparation).

[^54]:    ${ }^{1} \mathrm{~A}$ corpus of 460 such questions was extracted from a set of FAQ sites on the Internet (I thank Jochen Leidner for his help).

[^55]:    ${ }^{2}$ http://wordnet.princeton.edu/

[^56]:    ${ }^{3}$ http://multisemcor.itc.it/semcor.php
    ${ }^{4}$ http://wn-similarity.sourceforge.net/

[^57]:    ${ }^{5}$ I have tested all five available similarity measures on the dataset, but due to space restrictions only describe here the ones that produced the best results (for an overview of these measures, see Pedersen et al. (2005)).

[^58]:    ${ }^{6}$ On this subset of 82 agreeing pairs, WUP, LIN and LESK achieved accuracies of $68.3 \%, 67.1 \%$ and $81.7 \%$ respectively, which is higher than their results on the full set of 100 pairs.

[^59]:    ${ }^{7}$ http://en.wikipedia.org/wiki/Wikipedia

[^60]:    ${ }^{1}$ http://ucnk.ff.cuni.cz/english/index.html

[^61]:    ${ }^{1}$ I.e., phonological, morphological, syntactic, semantic and pragmatic.

[^62]:    ${ }^{2}$ The transliteration we use is adopted from (Sima'an et al. 2001) and repeated in the appendix for convenience.

[^63]:    ${ }^{3}$ Relative to, e.g., English.

[^64]:    ${ }^{4}$ In fact, a statistical study on a MH corpus has shown that the average number of possible analyses per word-form was 2.1 , while $55 \%$ of the word-forms were morphologically ambiguous (Sima'an et al. 2001).

[^65]:    ${ }^{5}$ In MH, conjunctions, relativisers, prepositions and definite markers must be attached in front of the stem, pronominal and inflectional affixes appear at the end of the stem, and derivational morphology shows up inside the stem. Thus, a sequence of morphological segments can only be conjoined in a certain order. To illustrate, although the MH form 'hkph' is ambiguous between three morphological analyses; (i) 'h' + 'kph' (the + coffee) (ii) 'hkph' (lap, surrounding) and (iii) 'hkp'+'h' (perimeter + of-her), restoring the surface forms that correspond to the different sequences in (i)-(iii) must result in the word-form 'hkph'.

[^66]:    ${ }^{6}$ In the complete MH corpus the average number of words per sentence is 17 while the average number of morphosyntactic segments is 26 .
    ${ }^{7}$ We are grateful to Roy Bar-Haim for providing us with the script which he wrote (Bar-Haim 2005).

[^67]:    ${ }^{8}$ The reason for choosing the step-wise architecture as our first model is twofold. Firstly, a step-wise architecture is computationally cheaper than a joint one, but more importantly, this is perhaps the simplest end-to-end architecture for MH parsing that one could imagine. Thus, in the lack of previous MH parsing results, it is suitable to serve as a baseline architecture against which to compare more sophisticated models.
    ${ }^{9}$ We further developed a third model, Model III, which is a more faithful approximation, yet computationally affordable, of equation (1.9). In Model III we percolate the ambiguity all the way through the integrated architecture by means of providing the parser with the n-best sequences of tagged morphological segments, and selecting the analysis $\left\langle\pi, t_{1}^{n}, s_{1}^{n}\right\rangle$ which maximizes the production $P\left(\pi \mid t_{1}^{n}, s_{1}^{n}\right) P\left(s_{1}^{n}, t_{1}^{n} \mid w_{1}^{m}\right)$. However, we have not yet obtained robust results for this model prior to the submission of this paper, and therefore we leave Model III for future discussion.

[^68]:    ${ }^{10}$ In this work we did not use the bootstrapping method for smoothing the lexical model nor the various heuristics for improved handling of OOV words proposed in (Bar-Haim 2005). The reason for working with bare probabilities as estimated from the corpus is to remain faithful to the probabilities we represented in the formal exposition.
    ${ }^{11}$ Covert definite article errors are counted at the POS tags level, and discounted at the phrase-level.
    ${ }^{12}$ Since we evaluate the models' performance on an integrated task, sentences in which one of the subcomponents failed to propose an analysis counts as zero for all subtasks.

[^69]:    ${ }^{13}$ This is mainly due to the size of the corpus and its annotation scheme, which lacks head-marking.

