Manfred Pinkal, Remko Scha,
Lenhart Schubert (editors):

Semantic Formalisms in Natural Language Processing

Dagstuhl-Seminar-Report; 57
22.02.-26.02.93 (9308)

ISSN 0940-1121

Das Internationale Begegnungs- und Forschungszentrum für Informatik (IBFI) ist eine gemeinnützige GmbH. Sie veranstaltet regelmäßig wissenschaftliche Seminare, welche nach Antrag der Tagungsleiter und Begutachtung durch das wissenschaftliche Direktorium mit persönlich eingeladenen Gästen durchgeführt werden.

Bezugsadresse: Geschäftsstelle Schloss Dagstuhl
Universität des Saarlandes
Postfach 1150
D-66041 Saarbrücken, Germany
Tel.: +49 -681 - 302 - 4396
Fax: +49 -681 - 302 - 4397
e-mail: office@dag.uni-sb.de

International Colloquium on

Semantic Formalisms in Natural Language Processing

IBFI, Schloß Dagstuhl
22nd – 26th February 1993

Organizers: Prof. Dr. Manfred Pinkal (University of the Saarland)
Prof. Dr. Remko Scha (University of Amsterdam)
Prof. Dr. Lenhart Schubert (University of Rochester)

The computation, representation and processing of semantic information is an important task in the development of natural language systems. There has been a noticeable convergence of ideas in recent years, centering on the use of contextual information and dynamic aspects of utterances on the one hand, and the need for structured information, which can be differentiated according to a specific domain of application, on the other hand. Several formalisms are available, which approach problems related to these ideas with different techniques and from different angles. The colloquium has contributed to clarifying the issues surrounding the use of these formalisms in semantics research both in theoretical semantics and in artificial intelligence.

Every day of the colloquium was devoted to one main theme:

- 23.02.93: Semantics Formalisms
- 24.02.93: Syntax-Interface / Underspecified Representations
- 25.02.93: Knowledge Processing and Semantics Formalisms
- 26.02.93: Non-monotonic Reasoning and Semantics Formalisms

Many well-known experts in semantics, logic, computational linguistics and computer science took part in the seminar, all of whom have made important contributions to research in these fields.

The interdisciplinary orientation of the seminar has made it possible

- to enhance the state of information in the relevant disciplines;
- to provide an overview of existing implementations and their power;
- to illuminate the theoretic status of different decisions in implementation;
- to improve our estimates of the practical potential of theoretical semantics formalisms.

Great emphasis was placed on the exchange of ideas between the different fields and discussions between the different approaches. Furthermore, a special working group examined a catalog of criteria for evaluating the different existing approaches. This work, and subsequent plenary discussion, isolated several parameters – pertaining to the realization state, the theoretical intentions, and the coverage – relative to which the systems differ to a great extent. Further careful planning will be necessary to arrive at meaningful comparative evaluations. A practical consequence of this has been an initiative to create a special interest group with the Association of Computational Linguistics on the topic of computational semantics. It was also agreed to cooperate on creating a database of representative examples of relevant natural language phenomena.

The participants felt that the colloquium was a very successful and stimulating event. In light of the importance of such research exchange, the next meeting is projected to take place in 1995. It is to be concerned specifically with the evaluation of formalisms which have been implemented and are used in applications.

Contents

Part One: Semantics Formalisms

Martin v. d. Berg
Dynamic Semantics
Alan Black
Using a Computational Situation-Theoretic Language to Investigate Contemporary Semantic Formalisms
Robin Cooper
Towards a General Semantic Framework
Jens-Erik Fenstad
Experiences with the Framework of Situation Semantics
Hideyuki Nakashima
Situated Dialog

Part Two: Syntax-Interface / Underspecified Representations

Jan-Tore Lønning
Constraints on Unification-Based Semantics
Sebastian Millies, Manfred Pinkal
Linking One Semantic Interpretation System to Different Syntactic Formalisms
Massimo Poesio
Inferring the Semantic Scope of Operators
Stephen Pulman
Quasi- Logical Form
Uwe Reyle
Dealing with Ambiguities by Underspecification: Construction, Representation and Deduction
Mark Steedman
The Interaction of Syntax and Semantics in CCG
Robert Moore *Session Summary*

Part Three: Knowledge Processing and Semantics Formalisms

Wolfgang Wahlster
Semantic Representation Languages and/or Knowledge Representation Languages
Hans Kamp
Discourse Representation Theory as a Knowledge Representation Formalism
Chung Hee Hwang, Lenhart Schubert
Meeting the interlocking needs of LF-computation, deindexing, and inference: An organic approach to general NLP
John Nerbonne
Natural Language Semantics and Compiler Technology
Bernd Mahr
E-theory as a semantic tool for natural language reconstruction and processing

Part Four: Non-Monotonic Reasoning and Semantics Formalisms

Bernhard Nebel, Espen Vestre
Belief Revision Methods: Formal Properties and Applications in NLP
Jeffrey Pelletier
Psychologically-oriented investigations into nonmonotonic reasoning
Richmond Thomason
Non-monotonic Formalisms for Natural Language Semantics
Frank Veltman
The Framework of Update Semantics
Henk Zeevat
Update Semantics and Discourse Grammar
Christopher Habel, Manfred Krifka
Session Summary

Dynamic Semantics

Martin van den Berg
ILLC/Department of Computational Linguistics,
Faculty of Arts, University of Amsterdam
vdberg@alf.let.uva.nl

A Short Introduction to Dynamic Semantics

Real life language is built up of utterances, not sentences. An utterance is a sentence interpreted in its context—including both linguistic and non-linguistic components. The traditional way to formalize this is the use of a vector of indices. For example, the sentence

(1) *I see John walk in the park*

translates as **see**(i, **walk**(**john**)). The interpretation of this in the context (i.e. the meaning of the utterance) is

(2) $\|\mathbf{see}(i, \mathbf{walk}(\mathbf{john}))\|^{t_{\mathbf{now}}, martin, w, \ldots, g, \ldots}$

Where the indices give all the information we need, including the utterance time (and place) ($t_{\mathbf{now}}$), the speaker (me), the world, some way of giving values to variables (an assignment g), and anything else you might need to distinguish this utterance from any other utterance resulting from the same sentence.

The interpretation of a sentence ϕ is a function $\|\phi\| : \mathbf{V} \to \Omega$ that gives a truth value for every vector of indices. We want this vector to give enough information to distinguish one utterance from every other. We need this, not only from some esotheric philosophical point of view, but also because earlier utterances are a major component of the context of interpretation of later sentences. The traditional way that indices are carried over in the recursive definition of the interpretation, of which the definition of the existential quantifier is probably the simplest example

(3) $\|\exists x\phi\|^{g} \;:=\; \exists d\|\phi\|^{g[x:=d]}$

works well when defining, as we do here, the compositional construction of a formula out of its parts. But it is not very suited for the open ended conjunction of utterances. The change goes "into" the formula, but doesn't "stay active" after the calculation.

Dynamic Semantics was developed[1] to make this view of utterances compatible with the observation that at least part of the context-vector is dependent on earlier utterances.

[1] As with any rational reconstruction, the real, historical, reasons are quite different. In the case of Dynamic Predicate Logic, for example, Groenendijk and Stokhof wanted to give a compositional, non-representational reformalization of Kamp's DRT.

This works as follows. Let V be a context vector (with possible contributions from earlier utterances). Then $\|\phi\|^V$ is defined as before. To this we add a function that "adds" the contribution of the utterance context ($[\![\phi]\!]^V$) ($[\![\phi]\!]^V(V)$). to the vector V: $[\![\phi]\!](V)$. The next two steps contain the crucial observations. A sentence ψ that follows the utterance $\|\phi\|^V$ is interpreted in the new context $W = [\![\phi]\!](V)$. This results in

$$(4) \quad \|\psi\|^W = \|\psi\|^{[\phi](V)}, \qquad [\![\psi]\!](W) = [\![\psi]\!]([\![\phi]\!](V)).$$

But now we observe that the truth value part ($\|\psi\|$) is contained in the second, because we can always add a truth value component to the vector.

$$(5) \quad [\![\psi]\!]^{\mathbf{ln},V\mathbf{i}} = \left\langle \alpha \cdot \|\psi\|^V, [\![\psi]\!](V) \right\rangle.$$

¿From now on it is plain sailing. The interpretation of this (dynamic) conjunction can now easily calculated by applying (5) twice:

$$(6) \quad [\![\phi \wedge \psi]\!]^{\mathbf{ln},V\mathbf{i}} = \left\langle \alpha \cdot \|\phi\|^V \cdot \|\psi\|^{[\phi](V)}, [\![\psi]\!]([\![\phi]\!](V)) \right\rangle.$$

Note that although this is a theory about utterances, not sentences, the way composite utterances combine is the same for all contexts. In other words, the conjunction can be defined as a function on the sentences. This probably explains the confusion that sometimes arise. It is sometimes suggested that dynamic semantics is about sentences, not utterances. We hope that the preceding discussion resolved this confusion. Dynamic Semantics is a theory about utterances, although it is a fortunate side effect of the formalism that we only need to talk about relations between sentences for our calculations.

A large number of components of the context vector can be changed by utterances. The point of reference might be changed by temporal adverbs or by modal verbs (modal subordination changes the current world). Two major examples of dynamic semantics are Update Semantics, which deals with pragmatic aspects of the utterance and acts on the world indices, and Dynamic Predicate Logic, which deals with internal reference and acts on the assignment component. We will from now on concentrate on the latter. We will omit all components of the vector except the assignment to make the formulas not more cluttered up than they already are. Because the output might not be unique, we will interpret formulas as relations between assignments.

Dynamic Predicate Logics

In dynamic predicate logics, formulas are relations between assignments (for an assignment as input, they are true for output asignments that result from the formula being true). As shown by the discussion in the previous section, formulas have both a truth value and a dynamic effect that changes the context of interpretation. These two are at least partially independent. A formula ϕ is true for a given input g if there is an output that witnesses this: $\exists k(g[\phi]k) = \top$. It is false if there is nothing that makes it true (and, an extra demand needed below, something that makes it false).

in DPL the treatment of predicates causes a problem that perculates up through the whole of the logic. It does not allow for a negation that simply exchanges truth values (i.e. the complement). In DPL a destructive negation is defined that does not allow for dynamic binding from within its scope. By giving a definition in three-valued logic and using strict definitions for the logical constants, we arive at a theory that is slightly more intuitive, and does allow for a less destructive negation.

Definition 1 (DPL*) *Partial Dynamic Predicate Logic (DPL^) has a similar syntax to the original DPL (the negation is written as $-$) with new operators ($!$, $+$ and $\cap$) the semantics:*

$$(7)\quad g[P(x_1,\ldots,x_n)]h \begin{cases} = \star & \text{if } g \neq h \\ = \top & \text{if } g = h \text{ and } |P(x_1,\ldots,x_n|^g = \top \\ = \bot & \text{if } g = h \text{ and } |P(x_1,\ldots,x_n|^g = \bot \end{cases}$$

$$(8)\quad g[\phi \wedge \psi]h \begin{cases} = \star & \text{if } \forall k(g[\phi]k = \star \text{ or } k[\psi]h = \star\,) \\ = \top & \text{if } \exists k(g[\phi]k = \top \text{ and } k[\psi]h = \top) \\ = \bot & \text{otherwise (cf. footnote 2)} \end{cases}$$

$$(9)\quad g[\varepsilon_x]h \begin{cases} = \star & \text{if } g \not\approx_x h \\ = \top & \text{if } g \approx_x h \end{cases}$$

$$(10)\quad g[-\phi]h \begin{cases} = \star & \text{if } g[\phi]h = \star \\ = \top & \text{if } g[\phi]h = \bot \\ = \bot & \text{if } g[\phi]h = \top \end{cases}$$

$$(11)\quad g[!\phi]h \begin{cases} = \star & \text{if } g \neq h \text{ or } \forall k g[\phi]k = \star \\ = \top & \text{if } g = h \text{ and } \exists k g[\phi]k = \top \\ = \bot & \text{otherwise} \end{cases}$$

$$(12)\quad g[+\phi]h \begin{cases} = \star & \text{if } g[\phi]k \neq \top \\ = \top & \text{if } g[\phi]k = \top \end{cases}$$

$$(13)\quad g[\phi \cap \psi]h \begin{cases} = \star & \text{if } g[\phi]h = \star \text{ or } g[\psi]h = \star \\ = \top & \text{if } g[\phi]h = \top \text{ and } g[\psi]h = \top \\ = \bot & \text{otherwise} \end{cases}$$

Most of this definition is straightforward[2], except maybe for the new operators $+$, the presupposition operator, and $\cap$, the static conjunction. the presuposition operator is so called because $-(+\phi \wedge \psi) = (+\phi \wedge -\psi)$. It can be used to force the truth of a formula whether or not it is embedded under one or more negations. The static form of conjunction could just as well have been defined in DPL, but was omitted since it was not relevant for the issues at hand. It can be used to express equivalence:

[2]It may not be immediately obvious what the falsity conditions of the conjunction are. $g[\phi \wedge \psi]h$ is false when two conditions hold. First there is the condition which is the same as in 2-valued logic: $\forall k(g[\phi]k \neq \top$ and $k[\psi]h \neq \top)$. Second, there is a condition to make sure that both terms are defined. Either $\exists k(g[\phi]k = \top$ and $k[\psi]h = \bot)$ or $\exists k(g[\phi]k = \bot$ and $k[\psi]h = \top)$ or $\exists k(g[\phi]k = \bot$ and $k[\psi]h = \bot)$.

(14) $\phi \leftrightarrow \psi = (\phi \rightarrow \psi) \cap (\phi \leftarrow \psi)$.

We can also define disjunctions and a *definedness* operator.

(15) $\phi \vee \psi = \neg(\neg\phi \wedge \neg\psi)$

(16) $\phi \cup \psi = \neg(\neg\phi \cap \neg\psi)$

(17) $\mathcal{D}\phi = \phi \cup \neg\phi$

With these operators at hand, we have a logic that can express dynamic analogues to all static logic operators (In a way, $-, +, \wedge, \cap$ are the only operators that can be reasonably defined independently of the structure of the states).
Note that closure, defined so easily in DPL by the double negation, needs an explicit definition in DPL$^{\wedge}$. We can use it to define a static negation $\sim\phi := -!\phi$ in DPL$^{\wedge}$ that mirrors the effects of the normal negation ($\sim$) in in DPL. Written out the definition of $\sim$ is:

$$
\begin{aligned}
(18)\quad g[\sim\phi]h &= \top && \text{if } g = h \text{ and } \neg\exists k\ g[\phi]k = \top \\
&= \bot && \text{if } g = h \text{ and } \exists k\ g[\phi]k = \top \\
&= \star && \text{otherwise}
\end{aligned}
$$

If we would use $-$ as our standard negation we would be in trouble because $-(\varepsilon_x \wedge \phi)$ and $(\varepsilon_x \wedge -\phi)$ are equivalent. Instead we use the definition of $\sim$ and make *that* dynamic. We do this by replacing the two occurances of $g = h$ by a condition that the dynamic negation of the formula holds.

$$
\begin{aligned}
(19)\quad g[\neg\phi]h &= \top && \text{if } g[-\phi]h = \top \text{ and } \neg\exists k\ g[\phi]k = \top \\
&= \bot && \text{if } g[-\phi]h = \bot \\
&= \star && \text{otherwise}
\end{aligned}
$$

This results in a negation (the strong dynamic negation) that has the truth values of $\sim$ and the dynamic effects if $-$. Note that $\neg\phi = + - (!\phi \wedge -\phi) \cap -\phi$.

Concluding Remarks

The logic in this note is a very simple dynamic predicate logic. If we want to make this into something that is useful for linguistics we have to give some way of dealing with plurals and generalized quantifiers (cf. [vdBerg 93a,b]). Furthmore, logics only formalize the *use* of discourse referents, not their *resolution*. For this, we have to make it the semantical component of a discourse grammar that implements the resolution (cf. [Prüst 91a,b], which gives an example of the sort of grammar we have in mind. This grammar has a static logic as semantic component, but gives a good idea of how a discourse grammar looks.).

References

[vdBerg93a] Berg M. H. van den, 1993b, 'Dynamic Generalized Quantifiers', in: van der Does J. and J. v. Eijck, 1991, *Generalized Quantifier Theory and Applications*, Dutch Network for Language, Logic and Information. To appear (1993?) in CSLI lecture notes.

[vdBerg 93b] Berg M. H. van den, 1993a, 'Full Dynamic Plural Logic', in: Bimbo K. and A. Mate (eds), 1993, *Proceedings of the Fourth Symposium on Logic and Language*, Budapest.

[Gabbay & Kempson 92] Gabbay D. and R. Kemson, 1992, 'Natural-Language Content and Information Flow: A Proof-Theoretic Perspective', manuscript.

[Groenendijk & Stokhof 91] Groenendijk J. and M. Stokhof, 1991, *Dynamic Predicate Logic*, Linguistics and Philosophy **14**, pp. 39–100.

[Prüst 91a] Prüst H. and M.H. v.d. Berg, 1991, 'A Coherent Approach to Underspecification in Natural Language Discourse', in: Caenepeel M. et.al. 1991, *Workshop on Discourse Coherence*, University of Edinburgh.

[Prüst 91b] Prüst H., M.H. v.d. Berg and R. Scha, 1991, *A Formal Discourse Grammar tackling Verb Phrase Anaphora*, ITLI-prepublication series CL-91-03, ILLC, Amsterdam. To appear in **L&P**.

Using a Computational Situation Theoretic Language to investigate Contemporary Semantic Theories

Alan W Black

Centre for Cognitive Science, University of Edinburgh

2 Buccleuch Place, Edinburgh EH1 9NW

awb@ed.ac.uk

Abstract of talk given at
Dagstuhl Workshop on Semantic Formalisms

DRAFT – March 6,1993

Abstract

One of the problems with contemporary semantic theories of natural language, such as Situation Semantics, Discourse Representation Theory and Dynamic Semantics, is that although they address similar semantic phenomena their widely differing syntax and semantics make it difficult to compare their treatments of similar natural language semantic phenomena. It would be useful to describe these different semantic theories in a common framework so they could be more easily compared, and in the long run allow cross-pollination of their various treatments.

In this extended abstract I will give a brief overview of a language called ASTL. ASTL is based on fundamental aspects of situation theory. It is designed to be used as a meta-language for describing aspects of natural language semantic theories, offering a common framework for the description of semantic theories. Because ASTL has an implementation, descriptions in ASTL can be used directly to derive semantic translations of the theories they describe. Also because ASTL is given a situation theoretic semantics it offers a semantics for any description written in it, which allows closer comparison of treatments in a theory.

Apart from the advantages of close comparison of theories (e.g. DRT and Dynamic Semantics) that ASTL allows, we will also discuss what makes ASTL suitable as a framework and how it compares with other possible semantic meta-theories (e.g. Prolog, feature systems, etc.).

Acknowledgements: This work was funded by SERC and DYANA-2.

1 Motivation

Although, in the field of computational semantics, we have the same ultimate goal of providing a computational treatment of natural language semantics, many different theories have been proposed as possible solutions. These theories although often addressing the same semantic phenomena may have apparently quite different solutions. The difference in syntax and semantics of theories can be clearly illustrated just by looking at three semantic translations of the same simple sentence. Given the simple sentence *"A man walks"* three contemporary theories would give something like the following translations.

Situation Semantics	$S \models \ll man, X; 1 \gg$
	$S \models \ll walk, X; 1 \gg$
Dynamic Logic	$\exists x\ [man(x)] \wedge walk(x)$

DRT

```
+----------+
|    X     |
+----------+
|          |
|  man(X)  |
|  walk(X) |
|          |
+----------+
```

Note that only the dynamic logic representation has an explicit existential quantifier but even it differs from standard first order logic. The semantics of dynamic logic is such that the x in $walk(x)$ actually does fall within the scope of the preceding existential.

Given the above three apparently different treatments it would be useful if we could offer a framework in which they could be described such that a close comparison could more easily be made. This is exactly the work that is more fully presented in [Black 93].

There is an analogy here with what happened around ten years ago in the field of computational syntax. PATR-II ([Shieber 84]) offered a language in which contemporary syntactic theories could be specified. PATR-II descriptions could then be directly executed. PATR-II was not proposed as a syntactic theory itself but only as a mechanism in which theories could be described. Even though syntactic theories were never fully defined in PATR-II, descriptions of key aspects of theories lead to a greater understanding of them and later theories of natural language syntax have benefited from the close comparison that was possible in PATR-II.

Here we will give a brief overview of the computational situation theoretic language ASTL which is designed to allow executable specifications of aspects of natural language semantic theories such as Situation Semantics, Discourse Representation Theory and Dynamic Logic. The specification of a theory is sufficient to allow translations to be given to natural language utterances. Furthermore because such descriptions are in the same framework a much closer comparison of the semantic treatments is made possible.

2 ASTL

ASTL is designed as a computational language which will act in some way like a meta-language suitable for describing aspects of natural language semantic theories. ASTL is based on fundamental aspects of situation theory and is given a formal syntax and semantics within a situation theoretic model. ASTL offers a representation for: individuals, parameters, variables, relations, facts, situations, situation types and constraints. In addition to these static objects a set of inference rules (and interpreter) is provided in order that new propositions may be derived from a basic description of situations and constraints. The main purpose of the language is to allow a semantic theory to be described, such that given an utterance it is possible to derive its semantic translation.

Although language processing (particularly deriving semantic translations from utterances) is an important aspect of ASTL, we do not want to exclude the possibility of using the translations derived from the description of a theory within ASTL for more general inference. Although that has not been exploited in [Black 93] it does seem to be a reasonable possibility.

Rather than present the formal aspects of the language, the following simple example will help illustrate the basic operation of the language.

```
Individuals    {h,t}
Relations      {happy/1, smiles/1}
Parameters     {S}
Variables      {*S, *T, *Y}
Situations
  ;; Basic situations -- i.e. who smiles where
  (SIT1 :: [S ! S != <<smiles,h,1>>
                S != <<smiles,t,1>>]
   SIT2 :: [S ! S != <<smiles,t,1>>] )
Constraints
  ;; if they smile they are happy
  *S : [S ! S != <<happy,*Y,1>>]
    <=
     *S : [S ! S != <<smiles,*Y,1>>].
```

Variables in ASTL are conventionally written as atoms prefixed with * (although the initial declarations actually define which atoms are variables, individuals, etc.). In the above ASTL description we have defined two basic situations, (SIT1 and SIT2). In SIT1 we know two facts, that h ("Hanako") smiles and that t ("Taro") smiles while in SIT2 we only know that t ("Taro") smiles. In general constraints, which are between situations, state that in all ways that the right hand side of the constraint is true the left hand side is true also. So in this particular example we are stating that if something smiles in a situation then it is also happy, in that same situation. The variable *S at each side of the <= operator ensures that we are talking about the very same situation.

Given the above simple description we can ask simple queries about the system of situations and constraint. For example we can ask.

```
astl> query *T:[S ! S != <<happy,t,1>>].
```

This can be glossed as: is there a situation in which t ("Taro") is happy? The implementation rather than simply answering "yes" or "unknown" will display any situations for which the query proposition is true thus the answer will be given as

```
SIT1
  happy(h)
  smiles(h)
  happy(t)
  smiles(t)
```

```
SIT2
  happy(t)
  smiles(t)
```

The output in ASTL is typically shown as boxes based on the situation theoretic notations of EKN ([Barwise & Cooper 93]). This box notation which is different from ASTL's basic linear form gives a much better display of what a situation supports, especially when a situation supports facts that have situations as arguments.

A major aspect which distinguishes ASTL from a simple logic programming language is that ASTL treats situations as first class objects (as is the case in situation theory). Situations can be used as arguments to facts. Self-reference is not a problem. The use of situations as objects allows inference to be local to situations. It also allows situations not mentioned in the initial declaration to be derived thus offering a powerful descriptive language.

Using ideas from Situation Theoretic Grammar (STG) [Cooper 89] we can use the basic aspects of ASTL to offer a framework for natural language syntax and hence lead to a method for describing the construction of natural language semantic translations for utterances. If we take the general view that the utterance of a piece of natural language can be represented by a situation we can easily use ASTL to represent the syntax of utterances. More specifically when I utter the noun phrase *"Hanako"* we can talk about the situation in which I uttered it. It has a time and a place. We can also attribute various syntactic properties to that situation. For example

"Hanako" $\leadsto$

```
Sit123
  cat(SIT123,ProperNoun)
  use_of(SIT123,"Hanako")
```

Given this treatment of utterances if such a situation is "next to" one in which I say *"walks"* we can infer that there is also a sentence situation covering the two words. In ASTL terms we can capture this with a constraint of the form

```
*S:[S ! S != <<cat,S,Sentence,1>>
        S != <<start,S,*Start,1>>
        S != <<end,S,*End,1>>]
<=
    *NP:[NP ! NP != <<cat,NP,NounPhrase,1>>
              NP != <<start,NP,*Start,1>>
              NP != <<end,NP,*Mid,1>>],
    *VP:[VP ! VP != <<cat,VP,VerbPhrase,1>>
```

```
VP != <<start,NP,*Mid,1>>
VP != <<end,NP,*End,1>>].
```

Because one of ASTL's major uses is in language processing we have included a special form of constraint specifically to deal with grammar rules, and hence the start and end point need not be made explicit in a special grammar rule form of constraint. However this special form may be completely defined in terms of the basic ASTL constraints—this is analogous to the relationship between Prolog's Definite Clause Grammar (DCG) rules and basic Prolog rules.

3 Using ASTL for describing theories

As the ultimate goal of computational semantic theories is to provide a computational treatment of the semantics of natural language it seems useful if not necessary that theories are specified in enough detail so that they can actually be used. Descriptions in ASTL are intended to be executable specifications of theories. The idea being that a description contains a formalisation of a semantic theory that is specific enough to allow the construction of semantic translations of natural language utterances.

In [Black 93] three such descriptions are given: Situation Theoretic Grammar [Cooper 89], Discourse Representation Theory [Kamp 81], and a form of dynamic semantics (based on DPL [Groenendijk & Stokhof 91]). Each of these descriptions are given with respect to the same language fragment which is sufficient for simple sentences and discourses including inter-sentential anaphora and donkey anaphora.

It should be added that ASTL is not a general implementation language for large natural language processing systems. It may develop that way in the future but the current implementation is only really suitable for smaller investigations of contemporary semantic theories. The present work has concentrated on both the representation of the theories' semantic translation and the construction of that form. As yet little work has been done on drawing inferences from the generated semantic translation of utterances, although ASTL does seem to have the basic capabilities to act in that capacity.

One possible representation for DRSs in ASTL is as parametric situation types. That is a DRS is effectively the type of a situation where it is true—with respect to a set of anchors for each free parameter (i.e. discourse marker), and of course appropriate conditions for quantifiers. DRSs are threaded through the basic syntactic structure in a way similar to the threading in the DRT description in [Johnson & Klein 86]. Given a set of words that are to be uttered and the description of DRT in ASTL the implementation can derive the syntactic structure of the utterance as the DRSs at each stage in the discourse. The outgoing DRSs for *"a man walks"* is

```
+-----------+
|    P1     |
+----+------+
| P1 |      |
+----+      |
|           |
|  man(X)   |
|  walk(X)  |
|           |
+-----------+
```

Notably there is no explicit domain in the representation of the DRS. A useful consequence of the ASTL representation of DRSs is that it is a (parametric) situation type and has a natural interpretation within the model. This shows that key aspects of DRT can be encoded within a situation theoretic model and that the encoding is in no way contrived.

4 Properties of ASTL

ASTL has sufficient expressive power for describing at least the core aspects of the three semantic theories mentioned above. It is interesting to step back and try to identify which properties of ASTL make it suitable as a meta-language for the formal specification of executable specifications of semantic theories. We can identify three main aspects:

The ability to represent complex structured objects and allow general relations between them.

The ability to have constraints between general objects and draw inferences from them.

A mechanism for reasoning about "variables" in the object language (as distinct from variables in the meta-language) and be able to describe binding mechanisms for these object language variables.

There is perhaps another property which is discussed as a possible extension to ASTL which is not currently included. That property is some form of abstraction, application and reduction which seems to be a basic property of some natural language semantic theories. However simple aspects of this can be modelled in the current version of ASTL.

With respect to situation theory the above three properties are core aspects of the theory. We can identify each of the properties with well accepted key parts of situation theory:

- situation, abstractions and types
- constraints
- parameters and anchoring

However we do not say that situation theory is therefore the only possible base for a general semantic meta-language. The above aspects are also found in other frameworks. It seems quite possible to define an attribute logic (of the general form as described in, for example [Johnson 88]) which has exactly those properties. Feature logics are very expressive and a wide selection of constructs are available which can be included in a logic. But arguably we could say that even if such a logic were defined it seems reasonable to claim that any implementation of that logic would equally be an implementation of ASTL as it would have exactly those properties which are fundamental to ASTL. We could also build such properties within Prolog. Although Prolog has only one "situation" it is easy to encode structured objects like situations in Prolog in fact some other logic programming languages already do this. Another area where we might also find such properties is the framework of object-oriented programming currently popular in computer science. The conclusion we wish to draw from this is that the essential properties of ASTL as stated above are not uncommon in other frameworks but as they are fundamental properties of situation theory it seems not unreasonable to present them within a situation theoretic framework. Also because we are working in the area of computational semantics it seems good that we can find the abstract properties we require in an already existing semantic theory.

5 Conclusion

ASTL offers both a method for formalisation of key aspects of contemporary semantic theories and because ASTL is a computational systems the formalisations in ASTL are directly executable. Once theories have been described in ASTL much closer comparisons can be made. The descriptions of DRT and dynamic semantics in ASTL given in [Black 93] seems to show that these theories differ only at some low level information content of expressions rather than any predictive or functional level. Also once descriptions are given in the same framework we can start to re-use treatments in one theory within another (e.g. adding a DRT like treatment of pronouns to a simple situation semantic treatment of language). This second aspect depends on there being at least some basic commonality between the theories but as we are interested in semantic theories that are describing similar phenomena this will often be the case.

In conclusion ASTL is a simple tool which does seem to help solve our original problem—how can we compare contemporary theories when they are presented with such different syntax and semantics. Although there is lots more to be done, ASTL seems to be a good start.

References

[Barwise & Cooper 93] J. Barwise and R. Cooper. Extended Kamp Notation: a graphical notation for situation theory. In *Situation Theory and its Applications, III*, CSLI Lecture Notes. Chicago University Press, forthcoming 1993.

[Black 93] A.W. Black. *A Situation Theoretic approach to computational semantics.* Unpublished PhD thesis, University of Edinburgh, Edinburgh, UK., 1993.

[Cooper 89] R. Cooper. Information and grammar. Technical Report RP No. 438, Dept of Artificial Intelligence, University of Edinburgh, 1989. Also to appear in J. Wedekind, ed. *Proceedings of the Titisee Conference on Unification and Grammar.*

[Groenendijk & Stokhof 91] J. Groenendijk and M. Stokhof. Dynamic Predicate Logic. *Linguistics and Philosophy*, 14:39–100, 1991.

[Johnson & Klein 86] M. Johnson and E. Klein. Discourse, anaphora and parsing. In *Proceedings of the 11th International Conference on Com-*

putational Linguistics, pages 669–675, Bonn, West Germany, 1986.

[Johnson 88] M. Johnson. *Attribute-value logic and the theory of grammar.* CSLI Lecture Notes Number 16. Chicago University Press, 1988.

[Kamp 81] H. Kamp. A theory of truth and semantic representation. In J. Groenendijk, T. Janssen, and M. Stokhof, editors, *Formal Methods in the Study of Language.* Mathematical Center, Amsterdam, 1981.

[Shieber 84] S. Shieber. The design of a computer language for linguistic information. In *Proceedings of the 10th International Conference on Computational Linguistics/22nd Annual Conference of the Association for Computational Linguistics*, pages 362–366, Stanford University, California, 1984.

Towards a general semantic framework*
ABSTRACT
Prepared for the Dagstuhl Seminar "Semantic Formalisms in Natural Lanugage Processing" 22nd-26th February, 1993

Robin Cooper

Centre for Cognitive Science
and
Human Communication Research Centre

University of Edinburgh

0 Introduction

I believe that if we are to develop general semantic formalisms for use in computational semantics, we should only do this on the basis of a deep understanding of the relationships between the various semantic theories on offer. In fact, I think it is more important that we develop a general semantic framework with associated algorithms and computational techniques which offers the opportunity to define reusable theory independent semantic modules than to define a formalism as such. There may indeed be a single canonical formalism but it may also be the case that there is a family of formalisms all based on the same framework which are suitable for different applications. In this paper I shall offer a few introductory remarks on what such a semantic framework might be and then focus on a contribution that might be made by one mathematical tool recently developed by Peter Aczel and Rachel Lunnon.

*Work on this paper was supported in part by Esprit BRA 6852, DYANA

1 What is a semantic framework?

A semantic framework should be a formal system which abstracts core notions which are shared by the various approaches to natural language semantics which are on offer. At the same time the framework should be rich enough to specify the particular proposals of the individual approaches so that it becomes clear in formal terms exactly where the approaches differ. The computational advantages of such a framework are that it enables the implementation of modules relating to the core notions, perhaps parameterized in order to account for differences in the different approaches. This means that there will be some general computational techniques available which can be used no matter which theory you choose to base your implementation on. Also, it means that implementations need not be restricted to one particular theory. It should give us the opportunity to mix and match analyses cast within various approaches. The fact that everything is embedded in a general framework should enable us determine which analyses from the different approaches are compatible.

There are three strategies for creating such a framework which might be pursued and I feel that they should ultimately complement each other:

semantic operators This is the approach introduced by Johnson and Kay (1990). It involves abstracting out various operators such as APPLY, NEW_INDEX, CONJOIN and interpreting these operators differently for different semantic theories such as Montague semantics, discourse representation theory and situation semantics.

formal specification This would involve the application of specification techniques developed in computer science as described, for example, in work on algebraic specification (Sannella and Tarlecki, 1992) and in research on the Logical Framework (Harper, Honsell and Plotkin, 1987).

semantic metatheory This involves the development of a general theory in which the various individual semantic theories can be cast.

In this paper I want to develop an example relating to the last of these options, since I feel that making progress on this front would contribute to the aims of the other two strategies.

2 Aczel-Lunnon abstraction in situation theory

I am going to concentrate on the notion of abstraction and try to argue that the particular kind of abstraction proposed by Aczel and Lunnon (1991) and Lunnon (1991) in connection with situation theory gives us a view of how Montague's semantics and discourse representation theory could be comprehended within a single metatheory which makes the two approaches interact and blend in an interesting way. For the sake of concreteness, I will use situation theory as the metatheory. While this gives us the additional advantage of drawing some parallels with situation semantics, I think it should become clear that Aczel-Lunnon abstraction is independent of situation theory[1] and that ultimately one may wish to consider a general semantic metatheory which is more general than situation theory but includes Aczel-Lunnon abstraction. Aczel and Lunnon developed their notion of abstraction originally in order to provide a model of abstraction as it had been described in informal accounts of situation theory. The idea is that abstracts are a particular kind of object in a structured universe. This contrasts with the standard[2] view of abstracts received from Montague's semantics as functions or rather λ-expressions which are interpreted as functions. While this will be important for what we are going to do there are two other features of this kind of abstraction which I would like to emphasize:

simultaneous abstraction Any number of parameters in a parametric object may be abstracted over simultaneously. While in standard λ-notations one may have expressions such as

$$\lambda x, y, z[\phi(x, y, z)]$$

this is to be construed as an abbreviation for

$$\lambda x[\lambda y[\lambda z[\phi(x, y, z)]]]$$

In Aczel-Lunnon abstraction, however, it is the set which is abstracted over. Thus arguments to the abstract can be supplied simultaneously and there is no required order.

indexing This feature is closely related to the previous one. Since abstraction over parameters results in an object in which those parameters do not occur[3], we have to have some way of determining how arguments are to be assigned to the abstract in the case where more than one parameter has been abstracted over. Aczel and Lunnon achieve this by defining the abstraction operation in terms of indexed sets of parameters, i.e. one-one mappings from some domain ("the indices") to the parameters being abstracted over. An important aspect of this for us is that we can use any objects in the universe as the indices.

In the remaining sections of this paper I shall sketch the following results:

[1]Certainly current presentations of Aczel-Lunnon abstraction are not in terms of situation theory but generalized set theory, set theory with the addition of abstracts as non-sets. See Lunnon(1992).

[2]at least for formal semanticists

[3]This is important in order to achieve α-equivalence, i.e. $\lambda x[\phi(x)] = \lambda y[\phi(y)]$

- Montague's combinatory techniques using abstraction can be recaptured using Aczel-Lunnon abstraction in a straightforward and unsurprising way since Montague only requires unary rather than simultaneous abstraction.

- Embedding this is situation theory gives us an interesting perspective on partial Montague grammar and the problem of identity of logically equivalent propositions. Essentially it shows that it is hard to recreate this classical problem in a structured universe even if you think of propositions as abstracts over possible worlds rather than situations.

- We can use Aczel-Lunnon abstraction to model Kamp's discourse representation structures by reconstructing them as predicates and exploiting the fact that we have simultaneous abstraction with indexing. This gives us an interesting perspective on discourse representation theory and allows its integration with Montague's combinatory techniques.

- The feature of non-selective binding so important to classical discourse representation theory is captured by introducing quantification over simultaneous abstracts, in a manner similar to the introduction of quantification in the standard λ-calculus.

- Discourse anaphora can be achieved by exploiting the fact that arbitrary indices are used in the abstracts. While the parameters are bound within an abstract the indices are freely available and can be used to encode discourse anaphoric relations. A predicate (DRS.) corresponding to the discourse so far can be combined with a predicate corresponding to the current sentence in a way that merges the roles in the two abstracts which have the same index.

In the full paper I will give an outline of the techniques used to achieve this.

References

Aczel, Peter and Rachel Lunnon (1991) Universes and Parameters, in *Situation Theory and its Applications, Vol. 2* ed. by Jon Barwise, Jean Mark Gawron, Gordon Plotkin and Syun Tutiya, CSLI

Barwise, Jon and Robin Cooper (1991) Simple Situation Theory and its Graphical Representation, DYANA deliverable R2.1C, *Partial and Dynamic Semantics III*, ed. by Jerry Seligman.

Barwise, Jon and Robin Cooper (in press) Extended Kamp Notation: a Graphical Notation for Situation Theory, in *Situation Theory and its Applications, Vol. 3*, ed. by Peter Aczel, David Israel, Yasuhiro Katagiri and Stanley Peters, Stanford: CSLI

Black, Alan (1992) Embedding DRT in a Situation Theoretic Framework, *Proceedings of the fifteenth International Conference on Computational Linguistics*, COLING-92, pp. 1116–1120

Black, Alan (forthcoming) *A Situation Theoretic Approach to Computational Semantics*, PhD thesis, Department of AI, University of Edinburgh.

Harper, Robert, Furio Honsell and Gordon Plotkin (1987) A Framework for Defining Logics, in *Symposium on Logic in Computer Science*, IEEE.

Johnson, Mark and Martin Kay (1990): *Semantic Abstraction and Anaphora*, Proceedings of COLING 90, Helsinki.

Kamp, Hans (1990) Prolegomena to a Structural Theory of Belief and Other Attitudes, in *Propositional Attitudes: The Role of Content in Logic, Language and Mind*, ed. by C. Anthony An derson and Joseph Owens, CSLI.

Kamp, Hans and Uwe Reyle (forthcoming) *From Discourse to Logic: Introduction to Model Theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*, Dordrecht: Kluwer

Lunnon, Rachel (1991) *Generalized Universes*, Ph.D. thesis, University of Manchester

Lunnon, Rachel (1992) Course notes for LLI Summer School.

Muskens, Reinhard (1989) *Meaning and Partiality*, PhD thesis, University of Amsterdam

Pinkal, Manfred (1991): *On the Syntactic-Semantic Analysis of Bound Anaphora*, Proceedings of the 5th EACL, Berlin

Sannella, Donald and Andrzej Tarlecki (1992) Toward Formal Development of Programs from Algebraic Specifications: Model-theoretic Foundations, *Proceedings of the Nineteenth International Colloquium on Automata, Languages and programming*, Lecture Notes in Computer Science, Springer Verlag.

Experiences with the Framework of Situation Schemata

Jens Erik Fenstad
Department of Mathematics, University of Oslo

On one view of natural language comprehension one recognizes two modules, the computational (i.e. Grammatical Space) and the conceptual (i.e. Semantic Space). A candidate for the representational link between the two is an algebraic theory of signs (in particular, lexical signs, but also, more generally, phrasal signs) in the form of a suitable class of (possibly typed) feature structures (examples: LFG, Situation Schemata, HPSG; theory: unification systems).

Much is known about possible structures for grammatical space; less attention has been paid to the structure of semantic space (but there is much insight to be derived from applied systems building). In the lecture I first discuss some standard approaches based on (partial) model theory and review some "experiences" such as the question-answering systems developed by Espen Vestre, the translation system PONS studied by Helge Dyvik, and the theory of locative prepositional phrases developed by Erik Colban. This work is based on the particular relational theory of meaning for situation schemata developed in Fenstad et al., *Situations, Language and Logic* (1987). As a language for semantic representation, use wasalso made of the particular logic system L3, which is a two-sorted first order language based on the partial format

in s: at l: r, a_1, a_2, ..., a_n; i

of situation semantics.

In all of these examples semantic space is roughly equated with some theory of data bases. And, one may argue, the partial logic L3 may be of interest in connection with the theory of deductive databases. But the theory of semantic spyce needs more structure, and we discuss in the lecture the notion of Conceptual Space (P. Gärdenfors), where the ideas of distance and convexity play an important role (natural kinds, prototype theory). In this connection we also discuss the limits of "algorithmic Logic" and pointed out the need for more "geometric modes of reasoning" (e.g. in the style of the theory of Mental Spaces by Johnson-Laird, and in the work of Barwise and Etchemendy).

Towards the end of the lecture we speculate a bit on how semantic space can be seen as the "phase space" of an underlying dynamics, which would be a theory of how concepts emerge and which also could account for certain types of non- monotonic reasoning in connection with lexical signs (see P. Smolensky and P. Gärdenfors). This ties in with different processing paradigms, e.g. the recent work of Kempen and Vosse on incremental syntax tree formation (note in this connection the "compatibility" with Dennett's metaphor of "multiple drafts" and Kosslyn's work on sentence monitoring and programming).

Situated Dialog

Hideyuki Nakashima*and Yasunari Harada†

Abstract

When we communicate through (natural) language, we do not explicitly say everything. Rather, both the speaker and the hearer utilize information supported by the utterance situation, which includes the mental states of the speaker and the hearer. Extreme and interesting cases are observed in the use of Japanese (in dialogue situations). Syntactic (or configurational) constraint of Japanese is weaker than those of English, in the sense that the speaker may omit almost any element in a sentence.

In this paper we present a model of the hearer in the light of situated reasoning and show how the missing information can be supplied from the situation.

1 Introduction

When we communicate through (natural) language, we do not explicitly say everything. Rather, both the speaker and the hearer utilize information supported by the utterance situation. For example, if we say "it is four o'clock" in Japan, it means "it is four o'clock in Japan". "In Japan" part is not made explicit in the utterance because it is obvious from the situation. More extreme and interesting cases are observed in the use of Japanese (in dialogue situations). Syntactic (or configurational) constraint of Japanese is weaker than those of English, in the sense that the speaker may omit almost any element in a sentence.

It is not correct to say that the information that are not made explicit are *omitted* because they are obvious. In many cases, we are not even aware that those pieces of information are important part of our communication. Our thought/inference itself is situated and thus makes certain aspect of information concealed from the outside manipulation and this fact is reflected in our use of language.

Situation Semantics[BP83] formalizes the relation between the situation (called discourse situation or *DU*) and the meaning of the utterance. However, its main purpose is to describe mappings from uttered elements of sentences and their "meanings". The formalism cannot be used to describe, for example, "how the listener identified the referent".

We will extend the formalism of Situation Semantics to cover mental activities of the speaker and the listener.

*Electrotechnical Laboratories

†CSLI. On leave from Waseda University.

2 Situated Inference

If we are in a situation about which we are reasoning, we should have some, possibly partial, information about the situation. We further assume that the information is represented as a set of infons[Dev91] about the situation. Infons are used to represent some properties of the situation.

A situation is characterized by the infons which the situation supports. Note that it is impossible to fully describe an actual situation by a finite number of infons. We will rather treat situation types each of which is defined by a finite set of infons. For more details of the formalism, see [NT91] and [NOK91].

Let us consider the proposition "birds fly" as an example of inference using an abstract situation. We propose to write this as information about an abstract situation, bird as follows:

$$bird \models \langle\langle \mathrm{fly} \rangle\rangle.$$

The above expression expresses that the infon ⟨⟨fly⟩⟩ holds in the situation *bird*. To understand it, one should imagine an abstract situation in which only the properties of birds are in question. In those situations, since any infon is about birds, there is no need to explicitly state it. In this paper we assume that any n-ary relation on birds can be represented as an n-1 -ary relation.

3 Example Dialogs

Since Japanese is a pro-drop language, we encounter numerous cases of subjectless sentences in naturally occurring conversations. But the speakers of Japanese usually find little difficulty in interpreting those utterances appropriately, although in certain circumstances, misunderstandings do occur. Identification of intended subject, however, is not a simple matter.

For instance, if you utter a one word assertion "Hungry, " you would very likely be understood as saying that you are hungry. On the other hand, one word question such as "Hungry?" would be interpreted as asking if the hearer is hungry. Where does the different supply for the missing argument come from?

4 Our Formalism

Here is our conjecture. Usually, missing information should be supplied from the hearer/reader's situation. It is quite natural since the hearer/reader does not have full access to the speaker/writer's situation. It is easier for the hearer to obtain information from his/her own supply. Queries and orders (note that the instruction sited above is a kind of order) fall into this category.

Assertions are special and not natural in this respect. When the hearer gets an information ⟨⟨hungry⟩⟩ (s)he stores it in the model of the speaker. On the other hand, (s)he will supply "the hearer" or "me" if (s)he hears

⟨⟨red-eyed⟩⟩

because (s)he has the contradicting information that the speaker is not red-eyed. Contradictory or already existing information will be passed on to the outer situation, whitch is the hearer's. This will explain that all of the following information go into the hearer's model rather than the speaker's.

Orders such as ⟨⟨run⟩⟩ always go outside because the hearer has no direct control over what the speaker does.

Queries are handled in a similar manner. If we have the information in the speaker's model we will answer the query using it. But we will pass on to another situation if we don't have the information to answer. The alternative situation is determined by the history of the dialog. The detailed machanism follows.

There are

- The current (mental) situation, called the C(urrent)-sit, on which attention is focused.
- A list of situations, called D(efault)-sits.

If there is no other cue, C-sit is the top item of D-sits.

The procedure:

1. Linguistically obtained information is stored in C-sit.
2. Conflict checking is achieved locally on C-sit.
3. If there is no conflicts detected, then end.
4. If there is a conflict, determine which information to move.
5. The lost information is moved to the element of D-sits. The new element is set to C-sit and repeat from step 2.

5 Analysis

Because of the limitation on space, we show the result of the analysis only for "hungry" with some of the situations.

Situations:

A: Speaker looking into the mirror.

C-sit is the mirror. Since there is only one person in the mirror, being hungry is attributed to the person: SPEAKER.

B: Speaker and Hearer looking into the mirror.

C-sit is the mirror. Since there are two persons in the mirror, C-sit doesn't give the hearer enough information. Another cue must be searched. SPEAKER/HEARER.

C: Speaker and Hearer looking at each other.

The interpretation depends on the previous history of the dialog. If C-sit includes only either of the speaker and the hearer, the information is attributed to the person. If C-sit includes both or none, another cue must be used.

Since, no visual information is available in this case (or, at least let us assume so), it is unlikely that the speaker says "hungry" without any focus of attention. He must have been guiding the hearer by C-sit.

Acknowledgment

The authors thank CSLI for giving us opportunities to carry out part of the research presented here.

References

[BP83] Jon Barwise and John Perry. *Situations and Attitudes.* MIT Press, 1983.

[Dev91] Keith Devlin. *Logic and Information I: Infons and Situations.* Cambridge Univ. Press, 1991.

[NOK91] H. Nakashima, I. Ohsawa, and Y. Kinoshita. Inference with mental situations. TR-91-7, ETL, 1991.

[NT91] Hideyuki Nakashima and Syun Tutiya. Inference *in* a situation *about* situations. In *Situation Theory and its Applications, 2*, pages 215–227. CSLI, 1991.

Constraints on Unification-Based Semantics

Jan Tore Lønning
Dept. of Linguistics, University of Oslo
P.O. Box 1102, Blindern, 0317 Oslo
Norway

Unification yields an efficient tool for grammar writing. In particular, one does not have to decide which part of a sentence which contributes a certain feature, e.g., number may come from the verb, the subject noun phrase, or both. Furthermore, a unification grammar computes an output structure, a feature structure which may be used for encoding the semantics of the sentence. Unification has been claimed to be particular suitable for this purpose, not only may several parts of the utterance contribute to the same part of the semantic description, but different modules, like syntax and prosody, may contribute to the meaning and constrain each other in a mutual and non-directional way (Fenstad et. al. 1987), and "a sign need not even have an isolable component that can be identified as its interpretation," (Pollard 1989).

This may be convenient for sub-phrases of the sentence. But if the feature structure associated with the full sentence shall represent a semantic content, a *minimal constraint* must be that it is possible to decide whether the feature structure represents a semantic content, and if it does, what this content is. A further possible constraint would be that the feature structure of each well-formed sentence actually contains a representation of a semantic content, a property we will follow Halvorsen (1983) and call *completeness of the semantic interpretation.*

Most fragments in the unification-based tradition associating sentences with semantics satisfy both constraints (e.g., Fenstad et. al. 1987, Pollard and Sag, 1988), but are not based on principles which guarantee that so will be the case for extensions to the fragments. But if our aim is broad coverage and compatibility between different approaches, such principles are necessary.

Technically, the picture is this. There are three domains, L, the set of strings, F, the set of feature structures, O, the set of semantic objects. A *grammar* G is a relation on $L \times F$ and a *semantic interpretation* S is a relation on $F \times O$.

Minimal Constraint: There is a primitive recursive function
$s : F \rightarrow 2^O$ such that $S = \{(f, o) : o \in s(f)\}$.

Completeness Constraint: $Dom(S) \subseteq Range(F)$.

As a first example, we can consider Montague grammar in this picture — even though it is not a unification grammar. It is natural to consider the disambiguated syntactic structures (derivation histories) as the representations. Each such structure has a unique, well-defined, computable interpretation, i.e., both constraints are satisfied.

1 Lexical-Functional Grammar

1.1 Description by analysis

In the first approach to semantic interpretation in Lexical Functional-Grammar (LFG) (Halvorsen 1983)—later called *description by analysis* (Halvorsen and Kaplan 1988)— a string was first associated with a complete and coherent f-structure. Then a semantic interpretation was derived from the f-structure. The strategy for deriving a semantic representation σ from a string w can be summarized as a three-step procedure:

1. Try to find an f-structure f which matches w.
2. Check f for completeness and coherence.
3. Derive a semantic representation σ from f.

The minimal constraint amounts to a well-defined procedure for step 3. The procedure is complete if the third step succeeds whenever the first two succeed.

Completeness is not integral to LFG as such, but may be obtained by constraining the grammar format and the semantic interpretation. Halvorsen (1983) proposes a set of such constraints cf. Reyle 1988, too). In particular, grammatical categories and functions were associated with types, like (simplified)

$$NP \mapsto \langle\langle e,t\rangle,t\rangle$$
$$\overline{S} \mapsto p$$
$$PRED(\uparrow SUBJ)(\uparrow VCOMP) \mapsto \langle\langle\langle e,t\rangle,t\rangle,\langle p,t\rangle\rangle$$

But this presupposes a correspondence between syntactic category and syntactic function which is not always assumed in LFG. For example, if something of category $\overline{S}$ may be a subject, we may expect correspondence between category and type, but not between syntactic function and type. A verb requiring a subject of category $\overline{S}$ may have a well-defined, coherent and complete f-structure when combined with a NP-subject, but the semantic interpretation may fail.

There are two possibilities. Either admit that step 3 in the procedure will not always succeed, semantics is an additional filter on well-formedness. Or restrict the grammars such that step 3 will always succeed. The latter can be obtained by restricting the class of LFGs or by enriching the functional structures by percolating the type information down into the f-structures, e.g., under the name of theta roles.

1.2 Co-description

In later proposals (Fenstad et. al. 1987, Halvorsen and Kaplan 1988), a structure representing the semantic content is computed alongside with the f-structure during parsing, changing the procedure into:

1. Try to find an f-structure f and a possible semantic structure σ which matches w.

2. Check f for completeness and coherence.

If f is not complete and coherent, σ cannot be expected to represent a semantic content. As with the description by analysis approach, whether completeness and coherence is sufficient will depend on which further constraints are put on the grammar.

2 Type-driven unification grammar

2.1 Categorial grammar in a unification setting

To obtain completeness in a pure unificational setting, say PATR (Shieber 1985), one can associate grammatical categories and types, and induce constraints on the grammars such that an empty subcategorization list always yields a feature structure which contains a representation of a well-defined semantic object. A way to achieve this is by starting by a recast of the simple categorial grammars (CGs) in a PATR-style unification grammar, and proceed to extend and alter the grammar in a controlled way which retains completeness.

As a CG, with a semantic component, associates a string with a category and a logical term of the type corresponding to the category, we must represent the category and the term in a feature structure. The cancellation schema with its associated semantics

$$\begin{array}{lll} X/Z & Z & \Longrightarrow & X \\ f & t & & f(t) \end{array}$$

can be expressed as a PATR-rule:

$$(1) \qquad \begin{array}{l} X \longrightarrow Y\ Z \\ \quad \langle X\ CAT \rangle = \langle Y\ CAT\ RES \rangle \\ \quad \langle Z\ CAT \rangle = \langle Y\ CAT\ ARG \rangle \\ \quad \langle X\ SEM\,FUN \rangle = \langle Y\ SEM \rangle \\ \quad \langle X\ SEM\,ARG \rangle = \langle Z\ SEM \rangle \\ \quad \langle Y\ CAT\ DIR \rangle = R \end{array}$$

and similarly with $Z\ \ Z\backslash X \Longrightarrow X$.

This is nothing but a faithful implementation of the Ajdukewicz–Bar-Hillel CGs. But we may extend and modify the format. First, we may add other features than CAT and SEM, say for agreement, and as rules take rules which has one of the cancellation schemata as a backbone and in addition equations with the new features. Call these grammars Type Unification Grammars (TUG). This will extend the (weak and strong) generative power, but will not disturb the completeness of the semantic interpretation procedure.

Second, we may introduce rules from flexible categorial grammars where the correspondence between category/type and interpretation is retained, as in $A/B \Longrightarrow (A/C)/(B/C)$ with semantics: $f \Longrightarrow \lambda G(\lambda x(f(G(x)))$ (R5 from Moortgaat 1988). This may be expressed as:

$$
(2) \quad
\begin{array}{l}
X \longrightarrow Y \\
\quad \langle X\ TYPE\ ARG\ ARG \rangle = \langle X\ TYPE\ RES\ ARG \rangle \\
\quad \langle X\ TYPE\ ARG\ RES \rangle = \langle Y\ TYPE\ ARG \rangle \\
\quad \langle X\ TYPE\ RES\ RES \rangle = \langle Y\ TYPE\ RES \rangle \\
\quad \langle X\ SEM\ LAM \rangle = \langle X\ SEM\ SCOPE\ SCOPE\ ARG\ FUN \rangle \\
\quad \langle X\ SEM\ SCOPE\ LAM \rangle = \langle X\ SEM\ SCOPE\ SCOPE\ ARG\ ARG \rangle \\
\quad \langle X\ SEM\ SCOPE\ SCOPE\ FUN \rangle = \langle Y\ SEM \rangle \\
\quad +\textit{additional equations for direction and possible other features.}
\end{array}
$$

Similarly, all rules proposed in the literature could be expressed in this setting, e.g. the rules R1-R6 from Moortgat (1988).

2.2 Lambda reduction

The format so far may construct semantic representations of λ-terms which are not fully reduced. One way to get simpler representations is to, instead of rule (1) above, implement the basic cancellation schemata as follows (Lambda-reduced Unification Grammars, LUG).

$$
(3) \quad
\begin{array}{l}
X \longrightarrow Y\ Z \\
\quad \langle X\ TYPE \rangle = \langle Y\ TYPE\ RES \rangle \\
\quad \langle Z\ TYPE \rangle = \langle Y\ TYPE\ ARG \rangle \\
\quad \langle X\ SEM \rangle = \langle Y\ SEM\ SCOPE \rangle \\
\quad \langle Z\ SEM \rangle = \langle Y\ SEM\ LAM \rangle \\
\quad +\textit{additional equations for direction and possible other features.}
\end{array}
$$

For this to work, the semantics of the functor must have a LAM-feature and a SEM-feature. One can define a normal form of lambda terms and show that each term is equivalent to a term in this form. There might be a problem if the feature structure represents a term containing a variable with several unrelated occurrences, as x in $\lambda x[every(\lambda x[man(x)])(\lambda y[love(x)(y)])]$. Moreover, we cannot avoid such terms altogether, as e.g., the application of $\lambda X[R(X)(X)]$ to $\lambda y[\psi]$ yields $R(\lambda y[\psi])(\lambda y[\psi])$. What we can do is to define a certain normal form of representations of the normal form terms which is restricted with respect to how parts of the structures may be shared, which we will call *good* structures, and show that for such structures the new rule format works.

2.3 Partially evaluated β-reductions

One lambda reduction is in general not enough; the end result may contain beta-redexes. One way to get a simpler result is to partially evaluate lambda-reductions in the lexicon. This sort of partial evaluation has been used and discussed by Pereira and Shieber (1987), Reyle (1988), and Moore (1989). It is well known that there are cases when it does not work, in particular (type-raised) NP-conjunction. Reyle showed that partial evaluation was always possible when each lambda binds at most one variable occurrence. We extend this and based on the concept of good representations we show that partial evaluation is possible in certain cases where more than one variable occurrence is bound by a lambda.

This will allow the most natural treatment of control verbs, e.g. in the representation of the semantics of "persuade" as $\lambda g[\lambda m[\lambda j[persuade(g(m))(m)(j)]]]$, g may be partially evaluated with respect to m.

2.4 Flexible categorial grammar and partial evaluation

The flexible categorial rules cannot be added to LUG as simply as they were to TUG, as they will not in general result in the proper normal forms. With partial evaluation, there is another option. (R5) can be implemented as

(4)
$$
\begin{array}{l}
X \longrightarrow Y \\
\quad \langle X\ TYPE\ ARG\ ARG\rangle = \langle X\ TYPE\ RES\ ARG\rangle \\
\quad \langle X\ TYPE\ ARG\ RES\rangle = \langle Y\ TYPE\ ARG\rangle \\
\quad \langle X\ TYPE\ RES\ RES\rangle = \langle Y\ TYPE\ RES\rangle \\
\quad \langle X\ SEM\ LAM\ LAM\rangle = \langle X\ SEM\ SCOPE\ LAM\rangle \\
\quad \langle X\ SEM\ LAM\ SCOPE\rangle = \langle Y\ SEM\ LAM\rangle \\
\quad \langle X\ SEM\ SCOPE\ SCOPE\rangle = \langle Y\ SEM\ SCOPE\rangle \\
\quad +\textit{additional equations for direction and possible other features.}
\end{array}
$$

Similar versions can be given for the rules R1–R6; call them L1–L6. These implementations can be shown to be correct: If one of L3-L6 is applied to a (partially evaluated) representation of a term, t, the result will be a legally partially evaluated representation of the term s, which is the result of applying the corresponding R-rule to t. Similarly, if R1 or R2 are applied to two signs, in which case some β-reductions may be executed in addition.

There are still differences between this approach and the most common practise in unfication-based semantic interpretations, but observe that in L1–L6, all the type equations and semantic equations are parallel. This opens for an alternative architecture where types and semantics are mixed, and in the next round for an architecture where syntactic features are mixed with the types and semantics as in Categorial Unification Grammars (Karttunen 1986, Uszkoreit 1986), Unification Categorial Grammar (Zeevat et. al. 1987). This will be quite similar to the use of subcategorization in HPSG. We claim these approaches are complete because their semantic component can be derived from a categorial one by partial evaluations.

3 Conclusions

We have considered two different approaches to the syntax- semantics interface in unification-based grammars, the *direct* type-driven approach and the *indirect* LFG-approach where the second stage, from the representations to the interpretations is type-driven. What is common to both approaches is that the way to impose completeness is a link between subcategorization and semantic types. The difference between the two is in the way subcategorization is carried out.

References

Fenstad, J.E., P.-K. Halvorsen, T. Langholm, J. van Benthem, 1987, *Situations, Language and Logic*, Reidel

Halvorsen, P.-Kr., 1983, "Semantics for Lexical-Functional Grammar", *Linguistic Inquiry*, **14**, 567–615

Halvorsen, P.-K. And R. Kaplan, 1988, "Projections and Semantic Description in Lexical-Functional Grammar", in *Proceedings of the International Conference of Fifth Generation Computer Systems*, ICOT, Tokyo, Japan.

Karttunen, L., 1986, *Radical Lexicalism*, CSLI-Report 86–68

Moore, R., 1989, "Unification-Based Semantic Interpretation", *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, 33–41.

Moortgat, M., 1988, *Categorial Investigations: Logical and Linguistic Aspects of the Lambek Calculus*, Foris, Dordrecht, Holland.

Pereira, F.C.N. and S.M. Shieber, 1987, *Prolog and Natural-Language Analysis*, CSLI Lecture Note, Stanford, CA.

Pollard, C. 1989, "The Syntax-Semantics Interface in a Unification-Based Phrase Structure Grammar", *Views of the Syntax/Semantics Interface*, KIT-Report 74, Technische Universitat Berlin, Berlin, Germany.

Pollard and Sag, 1988, *Information-Based Syntax and Semantics*, Vol. 1, CSLI Lecture Note, Stanford, CA

Reyle, U., 1988, "Compositional Semantics for LFG", in Reyle and Rohrer (eds.), *Natural Language Parsing and Linguistic Theories*, Reidel, Dordrecht, Holland.

Shieber, S., 1985, *An Introduction to Unification-Based Approaches to Grammar*, CSLI Lecture Note, Stanford, CA

Uszkoreit, H., 1986, "Categorial Unification Grammar", in *Proceedings of the 11th Coling.*

Zeevat H., E. Klein, J. Calder, 1987, "Unification Categorial Grammar", in *Edinburgh Working Papers in Cognitive Science, Vol. 1.*

Linking One Semantic Interpretation System to Different Syntactic Formalisms

Sebastian Millies
Manfred Pinkal
University of the Saarland
Dept. of Computational Linguistics
6600 Saarbrücken
Germany
{millies,pinkal}@coli.uni-sb.de

Paper presented at the Dagstuhl Colloquium on Semantic Formalisms in Natural Language Processing, February 1993

Summary

We propose a model of syntax-semantics interaction, in which an autonomous semantic module accesses syntactic information via abstract interface predicates. The details of the syntactic representation are hidden in the grammar-specific formulation of the interface. This approach makes semantics largely neutral with respect to different grammar formalism. The paper outlines the concept of the semantic core component, and describes interfaces to GB, HPSG, and LFG grammar systems.

1 Integrated and Modular Syntax-Semantics Interfaces

Different models for the interaction of syntax and semantics have evolved out of work in different grammar formalism. In the HPSG ([PS92]) approach, which we call an "integrated" approach, syntactic and semantic representations are encoded in the same data structure (called a "sign"). Crucially, semantically relevant parts may be spread out over the entire sign, and some parts of a sign may have both a syntactic and semantic function. The approach allows strong interactions between the different parts of a sign because information on all levels is present to every principle of grammar. This overall organization of the theory blurs the notion of what a semantic object is and of what a semantic operation consists in. As a consequence, it is difficult to ensure that intermediate semantic representations, and operations on such representations, make clear semantic sense. Also, the role of syntactic information in semantic processing is never explicitly described. A practical disadvantage is that distributed development of large NL-systems becomes difficult.

We follow an approach in which the syntax does neither contain semantic information nor instructions to build semantic representations. We have implemented a semantic interpretation system (called "SCOLD"[1]) which builds semantic representations while recursing over syntactic structures, inspecting the syntax to extract certain kinds of information. This method is generally known under the label of "description-by-analysis". The essential point, however, is that we have defined a core semantics which is largely independent of the grammar formalism used in the syntax. We have achieved this by providing a set of *abstract* syntax interface predicates, which are differently defined for each grammar formalism. The point of abstractness is to specify the functional role of syntax in the process of building semantic representations, hiding the details of how the requisite information is realized in a separate interface component. On the one hand, the

[1] "SCOLD" is an acronym for "Syntax-sensitive Computation of Logical Descriptions".

system is very flexible: It is possible to extract arbitrary information from syntax. But on the other hand, this happens in a controlled way in the interface, not on an ad-hoc basis directly in the formulation of semantic interpretation rules.

We use a semantic representation language which we call "λ-DRT". Among the most important distinguishing features of λ-DRT are the use of discourse markers (as in standard DRT, [Kamp84]) and the use of abstraction (as in the λ-calculus). It is type-driven, declarative and unification-based, using partially evaluated λ-expressions in the style of [PS87]. Our system constructs semantic representations bottom-up using a small set of semantic operators. The most important basic operation is a non-standard version of functional composition. The choice of functional composition (as opposed to application) allows a straightforward analysis of many problematic constructions, e. g. those involving quantifying into term phrases (cf. [Roo85]). The semantic operators correspond to basic semantic notions, and thus at each step the outcome of a semantic operation will be model-theoretically interpretable. Also, our operators allow a general formulation of semantic interpretation independent of the notation for the semantic representations. The particular choice of semantics formalism is not essential with regard to our concept of the syntax-semantics interface.

2 Linking SCOLD to GB, HPSG, and LFG

Semantic operators work on semantic objects which are associated with particular syntactic objects, e.g. nodes in a phrase structure tree in GB ([Cho81]), values of certain syntactic features in HPSG ([PS92]), or parts of f-structure in LFG ([KB82]).[2] The application of the semantic operators to their arguments is sensitive to the particular syntactic environment of the associated node. For example, there are three basic semantic operators called *compose, predication,* and *abstract,* which suffice for local semantic interpretation. Semantic interpretation proceeds by collecting the meanings of some set of nodes (the current "local domain") and freely applying the operators to them. In this process, the operators "consume" their arguments and "produce" a result, until only one semantic object is left, which is then assigned to some node. The local domain is given by the syntax-interface predicate *local-domain**.[3] The final result will be associated with the root of the domain, if this is a *compatible_type_assignment** to that node. A syntactic node may be compatible with a range of types.

Non-local semantic phenomena like quantifier scope or anaphora are also treated. Scope readings are generated by a version of Nested Cooper Storage ([Kel88]): There are semantic constructors *store* and *quantify-in* and interface predicates *non-locally-interpretable*, quant-in-permissible*,* and *constraints-on-scope**. The meaning of any constituent declared as non-locally interpretable may be stored, and later quantified into a meaning associated with a node where this is permissible, provided that the scope constraints hold. Anaphoric binding is implemented as the unification of discourse markers in semantic structure, provided that the nodes from where the markers originate fulfill *constraints-on-binding**.

The Semantic Interpretation Algorithm

In a nutshell, semantic interpretation simply consists in a recursive postorder traversal of the root domain and assignment of the result of the evaluation to the root, followed by enumerating binding possibilities.[4]

The GB Interface

The GB-interface is written so as to enable direct interpretation of S-structure. In GB the local domain contains the immediate subconstituents of a node, and as a consequence, the operators can

2 In the following, we will simply say "node", instead of "syntactic object".

3 Predicates from the syntax interface are in cursive font and marked by the diacritic '*'.

4 Binding relations can also be left unresolved until later stages of processing. The evaluation procedure returns an underspecified representation of possible bindings. The mechanism is described elsewhere.

only apply to semantic objects associated with sister nodes. We assume binary branching trees in GB and non-deterministically select one element from the domain as the functor, the other as the argument, returning an empty rest domain. Complements have raised type, so that normally complements are functors over heads, with the exception of empty functional heads. Because the algorithm is type-driven, only one of the combinations is possible in each case. The result of composing functor and argument will be assigned to the mother node. An application of the move-α transformation (leaving behind a co-indexed trace) will correlate in the semantics with an abstraction operation: When a moved functor is composed with an argument, first the argument is abstracted over with the referential index of the functor.[5] Referential indices in the syntax correspond to semantic variables, so that syntactic co-indexing guarantees that the correct variable is abstracted. In order to implement this, we expect the syntax to know when it has moved a constituent. For the treatment of quantifier scope, we provide a definition of scope-taking elements in syntactic terms and say that every element that takes scope may go in storage. As for retrieval, specifying both IP and NP as nodes that can be quantified in automatically takes care of quantifying into term phrases. We assume that the constraint on scoping uses subjacency. As for binding, the nodes from which pronoun and antecedent originate must agree in certain features, and one must c-command the other.[6] In order to implement these relations, every semantic object bears a pointer to its home syntactic node. The syntactic tests involved are the same as those used in syntax, there is no duplication of syntactic information for semantic purposes.

The HPSG Interface

The local domain of interpretation in HPSG is not the set of all daughters, as in GB, but the set of all daughters except the filler-daughter.[7] Fillers are reconstructed by HPSG into their D-structure (or NP-structure) position, either by postulating a trace, or by transforming the lexical semantics of the verb. Therefore, we never interpret them in their surface position. Complements are type-raised in semantics, and because of the absence of empty functional heads in our HPSG grammar, complements and adjuncts alike are uniformly treated as functors over heads. As a further difference to GB, we allow multiply branching structures. HPSG does not employ movement rules.[8] This is expressed in the syntax-interface in the most succinct manner possible, by specifying *theta-governed** as trivially true and *not-theta-governed** as trivially false. The interface predicates for generating different scopings are the same as in GB, except that we do not assume any syntactic constraint on the extraction domain[9] and that the non-configurational notion of o-command replaces the tree-based notion of c-command. For the purpose of computing o-command, every semantic object again bears a pointer to its origin, which in this case is a position on a SUBCAT-value.

The LFG Interface

The idea for an LFG-interface is to decompose f-structures into their constituent parts. The local domain is the union of all grammatical functions except non-thematic functions governed by the PRED-value, where set-valued functions contribute all the elements in the set. We then reduce this domain non-deterministically, relying on the types of the semantic objects to drive the composition. The result is assigned to the f-structure as a whole. However, there is one additional complication: In order to ensure that semantic representations for the complements of a head are applied in the correct order, the semantic objects in the domain will be indexed by grammatical function. Subcategorized functions must be applied first, and in the order in which they appear in the semantic value of the PRED function. Quantifier scope will be treated analogously to GB; binding will work with the notion of f-command. It remains to be determined how the relative "flatness" of

---

5 Referential indices themselves are also accessed by an interface function.

6 Our implementation adopts the constraint from [Rei83], with the revisions in [Pi91].

7 Note that we can recurse over the set of daughters, because it contains signs, but could not recurse over sets of SYNSEM values found on SUBCAT lists.

8 The HPSG analysis of long-distance dependencies does not have the effect of semantically relevant movement, as explained above.

9 No plausible analysis for syntactic constraints on quantifier scope in HPSG is known to the authors.

f-structures (as opposed to S-structure trees) influences the formulation of scope and binding constraints.

Characterizing the Dependence of Semantics on Syntax

We have designed a semantic interpretation module so that it can be adapted to be used with different grammar formalisms (or grammatical theories) by re-defining only the syntax-interface. However, complete neutrality with respect to the syntax formalism will probably be impossible to attain. It is therefore important to delimit the features of a grammar formalism which make it suitable to work in tandem with a separate, syntax-sensitive semantics module. In our work on SCOLD, we have encountered several problems which must be addressed by any model for a portable semantic interpretation system.

• The syntactic analysis of some particular phenomenon may be radically different in the various syntax formalisms. For example, in control constructions GB theory postulates an empty element PRO in an embedded clause, whereas HPSG embeds an unsaturated verbal complement. The crucial difference is that in HPSG, a semantically relevant object occurs outside the domain of recursion. In order to accomodate the HPSG analysis to a semantic framework where control verbs take propositions as arguments, one would have to add a "domain-completion rule" to the HPSG interface. Adding special rules of this kind works just fine. But while in the one case a semantic analysis follows naturally from a syntactic decision, in the other case an additional stipulation has to be made.

• A formalism may simply not provide the syntactic information necessary to implement some constraint on interpretation. The difficulty may be "accidental" – there is no essential reason why HPSG should not be able to come up with a constraint on quantifier scope, for example. But the difficulty may also be fundamental, as is the case with raising in HPSG. HPSG makes use of a meta-logical principle to control the interpretation of raising construction. The reason is that HPSG has no thematic roles in the syntax, but considers semantic roles alone. Thus, from the perspective of a modular semantics, HPSG seems deficient in the kind of syntactic information it provides.

• Our system currently assumes a *monostratal* syntactic analysis. In the GB case, we can straightforwardly adapt the interface to interpret LF instead of S-structure (all quantifiers are interpreted in situ), but we cannot presently exploit syntactic information on several levels of representation simultaneously. This might be considered a sensible constraint on syntactic theory, but must also be recognized as a major limitation for our system.

4 Implementation

The coverage of our semantics system currently includes head-complement structures of German and English, relative clauses, sentential adjuncts, and adjective constructions. Several constructions which require a complex interaction between syntactic and semantic constraints have been treated successfully in SCOLD linked to GB. One example is the so-called "donkey anaphora". We have implemented a working model for the theoretical description (in [Pi91]) of the integration of the basic c-command constraint on binding ([Rei83]) with semantic considerations. Another, semantically much more complex example is the analysis of predicative and attributive comparative constructions of German and English. The analysis heavily relies on functional composition, especially quantifying into term phrases (cf. [LP92]).

5 Conclusion

We have designed a modular system in which semantics can interact with syntax by accessing syntactic information flexibly, but in a controlled way. We have made the semantic interpretation system relatively independent of the grammar formalism used in syntax by giving a theory-neutral formulation of the semantics and providing specific syntax-interfaces for different grammar formalisms. This improves the theoretical perspicuity of the semantic interpretation process and the syntax-semantics interaction, facilitates porting of the semantics to different grammar systems, and supports the distributed development of large NL-systems. While the independence and theory-neutrality of semantics cannot be perfect, we have demonstrated that a poly-theoretic approach can get considerable mileage. We have implemented a prototype of our system.

Among the issues to be addressed by future work are the following:

- Representations in alternative semantic target formalisms will be built using the semantic operators in a theory-neutral way.
- The sequential architecture of the current system will be made more flexible. In particular, we will investigate a co-routining approach.
- We will investigate conditions that delimit the range of syntax formalisms and syntactic theories compatible with our approach in a more systematic way.

6 References

[Cho81] Chomsky, N. (1981): *Lectures on Government and Binding*, Foris, Dordrecht.

[Fen87] Fenstad, J./ Halvorsen, P.-K./van Benthem, J./Langholm, T. (1987): *Situations, Language and Logic,* Reidel, Dordrecht.

[JK90] Johnson, M./Kay, M. (1990): *Semantic Abstraction and Anaphora,* in: Proceedings of COLING 90, Helsinki.

[KB82] Kaplan, R./Bresnan, J. (1982): *Lexical-Functional Grammar,* in: Bresnan, J. (ed.), *The Mental Representation of Grammatical Relations,* The MIT Press, Cambridge, Ma.

[Kamp84] Kamp, H. (1984): *A Theory of Truth and Semantic Representation*, in: J. Groenendjik (ed.), *Formal Methods in the Study of Language*, Foris, Dordrecht.

[Kel88] Keller, W. (1988): *Nested Cooper Storage,* in: Reyle, U./Rohrer, Chr. (eds.), *Natural Language Parsing and Linguistic Theories,* pp. 432-447, Reidel, Dordrecht.

[LP92] Lerner, J.-Y./Pinkal, M. (1992): *Comparatives and Nested Quantification*, in: Proceedings of the Amsterdam Colloquium on Logic, Language and Information.

[Pi91] Pinkal, M. (1991): *On the Syntactic-Semantic Analysis of Bound Anaphora*, in: Proceedings of the 5th EACL, Berlin.

[PS92] Pollard, C./Sag, I. (in prep.): *Head-Driven Phrase Structure Grammar,* to be publ. by University of Chicago Press and CSLI.

[PS87] Pereira, F./Shieber, S. (1987): *Prolog and Natural Language Analysis,* CSLI Lecture Notes 10, CSLI, Stanford.

[Rei83] Reinhart, T. (1983): *Anaphora and Semantic Interpretation,* University of Chicago Press.

[Roo85] Rooth, M. (1985): *Association with Focus,* Dissertation, Dept. of Linguistics, University of Massachusetts, Amherst.

ASSIGNING A SEMANTIC SCOPE TO OPERATORS
Extended Abstract for the Dagstuhl Worshop

Massimo Poesio
Department of Computer Science
University of Rochester
Rochester, NY 14627-0226, USA
poesio@cs.rochester.edu

February 23, 1993

Because new ways of obtaining distinct interpretations for sentences are continuously discovered, coming to grips with ambiguity is becoming more and more of a necessity for developers of natural language processing systems, linguists and psychologists alike. Two distinct questions have to be answered: how can listeners (and how should machines) deal with the combinatorial explosion of readings? Do we really use the brute-force strategy of considering all of the available readings, and then choose among them? And, if we do choose among several readings, how is that done? Among the various sorts of ambiguity, the *scopal* ambiguity of operators,[1] in particular, has recently become again the focus of much interest [Alshawi and Crouch, 1992; Deemter, 1991; Fenstad *et al.*, 1987; Poesio, 1991; Reyle, 1991].

I believe that in order to develop systems which interact with humans we need to solve the problem of combinatorial explosion, and to do that we need a plausible theory about the process by which people arrive at a preferred interpretation for scopally ambiguous sentences. I am not aware of any non-heuristical solutions to the problem of how to choose one reading in the natural language processing literature.[2] Several linguistic and psycholinguistic proposals exist, but none of the principles proposed in this literature appears to be applicable in all contexts and, furthermore, they appear to interact in ways which are largely unexplained, as shown by Kurtzman and MacDonald's work [1992] on experimentally verifying the validity of these principles.

Kurtzman and MacDonald's most important findings were as follows:

[1] I use here the term *operator* as it is used by Heim [1982], i.e., to mean either quantifier or modal/tense operator.

[2] See [VanLehn, 1978; Hurum, 1988; Moran, 1988] for examples of state-of-the-art techniques.

1. A preference for the subject in active sentences to take wide scope was indeed observed; interestingly, however, this preference was stronger when the quantified phrase was of the form *a P R'ed every Q* than when it had the form *every P R'ed a Q*. (Contrary to what a weight heuristic would predict.)

2. In the case of passive sentences, no significant preference for the subject taking wide scope was observed.

3. Most interestingly, a preference was observed for the embedded NP in a complex NP to take wide scope: for example, in (1), "an admiral" would tend to take wide scope.

   (1) Each daughter of an admiral married a captain.

   This result is not predicted by any of the principles tested by Kurtzman and MacDonald.[3]

Kurtzman and MacDonald's results, while not conclusive, do serve to constrain the range of possible proposals. The most interesting observation is that none of the principles proposed in the literature can account for all the observed effects, and actually we have counterexamples to all of them, including the lexical preferences. In particular, no evidence for a Left-to-Right processing principle of the sort proposed by Lakoff [1971] was found. The facts about complex NP do seem to suggest that structural factors play a role, contrary to the view suggested in [Katz, 1980] according to which preferences can be explained entirely at a pragmatic level ([1992], p.42). Kurtzman and MacDonald also hypothesize that "...processes that are not strictly dedicated to the interpretation of scope relations may nonetheless influence the interpretation of quantifier scope ambiguities." ([1992], p.22). Finally, Kurtzman and MacDonald conclude that "...the results leave open the question of whether the building and selection of representations of scope are mandatory processes" ([1992], p.45).

The hypothesis I present in the paper is based on Kurtzman and MacDonald's results, as well as on my study of transcripts of actual conversations in which people have to accomplish a task,[4] thus focusing on cases in which people have strong intuitions about the scope of the operators, and avoiding artificially sounding utterances or utterances that people find difficult to understand.

My hypothesis about the process by which operators are assigned their scope in our dialogues can be summarized as follows:

> The operators in a sentence **s** get assigned their scope as a side effect of the process of interpretation which builds up a model of the situation represented by the sentence. This model is constructed by applying 'model construction rules' to the initial interpretation of the sentence, assumed to be a tree isomorphic to

[3]Examples like (1) were considered by May in his thesis [May, 1977] and discussed by Reinhart.

[4]These transcripts have been collected in the framework of the TRAINS project [Allen and Schubert, 1991]. The conversations take place between two people trying to transport goods by train within a certain deadline. The conversational participants are given a map of the 'world,' and generic information about the capabilities of engines, boxcars, etc. The transcripts of the conversations collected in 1991 are presented in [Gross *et al.*, 1992].

the surface structure of the sentence, with lexical items replaced by their semantic interpretation. A model construction rule exists for each operator, which is applied when the contextual aspects of the operators' interpretation have been resolved—in the case of quantifiers, for example, the contextual parameter in question is the 'resource situation' of the quantifier. The relative scope of operators thus depends on the order in which these contextual factors are determined, and scope relations are determined by the relations of dependency among the contextual factors—e.g., by informational inclusion relations between the 'resource situations' that an agent already knows, or may assume to exist.

To those familiar with DRT the relation between this proposal and the interpretation method proposed in DRT—the DRS construction algorithm—should be apparent. In this paper, the model of an utterance is represented as a DRS, and the 'model construction rules' are DRS rules. What I am proposing, then, is that the interpretation of each operator has a contextual aspect, and that the DRS construction rules for that operator are only applied when the contextual aspect is resolved (by independent processes of pragmatic reasoning).

Consider (2), for example. What I am proposing is that whether "every kid" or "a tree" takes wide scope depends on where the listener 'starts' from when building a model of the sentence: if she starts by first identifying the group of kids that "every" is quantifying over, and then proceeds to 'build' identify for each of these kids a situation which contains a tree the kid is climbing, then "every kid" will take wide scope. In DRT terms, this is like saying that which operator takes wide scope depends on whether the DRS construction rule for universals or that for indefinites applies first, and this depends on whether the construction of the model is started by adding a group of kids to the root DRS or by adding a tree.

(2) Every kid climbed a tree.

¿From the point of view of this proposal, the non-structural disambiguation factors presented in the literature can be explained as either (i) manifestations of the way in which the model of an utterance is built, or (ii) resulting from the constraints imposed by operators on the model resulting from the application of the construction rule. I'll consider two of these factors: the topic principle proposed by Katz and Ioup's hypothesis that lexical factors like the 'quantifier hierarchy' play a role. I assume that an NP is taken to be 'in topic' if its relation with the prior context can be determined on the basis of very simple inferences (say, by lexical priming effects), which implies that the resource situation of that NP can be readily identified. The resource situation of the operators not in topic will be determined on the basis of the resource situation of the NP in topic, resulting in informational inclusion relations, as discussed above. Consider for example (3):

(3) a. We professors believe that bringing the students in contact with nature is essential both for their personal growth and their physical well-being, so we often organize little outings.

b. This morning, for example, we went with the class to explore the woods near Cayuga Lake.

c. The chidren got very excited;

d. every kid climbed a tree.

In this case, according to the hypothesis I presented, the preference for "every kid" to take wide scope over "a tree" is explained by the fact that, that NP being in topic, its relation with the rest of the story is suggested by lexical priming factors, and therefore its resource situation is identified first, while the resource situation of "a tree" is identified in relation with the resource situation of "every kid."

Because subjects are often the topic of sentences in English, it seems plausible to stipulate that listeners learn a 'weak' default rule suggesting that, when no context is available, the NP in subject position is the topic of the sentence. The preference for subjects to take wide scope could then be explained in terms of the hypothesis presented above as well. (This possibility needs to be explored in more detail.)

Consider now the case of the other disambiguation factor proposed by Ioup, the lexically encoded preference for certain operators to take wide scope. Definite descriptions are the paradigmatic case of operator that tends to take wide scope; this preference is the strongest disambiguation effect to be observed in our dialogues. The hypothesis presented above can account for this preference as well: because the choice of a resource situation for definite descriptions is restricted by the constraint that this resource situation be shared among the conversational participants, a definite description may take narrow scope with respect to another operator only if the resource situation of this other operator can be assumed to be shared. In practice, this rarely happens; the known cases of other operators taking wide scope over definites, such as (4) and (5), are all cases in which the definite has to be interpreted relative to the resource situation of another operator, and this relation is specified by generic knowledge that can be assumed to be shared:

(4) Every school sent the principal to the meeting.

(5) I am always tired at the end of the semester.

(In our dialogues the resource situation for definite descriptions is, in most cases, the current visual scene.) More in general, my hypothesis predicts that those operators whose resource situation has to be identified independently from the resource situations of other operators should tend to take wider scope.

In order to make the hypothesis just presented more precise, we need (i) to describe the process of disambiguation in more detail, and (ii) to study the procedures by which the resource situations of different kinds of operators get identified. In this paper, I concentrate on the first part of the task, and I present a proposal about the input to the process of scope disambiguation and the connection between this process and other processes of discourse interpretation. A description of my work on providing a detailed account of the process of definite description interpretation, and, in particular, the processes by which the resource situation of definite descriptions are identified, is in [Poesio, 1993]. I am currently working with M. Kameyama and R. Passonneau to provide an account of the process by which the resource situation for tense gets identified [Kameyama *et al.*, 1993].

I propose that the input to scope disambiguation, as well as to the other discourse interpretation processes, is an intermediate representation called *logical form* (LF). The logical form is structurally identical to the parse tree, except that lexical items have been replaced by their semantic interpretation; neither the scope of operators nor the interpretation of anaphoric expressions have been interpreted yet.[5] The logical form is the interface between the parser and lexical interpretation modules on the one side, and the 'pragmatic processor' on the other side. The idea of a 'logical form level' as a way for splitting up the work between context-dependent and context-independent aspects of natural language interpretation has a long history both in NLP [Webber, 1978; Woods, 1978; Schubert and Pelletier, 1982; Fenstad *et al.*, 1987; Alshawi, 1992] and in linguistics, especially in the generative framework [Chomsky, 1977; May, 1977; May, 1985]. What is new in my proposal is that, first of all, the LF I use has a model-theoretic interpretation, so that a notion of *compatible disambiguation* can be defined. Secondly, I insist for preserving in the logical form the information about surface structure. There are two reasons for this: first of all, I believe that part of the explanation for the scope puzzle is that constraints—like the Scope Constraint [Heim, 1982] or the constraints on the scope of polarity items [Ladusaw, 1977]—greatly reduce the number of available interpretations.[6] Similar, well-known constraints also affect other discourse interpretation processes, such as the choice of an interpretation for pronouns. Both kinds of constraints are usually assumed to be structural. Secondly, structural notions such as parallelism play an important role also in *suggesting* preferred interpretations (e.g., for pronouns: cfr. Kameyama's 'property sharing' effects in centering [Kameyama, 1985]).

The production of the Logical Form initiates a process consisting of repeated phases of hypothesis generation, verification, comparison with other hypotheses and, possibly, acceptance, much as in, say, Hurum's proposal [1988] or the proposals based on abduction [Charniak and Goldman, 1988; Hobbs *et al.*, 1988]. The main distinguishing feature of my theory is that it comes with a proposal about the kind of information used for generating hypotheses about the relative scope of quantifiers—which allows me, for example, to make predictions about how difficult it will be to interpret certain sentences. A second difference is that I propose to formalize the process of hypothesis formation and selection in terms of operations on a *mental state* structured as in Asher and Kamp's proposals [Kamp, 1990]. The idea is that the 'perceptual event' of obtaining a logical form for an utterance results in adding to the current mental state a *verbal experience*—a mental object of a particular type whose content is the logical form. This mental event triggers additional operations, among

[5] This should not be interpreted as meaning that I assume the parser to build a complete interpretation for an utterance prior to the intervention of pragmatic reasoning. On the contrary, the assumption in TRAINS is that the parser ships 'chunks' of interpretation to the modules which follow. I will leave this issue aside in this paper.

[6] A disambiguation factor which plays an important role in our dialogues and, to the best of my knowledge, can only be explained in structural terms is the fact, discussed among others by Heim in [Heim, 1987], that indefinites in there-insertion sequences like (6) take narrow scope with respect to tense.

(6) There is an engine at Avon.

which ones which produce hypotheses about the model of the sentence, and are basically modified versions of the DRS construction rules.

References

[Allen and Schubert, 1991] Allen, J. F. and Schubert, L.K. 1991. The TRAINS project. TRAINS Technical Note 91-1, University of Rochester, Department of Computer Science.

[Alshawi and Crouch, 1992] Alshawi, H. and Crouch, R. 1992. Monotonic semantic interpretation. In *Proc. 30th. ACL*, University of Delaware. 32–39.

[Alshawi, 1992] Alshawi, H., editor 1992. *The Core Language Engine*. The MIT Press.

[Charniak and Goldman, 1988] Charniak, Eugene and Goldman, Robert P. 1988. A logic for semantic interpretation. In *Proc. ACL-88*, Buffalo, NY. 87–94.

[Chomsky, 1977] Chomsky, N. 1977. On *wh*-movement. In Culicover, P.; Wasow, T.; and Akmajian, A., editors 1977, *Formal Syntax*. Academic Press, New York. 71–132.

[Deemter, 1991] Deemter, Kees van 1991. *On the Composition of Meaning*. Ph.D. Dissertation, University of Amsterdam.

[Fenstad *et al.*, 1987] Fenstad, J.E.; Halvorsen, P.K.; Langholm, T.; and Benthem, J.van 1987. *Situations, Language and Logic*. D.Reidel, Dordrecht.

[Gross *et al.*, 1992] Gross, D.; Allen, J.; and Traum, D. 1992. The TRAINS 91 dialogues. TRAINS Technical Note 92-1, Computer Science Dept. University of Rochester. to appear.

[Grosz *et al.*, 1896] Grosz, B. J.; Jones, K. Sparck; and Webber, B. Lynn, editors 1896. *Readings in Natural Language Processing*. Morgan Kaufmann, Los Altos, CA.

[Heim, 1982] Heim, I. 1982. *The Semantics of Definite and Indefinite Noun Phrases*. Ph.D. Dissertation, University of Massachusetts at Amherst.

[Heim, 1987] Heim, I. 1987. Where does the definiteness restriction apply? In Meulen, A.ter and Reuland, E., editors 1987, *The Representation of (In)definitess*. MIT Press.

[Hobbs *et al.*, 1988] Hobbs, J. R.; Stickel, M.; Martin, P.; and Edwards, D. 1988. Interpretation as abduction. In *Proc. ACL-88*, Buffalo, NY. 95–103.

[Hurum, 1988] Hurum, Sven 1988. Handling scope ambiguities using domain-independent heuristics. Technical Report TR 88-12, University of Alberta, Edmonton, Canada.

[Kameyama *et al.*, 1993] Kameyama, M.; Passonneau, B.; and Poesio, M. 1993. Temporal centering. In *Proc. ACL-93*, Columbus, OH.

[Kameyama, 1985] Kameyama, M. 1985. *Zero Anaphora: The case of Japanese*. Ph.D. Dissertation, Stanford University, Stanford, CA.

[Kamp, 1990] Kamp, H. 1990. Prolegomena to a structural account of belief and other attitudes. In Anderson, C. A. and Owens, J., editors 1990, *Propositional Attitudes—The Role of Content in Logic, Language, and Mind*. University of Chicago Press and CSLI, Stanford. chapter 2, 27–90.

[Katz, 1980] Katz, J. J. 1980. Chomsky on meaning. *Language* 56:1–41.

[Kurtzman and MacDonald, 1992] Kurtzman, H. and MacDonald, M. 1992. Resolution of quantifier scope ambiguities. to appear.

[Ladusaw, 1977] Ladusaw, W. 1977. *Polarity Sensitivity as inherent scope relations*. Ph.D. Dissertation, University of Texas, Austin.

[Lakoff, 1971] Lakoff, George 1971. Semantic interpretation in generative grammar. In Steinberg, D. A. and Jakobovits, L. A., editors 1971, *Semantics: An interdisciplinary reader in philosophy, linguistics, anthropology, and psychology*. Cambridge University Press, Cambridge, UK.

[May, 1977] May, R. 1977. *The Grammar of Quantification*. Ph.D. Dissertation, MIT, Cambridge, MA.

[May, 1985] May, R. 1985. *Logical Form in Natural Language*. The MIT Press.

[Moran, 1988] Moran, D. 1988. Quantifier scoping in the SRI Core Language Engine. In *Proc. ACL-88*, Buffalo, NY. 33–40.

[Poesio, 1991] Poesio, Massimo 1991. Relational semantics and scope ambiguity. In Barwise, J.; Gawron, J. M.; Plotkin, G.; and Tutiya, S., editors 1991, *Situation Semantics and its Applications, vol.2*. CSLI, Stanford, CA. chapter 20, 469–497.

[Poesio, 1993] Poesio, M. 1993. A situation-theoretic formalization of definite descriptions interpretation in plan elaboration dialogues. In Israel, D. and Peters, S., editors 1993, *Situations Theory and its Applications, v.3*. CSLI. To appear.

[Reyle, 1991] Reyle, U. 1991. Dealing with ambiguities by underspecification: A first order calculus for unscoped representations. In Dekker, P. and Stokhof, M., editors 1991, *Proc. of the Eight Amsterdam Colloquium*. University of Amsterdam. 493–512.

[Schubert and Pelletier, 1982] Schubert, L. K. and Pelletier, F. J. 1982. From English to Logic: Context-free computation of 'conventional' logical translations. *American Journal of Computational Linguistics* 10:165–176.

[VanLehn, 1978] VanLehn, Kurt A. 1978. Determining the scope of english quantifiers. Technical Report AI-TR-483, Artificial Intelligence Laboratory, MIT, Cambridge, MA.

[Webber, 1978] Webber, B. L. 1978. A formal approach to discourse anaphora. Report 3761, BBN, Cambridge, MA.

[Woods, 1978] Woods, William A. 1978. Semantics and quantification in natural language question answering. In Yovits, M., editor 1978, *Advances in Computers*. Academic Press, New York. 2–64. Also in [Grosz *et al.*, 1896].

Quasi-Logical Form

Stephen G. Pulman (SRI International and Cambridge University)
Hiyan Alshawi (AT&T Bell Labs)

Quasi-logical form (QLF) is a type of natural-language-oriented higher order logic extended with constructs for representing the meanings of contextually specified items like referring noun phrases, ellipsis, underspecified relations (e.g. possessives) and other similar phenomena. In interpretation, a sentence is analysed into one or more QLFs via compositional syntactic and semantic rules. The resolution of contextually specified information is represented via the monotonic addition of information to the QLF. This has the effect of narrowing down the range of possible interpretations possible: a fully resolved logical form is, roughly, a proposition. However, unresolved QLFs are also coherent logical objects, with a denotation that reflects their partiality.

The process of resolving QLFs, being monotonic, is fully reversible. This has substantial practical advantages, enabling the same grammar to be used to generate sentences from QLFs at different degrees of resolution. For example, the sentence corresponding to a resolved QLF will make explicit the choice of contextual elements that were made in interpreting the unresolved version. The Core Language Engine developed at SRI Cambridge implements this functionality.

The paper presents several examples of the use of QLF in linguistic analysis, in several different applications: in particular, database query and transfer-based translation. We also sketch a formal semantics in terms of supervaluations for a first order version of QLF (due to Alshawi and Crouch).

Dealing with Ambiguities by Underspecification: Construction, Representation and Deduction.

U.Reyle
Institute for Computational Linguistics,
University of Stuttgart,
Keplerstr. 17, 7000 Stuttgart 1

Utterance interpretation is based on a relation between the linguistic form of the utterance and its meaning. Current approaches to natural language understanding assume that the linguistic form of an utterance is given by some syntactic analysis and that the relation between this form and its meaning is characterised by a translation process into some semantic representation structure. In almost all cases this relation is not functional. Whenever semantical ambiguities arise there is a set of meanings associated with a single form. To decide whether some other sentence logically follows from this form it has to be shown that it follows from each of these associated meanings. It is, therefore, the notion of *disjunction* on which such a theory of meaning relies.

In this paper we develop a theory of language meaning that represents scope ambiguities by underspecified structures. The translation into semantic form will thus be functional. The way ambiguities will be represented does not correspond to any of the usual concepts of formalizing ambiguities by means of disjunctions (of completely specified structures). A proof theory is provided that relates these structures directly, without considering cases.

Consider the following argument.

(1) Many a problem about the environment preoccupies every politician. Every politician who many a problem about the environment preoccupies proposes a solution.

Every politician proposes a solution.

The first sentence, P_1, of (1) has two readings. The first reading, P_1^1, is the one where the scope relation of its two NPs corresponds to their linear order; the other reading, P_1^2, gives **every politician** wide scope over **many a problem about the environment**. In the second sentence, P_2, **many a problem about the environment** cannot have wide scope over **every politician**, because the scope of (proper) quantifiers is bound to their local domain - in this case to the relative clause. So, its ambiguity depends only on the way the indefinite **a solution** is interpreted: either as specific indefinite, or as dependent on **every politician**. Therefore the second sentence has two readings, call them P_2^1 and P_2^2. Also for the same reason the conclusion, G, of (1) is two times ambiguous. Thus there are four possible readings of the premiss set of (1), and two readings of the conclusion, G^1 and G^2.

How are we going to relate these two readings of the conclusion to the set of readings of the premisses? What are the inferential properties of ambiguous representations? And can they be characterized by one and only one notion of logical consequence?

We already emphazised that we will not present a theory that *represents* the meaning of ambiguous sentences or texts by the disjunction of their meanings. Thus we will not represent the meaning of the premiss set of (1) by

(2) $$(P_1^1 \wedge P_2^1) \vee (P_1^2 \wedge P_2^1) \vee (P_1^2 \wedge P_2^2) \vee (P_1^1 \wedge P_2^2)$$

but by $P_1^0 \wedge P_2^0$, where P_1^0 and P_2^0 are the *underspecified representations* of the meanings of P_1 and P_2, respectively. The *truth conditions* that our theory assigns to these underspecified representations will, however, guarantee that $P_1^0 \wedge P_2^0$ is true just in case (2) is. And the deduction rules will be such that no recursion to the four cases in (2) is necessary. Our approach thus has not only the advantage to provide a solution to the combinatorial explosion that goes off in any proof that uses (2) as premiss. It also provides a solution to what is called *mapping problem* by [Kempson/Cormack]: $P_1^0 \wedge P_2^0$ is a representation that comes quite close to the combination of the syntactic structures of P_1 and P_2, which clearly isn't the case for (2).

We consider an ambiguous sentence to be true in a model if and only if one of its disambiguations is. And we say that $P_1^0 \wedge P_2^0 \models G$ if every model of $P_1^0 \wedge P_2^0$ is also a model of G. To see that $\models$ is indeed a proper consequence relation the reader easily convinces himself that it satisfies the basic properties a consequence relation should obey,[1] namely reflexivity, transitivity and monotonicity.

One might consider other possibilities as well. For example one may abandon the policy to reckon with the worst as regards the premises and accept the argument one has to prove already if its conclusion follows from some of the readings of its premises, and not necessarily all of them. Or one may define the consequence relation in such a way that the conclusion follows if each – and not only one – of its readings is true in the models that satisfy the premisses. We reject both options (i) because they violate reflexivity and (ii) because we think that a possible deviance from the consequence relation as we defined it is the result of interpretative principles which rely on some definition of coherency of discourses or dialogues. We are not in the position to touch the matter in this paper. We think, however, that if such a definition were available then its effect would be simply to eliminate readings that otherwise were available. Thus the task of drawing inferences will not be affected and so will our consequence relation.

How are we going to represent the meaning of sentences without specifying the scope relations between their quantifiers? There are quite a few proposals in the literature. (See for example [Schubert/Pelletier], [Fenstad et. al.], [Hobbs/Shieber], [Nerbonne].) What all these proposals have in common is the idea of deriving unscoped representations which then may be transformed algorithmically into sets of corresponding disambiguated representations. If the algorithm is simple and effective, there is certainly a benefit to all these approaches. But effective as the algorithm might be, it has the disadvantage of being obligatory. Even though there is an unscoped representation for (1), it has to be translated into a representation of the form

[1] See [Gabbay 91].

(3) $(P_1^1 \wedge P_2^1) \vee (P_1^2 \wedge P_2^1) \vee (P_1^2 \wedge P_2^2) \vee (P_1^1 \wedge P_2^2) \vdash G^1 \vee G^2$

for the purpose of deductive manipulation. As a consequence each of the representations P_1^1, ..., P_2^2 of the different readings of the premise set overspecifies its meaning – and this overspecification has then to be compensated for by taking the disjunction (2) of all possible combinations.

But there is a further disadvantage of the mentioned representations. Consider the sentence

(4) Every professor who recommends a book is admired.

which we may represent à la [Schubert/Pelletier] by

(5) $admired(\forall x(professor(x) \wedge recommend(x, \exists y book(y))))$

What the papers cited have in common is that they do not have a dynamic representation of the meaning of the indefinite **a book** which accounts for the fact that it is interpreted as universally quantified if it has narrow scope with respect to **every professor** and that the other reading assigns it an existentially quantified meaning. This means that the disambiguation algorithm must deal with the problem of choosing the correct quantification type when creating the different meanings.[2]

Thus in the framework of unscoped representations, sentences such as (4) cause the same problem as donkey sentences do. The problem is that the indefinite article is regarded as expressing existence. In DRT, this problem does not occur because the existential import of the indefinite in the cases where it has wide scope in (4) is not a consequence of the meaning of the NP as such, but rather of the way truth is characterized.

The base for our unscoped representations is the separation of information about the structure of a semantic form and of the content of the information bits the semantic form combines. Consider the language of DRSs. DRT represents meaning as the result of an interpretation process in a way that also suits the interpretation of subsequent input. It encodes the semantic connections between successive pieces of sentences or texts – such as, for instance, those produced by pronouns whose anaphoric antecedents occur in ealier sentences – which are largely responsible for the cohesion that distinguishes genuine texts from mere successions of unconnected sentences. The task of establishing the set of semantic connections for a given text relies heavily on the structure of DRSs. In the case of two pieces of discourse being anaphorically linked, for example, the set of possible antecedents is restricted by this structure. Note that the structural information is exploited only when the construction of the meaning representation of that piece of text in which the antecedent occurs has already been accomplished. The constraints that restrict the possible semantic

[2] In [Fenstad et. al.] the problem is approached from a somewhat different perspective: In order to get the non-specific reading of **a book** they analyze the NP **Every professor** who recommends **a book** as binary quantifier.

connections are metalevel constraints. I.e., they are not part of the meaning of linguistic entities, but are used to restrict the set of wellformed DRSs. The language of underspecified DRSs will allow us to express such constraints in the object language. We will, therefore, be able to associate structural constraints declaratively with lexical entries. This does not only apply to constraints that govern anaphoric linkage, but also to constraints that restrict scope ambiguities.

In order to achieve this we express structural information by a language with one predicate $\leq$ that relates individual constants l, called *labels*. The constants are names for DRSs. They are also used to position DRS-conditions at the right place in the hierarchy. This is done by writing l:γ for an occurrence of a DRS-condition γ in a DRS named l.

Given such a separation of structural information and purely linguistic content we are able to indicate that, for example, proper names always end up in the top-level DRS. Assume that the label of the top-level DRS is $l_{\top}$, then we can specify the target position of any proper name π in the lexicon by writing $l_{\top}:\pi$. The scope potential of indefinite descriptions – like **a book** in (4) – is also dealt with in the lexicon: suppose that the meaning component of the indefinite is given by (a set of conditions of the form) $l^{0}:\gamma$; then we can express the fact that the indefinite may take arbitrarily wide scope by adding a $l \leq l^{0}$, where l represents the minimal position l^{0} can occupy.

The construction of meaning representation for a given sentence will then consist in relating names that show up in conditions associated with the phrases to be combined. To say that, for example, the subject of a sentence has to have wide scope over its object we enrich the structural information built up so far by the additional formula $l^{0} \leq l$, where l is the label associated with the meaning of the subject and l^{0} is the label associated with the object. This process of enrichment is characteristic for the construction of meaning representation: information from different sources (syntactic and semantic knowledge as well as knowledge about the world) may be added in a monotonic manner to narrow down the possible range of readings.

The main advantage of this approach is that it comes closer to a representation of meaning that has been thought desirable by scholars from different areas especially from the field of cognitive science. ¿From a cognitive perspective it seems plausable that the recipient of an ambiguous sentence often forms a representation of it that is underspecified with respect to its scope relationships.

References

[Fenstad et. al.] Fenstad, J.E. / Halvorsen, P.-K. / van Benthem, J. (1987) *Situations, Language and Logic*, Reidel Publishing Company, 1987

[Gabbay 91] Gabbay, D. (1991): "Theoretical Foundations for Non-monotonic Reasoning. Part 3: A General Theory of Structured Consequence Relations", ms. London

[Kempson/Cormack] Kempson, R. M. / Cormack, A. (1981) "Ambiguity and Quantification". In: *Linguistics and Philosophy* 4, 259-309

[Hobbs/Shieber] Hobbs, J.R. / Shieber, S.M. (1987): "An Algorithm for Generating Quantifier Scopings", In: *Computational Linguistics* 13

[Nerbonne] Nerbonne, J. (1992), "A Feature Base Syntax/Semantics Interface", ms. Saarbrücken

[Schubert/Pelletier] Schubert, L.K. / Pelletier, F.J. (1982), "From English to Logic: Context-Free Computation of Conventional Logic Translations", in: *Journal of the Association for Computational Linguistics*

Relation of Syntax and Semantics in CCG

(Extended Abstract)

Mark Steedman

University of Pennsylvania

§1 Combinatory Prosody

In [12] and [13], I have argued that the notion of intonational structure formalised by Pierrehumbert, Selkirk, and others, can be subsumed under a rather different notion of syntactic surface structure, that emerges from the "Combinatory Categorial" theory of grammar. This theory engenders surface structure constituents corresponding directly to phonological phrase structure. Moreover, the grammar assigns to these constituents interpretations that directly correspond to what is here called "information structure" - that is, the aspects of discourse-meaning that have variously been termed "topic" and "comment", "theme" and "rheme", "given" and "new" information, and/or "presupposition" and "focus".

Certain syntactic categories and constructions, such as particles like *only* and *even* also "associate with focus" in the sense that they conspire with intonation in utterances like the following to yield a fixed information structure, carrying presuppositions about the background context.

(1) Harry only introduced MARY to Alice.

Here the effect is not only to make *Mary* "focus" or the center of attention, and the fact that *Harry introduced someone to Alice* the background, but also, as Rooth [9] and von Stechow [11] have pointed out, to entail that Harry did not introduce anyone *else* to Alice. The present paper reviews the intonational theory and examines its applications to this problem. The claim is that CCG offers an account of surface structure that is more directly related to a compositional semantics for these particles than more traditional alternatives.

¶1.1 The Problem

Consider the prosody of the sentence *Mary admires corduroy* in the following pair of discourse settings, which are adapted from Jackendoff [4, pp. 260]:

(2) Q: Well, what about the CORduroy? Who admires THAT?

A: (MARy) (admires CORduroy).
H* L L+H* LH%

(3)Q: Well, what about MARy? What does SHE admire?

A: (MARy admires) (CORduroy).
L+H* LH% H* LL%

In these contexts, the main stressed syllables on both *Mary* and *corduroy* receive a pitch accent, but a different one. In the former example, 2, there is a prosodic phrase on *Mary* made up of the pitch accent which Pierrehumbert calls H*, immediately followed by an L boundary. There is another prosodic phrase having the pitch accent called L+H* on *corduroy*, preceded by null or interpolated tone on the words *admires*, and immediately followed by a boundary which is written LH%. (I base these annotations on Pierrehumbert and Hirschberg's [7, ex. 33] discussion of a similar example.)[1] In the second example 3 above, the two tunes are reversed: this time the tune with pitch accent L+H* and boundary LH% is spread across a prosodic phrase *Mary admires*, while the other tune with pitch accent H* and boundary LL% is carried by the prosodic phrase *corduroy* (again starting with an interpolated or null tone).[2]

The meaning that these tunes convey in these contexts is intuitively very obvious.[3] As Pierrehumbert and Hirschberg point out, the latter tune seems to be used to mark some or all of that part of the sentence expressing information that the speaker believes to be *novel to the hearer*. In traditional terms, it marks the "comment" – more precisely, what Halliday called the "rheme". In contrast, the L+H* LH% tune seems to be used to mark some or all of that part of the sentence which expresses information which in traditional terms is the "topic" – in Halliday's terms, the "theme". For present purposes, a theme can be thought of as conveying *what the speaker assumes to be the subject of mutual interest*, and this particular tune marks a theme as *novel to the conversation as a whole*, and as standing in a contrastive relation to the previous theme. (If the theme is not novel in this sense, it receives *no* tone in Pierrehumbert's terms, and may even be left out altogether.)[4] Thus in 3, the L+H* LH% phrase including this accent is spread across the phrase *Mary admires*.[5] Similarly, in 2, the same tune is confined to the object of the open proposition *admires corduroy*, because the intonation of the original question indicates that admireing corduroy *as opposed to some other stuff* is the new topic or theme.

It follows that the position of the pitch accent in the phrase has to do with a further orthogonal dimension of information structure *within both theme and rheme*, corresponding to *the interesting bit* of either information unit. This is what Halliday called "new" infor-

[1]We continue to gloss over Pierrehumbert's distinction between "intermediate" and "intonational" phrases.

[2]The reason for notating the latter boundary as LL%, rather than L is again to do with the distinction between intonational and intermediate phrases.

[3]I do not of course intend to claim that these are the *only* meanings that these tunes can convey.

[4]Here I depart slightly from Halliday's definition. The present proposal also follows Lyons [6] in rejecting Hallidays' claim that the theme must necessarily be sentence-initial.

[5]An alternative prosody, in which the contrastive tune is confined to *Mary*, seems equally coherent, and may be the one intended by Jackendoff. I believe that this alternative is informationally distinct, and arises from an ambiguity as to whether the topic of this discourse is *Mary* or *What Mary admires*. It too is accepted by the rules below.

mation, in contrast to the "given" information accompanied by the null tone. However, the term "new" is an unfortunate one, since the information in question may, if it is part of the theme, have been mentioned before. I shall use the term "focus", which stands in contrast to "background". This usage is illustrated in the following example:

(4) Q: I know that Mary's FIRST degree is in PHYSICS.
But what is the subject of her DOCTORATE?

```
A:   (Mary's      DOCTORATE)      (is in        CHEMISTRY)
                  L+H* LH%                      H* LL%
     Background   Focus           Background    Focus
          Theme                        Rheme
```

Here the theme is *Mary's doctorate*, where the head noun is emphasised because it stands in contrast to another of her qualifications. The rheme is that it *is in chemistry*, where chemistry is emphasised in contrast to another subject.

¶1.2 CONSTITUENCY AND INTONATION.

CCG was originally devised as an account of coordinate phenomena such as Right Node Raising, as in the following sentence:

(5) Mary admires, but I detest, corduroy

According to CCG, simple sentences like *Mary admires corduroy* have not only the traditional surface structure (b), but also the non-standard surface structure (a):

```
(6) a.  Mary      admires   corduroy     b.  Mary      admires   corduroy
      --------- --------- --------         --------- --------- --------
         NP     (S\NP)/NP    NP                 NP     (S\NP)/NP    NP
      -------->T                           -------->T
      S/(S\NP)                             S/(S\NP)
      ------------------>B                           --------------->
            S/NP                                             S\NP
            ------------------>            ------------------->
                    S                                S
```

In fact more complex sentences may have very many alternative non-standard derivations, in addition to the standard one, for each sense-semantic reading. However, all such derivations are guaranteed to deliver an interpretation expressing identical function-argument relations.

It is therefore immediately tempting to equate the two intonational structures exhibited in 2 and 3 with the alternative CCG surface structures exhibited in 6. The earlier papers show how CCG can be made sensitive to prosodic information, so that when prosodic boundaries are present (which of course is frequently *not* the case), then only one of the two alternatives will be present. The papers also show that the modified grammar correctly associates interpretations corresponding to the theme and the rheme with the major constituents of the derivation, and that the account generalises to the (more ambiguous and more frequent) cases in which the discourse-infomational partition is not explicitly marked.

§2 INFORMATION STRUCTURE AND "FOCUS"

The incorporation into the domain of grammar proper of the distinction between theme and rheme, together with the finer distinction between background and focus, means we are in a position to address a wider range of questions in discourse information that have been identified with the notion of "focus" (cf. Jackendoff [4]; Chomsky [1]; [9]). In particular, we are in a position to ask whether this grammar captures phenomena of "focus" that have been identified in semantic accounts of the focussing particles. The paper will investigate the predictions that CCG makes concerning sentences like the following:

(7) a. (John only introduced)(BILL to Sue)
b. = (John introduced)(only BILL to Sue)
c. (John only introduced Bill)(to SUE)
d. (John only introduced)(BILL to SUE)

The paper also examines some apparent counterexamples noted by Rooth, where it appears that information structure of this kind can violate some well-known constraints on syntactic structure, thus threatening the CCG claim of isomorphism between syntax, intonation structure and information structure. These apparent counterexamples concern sentences like the following:

(8) They only asked whether you knew the woman who chairs the ZONING board.

The presupposition here seems to be that *they only asked whether you knew the woman who chairs something.* However, this cannot be a constituent of syntax, because it is in violation of the complex NP constraint:

(9) *Which board did they ask whether you knew the woman who chairs?

Such examples were used by Roth to argue agains a "movement" account of focus. However, it looks at first glance as though they are equally telling against the present theory of discourse information.

It is certainly thie case that the present theory does not allow the sentence to be split into *the zoning board* as rheme and the rest of the sentence as theme. If it did, then the following intonation, in which this illegal constituent is marked as theme by the theme tune would wrongly be allowed:

(10) *(They only asked whether you know the woman who chairs)$_{L+H^*LH\%}$ (the ZONING board)$_{H^*LL\%}$

The paper will argue that the present theory already accounts correctly for 8. The theory implies there are actually *two* backgrounds involved, one belonging to the open proposition or theme, and one stemming from its complement, the rheme. Both backgrounds are marked (or rather, unmarked) by the null tone. Both are presupposed, and therefore affect the p-set, or set of related propositions that are denied. The following example is in fact only one of the information structures that the grammar given earlier will permit.

(11) (They only asked whether you knew)(the woman who chairs the ZONING board).

The evidence is as follows. First, we know that *only* does not simply associate with *the zoning board*, because 8 does not mean the same as the following:

(12) They asked whether you knew the woman who chaired only the ZONING board.

That is, 8 does not entail that they did not ask whether you knew the woman who chaired the zoning board and the parking permit committee, as this analysis would imply. In contrast, 8 *can* mean the same as the following example, which is provided with a contextual question motivating the division concerned:

(13) Which women did they ask whether I knew?
(They asked whether you knew)(only the woman who chairs the ZONING board)

We can tell that this is so by *marking* part of the theme hypothesised in 11 as focus, using the theme tune L+H* LH%:

(14) (They only asked whether you $(\text{KNEW})_{L+H^*LH\%}$)(the woman who chairs (the ZONING board).

In both cases, the p-set or set of negative entailments includes the following, just as it would if the grammar were able to build a monolithic open proposition **They asked whether you knew the woman who chaired ...*:

(15) a. They didn't ask whether you knew the woman who chairs the Parks Committee.
b. They didn't ask whether you knew the man who (co-)chairs the Zoning Board.
(etc.)

The difference is that part each of the entailment stems from the open proposition or theme, and part of it stems from the rheme.

In the longer version of this paper I shall provide a more complete grammar than in earlier papers, together with a semantics for the particles incorporating Mats Rooth's notion of a binary denotation for clauses. The theory generalises to certain cases of "multiple focus" considered by Krifka [5], and exemplified by sentences like the following:

(16) Even HARRY drank only WATER.

Acknowledgements

The research was supported in part by NSF grant nos IRI90-18513 and CISE IIP, CDA 88-22719 , DARPA grant no. N00014-90-J-1863, and ARO grant no. DAAL03-89-C0031.

References

[1] Chomsky, Noam: 1970, 'Deep Structure, Surface Structure, and Semantic Interpretation', in D. Steinberg and L. Jakobovits, *Semantics*, CUP, Cambridge, 1971, 183-216.

[2] Dowty, David: 1988, Type raising, functional composition, and non-constituent coordination, in Richard T. Oehrle, E. Bach and D. Wheeler, (eds), *Categorial Grammars and Natural Language Structures*, Reidel, Dordrecht, 153–198.

[3] Halliday, Michael: 1967, *Intonation and Grammar in British English*, Mouton, The Hague.

[4] Jackendoff, Ray: 1972, *Semantic Interpretation in Generative Grammar*, MIT Press, Cambridge MA.

[5] Krifka, Manfred: 1992, 'A Compositional Semantics for Multiple Focus Constructions', incomplete reference.

[6] Lyons, John: 1977. *Semantics, vol. II*, Cambridge University Press.

[7] Pierrehumbert, Janet, and Julia Hirschberg, 1987, 'The Meaning of Intonational Contours in the Interpretation of Discourse', ms. Bell Labs.

[8] Prince, Ellen F. 1986. On the syntactic marking of presupposed open propositions. Papers from the Parasession on Pragmatics and Grammatical Theory at the 22nd Regional Meeting of the Chicago Linguistic Society, 208-222.

[9] Rooth, Mats: 1985, *Association with Focus*, unpublished PhD dissertation, University of Massachusetts, Amherst.

[10] Selkirk, Elisabeth: 1984, *Phonology and Syntax*, MIT Press, Cambridge MA.

[11] von Stechow, Arnim: 1989, 'Focussing and Backgrounding Operators', Fachgruppe Sprachwissenschaft der Universität Konstanz, Arbeitspapier Nr. 6.

[12] Steedman, Mark: 1990, Structure and Intonation in Spoken Language Understanding, *Proceedings of the 28th Annual Conference of the ACL*, Pittsburgh, June 1990, 9-17.

[13] Steedman, Mark: 1991, Structure and Intonation, *Language*, 68, 260-296.

Session Summary

Robert C. Moore

Wednesday morning, 24-02-93

The three talks in this session were tied together by a common focus on the problem of quantifier scope; the first two presenting representations in which scope could be underspecified, and the third dealing with the factors that resolve scope ambiguity.

The first talk of the day was a presentation on underspecification in quasi-logical forms (QLFs) as used in the Core Language Engine (CLE) at SRI Cambridge, given by Stephen Pulman. This talk was in place of "Monotonic Semantics and Vague Logic" by Hiyan Alshawi, who was unable to attend the Seminar. The discussion of Pulman's presentation focussed on the ability of QLFs to represent various kinds of underspecification of interpretation of natural language utterances. While QLFs did seem to be weaker in this regard than some other formalisms with apparently similar goals, Pulman made the point that many of the types of constraints that were being asked about would be expressed in resolution rules in the CLE, rather than in the QLFs themselves.

The next talk was by Uwe Reyle of the University of Stuttgart on "Underspecification: Construction, Representation and Deduction." In this talk, an approach to underspecification in discourse representation structures was presented, along with a proof system for the underspecified forms. Much of the discussion focussed on the proper definition of logical consequence for such a system. Richmond Thomason expressed reservations about Reyle's definition that for every true interpretation of the premises there was some true interpretation of the conclusion, on the grounds that if a speaker or hearer has in mind a different interpretation of the conclusion, the inference may not be valid. Reyle responded that there were several formal possibilities for the notion of logical consequence in his system, but the one he chose was the only one that possessed the three abstract properties of reflexivity, transitivity, and monotonicity held to be characteristic of a consequence relation.

The final talk of the morning session was by Massimo Poesio of the University of Rochester on "Inferring the Semantic Scope of Operators," which presented an approach based on the hypothesis that operators get assigned their scope as the result of establishing relations of informational inclusion between the "resource situations" containing the objects quantified over by the operators. The discussion focussed mainly on the interplay of structural and pragmatic factors in determining scope.

The morning ended with a general discussion of some topics that seemed to connect the papers. All the papers dealt with issues of scope, but Reyle's paper was quite explicit in assuming that scope could remain unresolved all the way through the process of reasoning with underspecified interpretations, while Poesio dealt with how to resolve such underspecifications. Pulman's talk could be viewed as offering a middle ground, in that like Poesio he viewed interpretation as including the resolution of scope, but like Reyle, his formalism permitted semantically well-defined expressions in which scope is not fully resolved. Another issue raised by both Pulman's and Reyle's talk was that of compositionality. While in both systems the representations for complete utterances were semantically well-defined, it was not clear that all well-formed phrases within an utterance had well-defined semantic interpretations. Finally, there was some discussion of the question of whether the type of underspecification allowed by Pulman and Reyle should be thought of as ambiguity or vaugeness (i.e., disjunction). Some traditional tests for this distinction were recalled that seemed to indicate that these were ambiguities.

Semantic Representation Languages and/or Knowledge Representation Languages

Wolfgang Wahlster
DFKI Saarbrücken

We discuss the conflicting design goals of Semantic Representation Languages (SL) and Knowledge Representation Languages (KL). A KL supports the representation of conceptual knowledge and world knowledge, the retrieval of knowledge (i.e. the efficient indexing of large sets of knowledge units), knowledge acquisition (i.e. inductive inference and learning techniques), various inference services such as classification, subsumption, realisation, forward/backward deduction, abduction, temporal projection and default reasoning; updates and revisions, and the construction of vision systems, expert systems, learning systems, NL systems and robots.

On the other hand, an SL supports the representation of the meaning of an utterance in the discourse context, semantic construction (i.e. getting from a parse to the meaning representation), semantic evaluation (i.e. inferences to resolve ambiguitites, anaphoric references and ellipsis, check presuppositions, compute implicatures), the use as a target language for parsing NL, the use as asource language for the production of NL, layered representations of intermediate processing levels (underspecification, unscoped operators), paraphrase generation, incremental analysis and generation, and the construction of NL dialog, text understanding and translation systems.

In this context, we analyse SLs, like DRT, EKN, DPL, US and SS as semantic frameworks and concrete languages like NLL, QLF and EL. We suggest that typed feature structures are an adequate format for both SR and KR. We show that it is not possible to design a single representation language that supports the need of KR and SR at the same time.

Discourse Representation Theory as a Knowledge Representation Formalism

Hans Kamp
Universität Stuttgart

This talk gives an overview of the different representation languages that have been developed within DRT. Although the primary domain of DRT is the semantics of natural language, the semantic representation languages which have been proposed within this approach for originally strictly NL semantic purposes can also be looked at as representation formalisms tout court, which can be used for the representation of non-linguistic as well as linguistic information. The talk gives a survey of the three formalisms, focussing on their formal semantics (model theory) and their meta-mathematical properties (axiomatizability, decidability, complexity).

Meeting the interlocking needs of LF-computation, deindexing, and inference: An organic approach to general NLU

(Extended Abstract)

Chung Hee Hwang & Lenhart K. Schubert

Department of Computer Science, University of Rochester
Rochester, New York 14627-0226, U. S. A.

The Organism Awaits: A Manifesto

We report here on an approach to theoretical and practical NLU that aspires to be complete and comprehensive, with respect to all the major syntactic, semantic, and pragmatic phenomena encountered in NL texts. Language understanding, on our view, is a highly *organic* phenomenon, in the sense that each facet is strongly dependent on the others. If surface form determines logical form, and logical form determines the ultimate meaning representation, and the ultimate meaning representation determines further conversational (and other) behavior — and if all of these transductions are mediated by inferential use of world knowledge and by a shifting context of salient features of the discourse situation — then surely there is a point where study of isolated features of this organic whole becomes less profitable than an attempt to see it in its entirety.

We think that the need for integration and a global perspective is especially pressing in computational linguistics, since most facets of our problem do not even have a clearly discernible shape *independently* of their relation to other facets. Even syntactic structure, the most accessible aspect of language, is moot, and logical form, semantic representation, knowledge representation, context, and inferential operations are utterly hypothetical, and tightly interlocked. Thus, work done on one issue while simplifying or ignoring the rest is almost certain to go off in quite different directions than work which attempts to keep in mind all constraints and desiderata at once.

Compared to the 70's and 80's, present prospects for principled, integrated NLU are greatly improved. Considerable strides have been taken in our understanding of all aspects of language processing: e.g., grammar, parsing, theories of intention, speech acts and discourse structure, etc. And most importantly from our perspective, new logical frameworks such as DRT, situation semantics, and type and property theories have been developed to address various long-standing semantic conundrums, such as the semantics of attitudes, anaphora, kinds, substances and collections, properties, propositions, events, and tense and aspect.

How, then, do we propose to go about "putting it all together"? For good reason, the centerpiece of our effort has turned out to be the semantic representation/knowledge representation, called *episodic logic* (EL). After all, it is the choice of representation which determines how easily we can derive content from surface from, how fully we can capture the semantic nuances of NL text, and how readily we can perform needed inferences.

EL is a first-order logic that is very expressive, formally interpretable, and easily derived from surface utterances, yet allows efficient inference. It is based on a Montague-style coupling between syntactic form and logical form, while incorporating from situation semantics the idea that sentences describe situations (events, states, episodes, eventualities, etc.). Moreover, all of this is implemented in at least a preliminary way. The EPILOG system [11], the computer implementation of EL, makes quite complex inferences, e.g., with utterances from the TRAINS domain [1], telex reports for aircraft mechanical problems in the ARMS project [10], and excerpts from *Little Red Riding Hood* story [12]. These experiments show that the inference chain is straight-forward, despite the richness of the logic, and that the knowledge it is based on is uncontrived (it corresponds quite directly to English sentences, and each individual piece of knowledge arguably is formulated at a maximally general level, rather than being particularized to the needs of a specific story). We now briefly review EL and its role in a comprehensive, modular approach to NLU. We first describe semantic representation and inference rules in EL, and then explain how one can get episodic logical form from English input. For formal semantics of the logic, see [6, 7].

The Episodic Logical Form

Among the most important features of EL are its liberal ontology and NL-like *expressiveness*. These features make it easy to derive EL-translations of English sentences, while also providing a basis for concise, easily understood inferences. EL syntax allows lambda abstraction, restricted quantifiers, modal operators and propositional attitudes, predicate modifiers, kind forming operators, nominalization operators, action abstraction, DRT-like anaphoric variables, generic conditionals, and other non-standard constructs. Most importantly, however, it makes use of episodic (situational) variables in the representation of episodic sentences, making implicit temporal and causal relationships between situations explicit.

To give an idea of the syntax, we show below an ELF representation of the sentence "An object bumped into the left wing of the airplane, causing it to get a crack" (with certain simplifications).[1]

(1) (The x: [x airplane]
(The y: [[y ((attr left) wing)] $\wedge$ [y part-of x]]
($\exists e_1$: [[e_1 before *Now1*] $\wedge$ [($\exists w$:[w object] [w bump-into y]) ** e_1]]
($\exists e_2$: [e_1 cause-of e_2] [($\exists z$:[z crack] [y get z]) ** e_2]))))

This sentence introduces two episodes: e_1, an episode of "some object bumping into the left wing y," and e_2, an episode of "wing y getting a crack." Note the clause [e_1 cause-of e_2]

[1] *Now1* in the following formula is a term that will be replaced with an appropriate nonindexical term (possibly a clock time) at the deindexing stage. attr is an operator that transforms a predicate into an *attr*ibutive predicate modifier. Also, note that we use infix notation for readability and restricted quantification of the form (Q α: Φ Ψ), where Q is a quantifier, α is a variable, and Φ and Ψ are formulas. Readers are referred to [6] for details of EL syntax.

which shows the causal relationship between the two episodes. $[\Phi ** \eta]$ means formula Φ characterizes (or, completely describes) episode η. A weaker form of this modal operator is '$*$'. $[\Phi * \eta]$ means that Φ partially describes (or, is true in) η. This is similar to the $\models$ ("support") relation of situation semantics [2, 3], except that we are relating sentence intensions (partial mappings from situations to truth values), rather than "infons," to situations.

The following example illustrates attitude predicates and kind abstraction. `That`, `K`, and `Ka` below are nominalization operators that form a proposition from a sentence intension, a kind of property from a predicate, and a kind of action from an action predicate, respectively.

(2) ($\exists e_1$: [e_1 at-about *Now2*]
[[Mary believe (That ($\exists e_2$: [e_2 before e_1]
[[(Ka swim) prevent-from Mary (Ka (gain (K weight)))] ** e_2]
** e_1]))

Mary believes that swimming prevented her from gaining weight.

There is also a nominalization operator `Ke` that forms a *kind of event* from a sentence intension.

Next shown is an example of probabilistic conditionals, i.e., extensional generic conditionals (with some simplifications).

(3) ($\exists x$:[[x aircraft] $\wedge$ [(age x year) $<$ 3]] ($\exists y$:[y crack][y located-on x]))
$\rightarrow_{x,\,y,\,.8}$ ($\neg$ [y due-to (K corrosion)])

If an aircraft that is less than 3 years old has a crack,
usually the crack is not due to corrosion.

The rule says, roughly, that in at least 80% of the situations in which the antecedent is true, the consequent will also be true (x, y are controlled variables). Note the DRT-like treatment of indefinites in the rule; that is, existential variable y occurs outside its quantifier scope. This is allowed in `EL` thanks to the parameter mechanism that carries the variable binding beyond the scope of variables. (The interpretation of such free variables is done much as in DRT [8].)

Space limitations prevent a discussion of semantics, but we should remark that unlike situation semantics, `EL` is based on an ontology that allows *possible situations*. These are much like "partial possible worlds," in that symbols are assigned partial extensions (and antiextensions) relative to them.

Inference Rules in Episodic Logic

The main inference rules we have developed are *Rule Instantiation* (`RI`) and its dual *Goal Chaining* (`GC`), which resemble forward and backward chaining rules in expert systems. These rules are formally stated below.

<u>`Rule Instantiation (RI)`</u>

Rule instantiation, which is heavily used in input driven inference, allows arbitrarily many minor premises to be matched against arbitrarily deeply embedded subformulas of a rule. It subsumes *modus ponens* and *modus tollens*, but can also instantiate generic conditionals. In the unit probability version, with just one minor premise ("fact"), the `RI` rules are:

$$\frac{R^{\Gamma}(\Phi),\ F^{+}(\Psi)}{R^{\Gamma}_{\sigma}(\neg(F^{+}_{\sigma}(\bot)))} \qquad \frac{R^{\Gamma}(\Phi),\ F^{+}(\Psi)}{F^{+}_{\sigma}(R^{\Gamma}_{\sigma}(\top))}$$

where σ unifies Φ, Ψ. R stands for 'Rule', and F for 'Fact'. $\top$ and $\bot$ are truth and falsity respectively. The $+$ and $-$ signs are intended to indicate positive and negative occurrence of the embedded Φ, Ψ formulas being unified. Unification is defined in a way that allows substitution for explicitly quantified, "matchable" variables. A variable in a rule or fact is matchable if it is bound by a positively occurring universal quantifier or negatively occurring existential quantifier. The first rule is *sound* if Ψ contains no unmatchable free variables which are bound in F as a whole. The second rule is sound if Φ contains no unmatchable free variables which are bound in R as a whole. So in particular, the first rule is sound if F contains only constants and top-level universal (hence matchable) variables. The rules work similarly for generic conditionals.

`Goal Chaining (GC)`

Goal chaining, which dominates goal-driven inference, is a pair of very general chaining rules. Chaining from rule consequents to antecedents is a special case. The following are the `GC` rules in the unit probability case:

$$\frac{R^{+}(\Phi),\ ?G^{+}(\Psi)}{?G^{+}_{\sigma'}(\neg(R^{+}_{\sigma'}(\bot)))} \qquad \frac{R^{+}(\Phi),\ ?G^{+}(\Psi)}{?\neg(R^{+}_{\sigma'}(\neg(G^{+}_{\sigma'}(\top))))}$$

where σ^0 "antiunifies" Φ, Ψ (i.e., with positive existentials and negative universals in G regarded as matchable). R stands for 'Rule', and G for 'Goal'. The first rule is *sound* if Φ contains no unmatchable free variables which are bound in R as a whole. The second rule is sound if Ψ contains no unmatchable (e.g., top-level universal) variables which are bound in G as a whole.

The general version of `GC` allows arbitrarily many subsidiary knowledge base facts to be invoked in the process of chaining from the given goal to a subgoal. There is also another class of goal-directed methods that consists of standard natural deduction rules such as proving a conditional by assuming the antecedent and deriving the consequent, or proving a universal by proving an arbitrary instance of it.

These rules are partially implemented in the EPILOG system [11], a hybrid reasoning system combining efficient storage and access mechanism, forward and backward chaining, agenda-driven control structure, and multiple "specialists" for taxonomies, temporal reasoning, etc.

Computing Episodic Logical Form

A crucial feature of EL with respect to the goal of building general NLU systems is the ease with which EL-representations are derived from surface syntax. The initial translation from phrase structure to the preliminary indexical logical form (LF) is accomplished with GPSG-like syntactic and semantic rules; the final nonindexical episodic logical form (ELF) is obtained by simple recursive deindexing rules. Such a transformation is essential because, to be useful for *inference*, a situational logic must be nonindexical. Our deindexing algorithm uniformly handles tense, aspect, and many temporal adverbials and their interaction, and brings the context information into the logical form, removing context dependency.

For example, the logical form of sentence

> *"A mechanic repaired Crack8 yesterday"*

is easily computed using the following (somewhat simplified) lexical and phrase structure rules, annotated with corresponding semantic rules.

> DET[indef] ← *a* ; ∃
> N ← *mechanic* ; mechanic
> NP ← DET N ; < DET^o N^o >
> NP ← *Crack8* ; Crack8
> V[past] ← *repaired* ; < past repair >
> VP[trans] ← V NP ; (V^o NP^o)
> ADV[ep-mod] ← *yesterday* ; (during *Yesterday*)
> ADVL[post-mod] ← ADV[ep-mod] ; $\lambda P \lambda x$ ((adv-e ADV^o) [x P])
> VP ← VP ADVL[post-mod] ; (ADVL^o VP^o)
> S ← NP VP ; [NP^o VP^o]

As mentioned, angle brackets indicate unscoped expressions. Applying these rules gives us the initial, unscoped logical form ULF (a) below, which will then be scoped as LF (b). *Yesterday* is an indexical term, and *past* is an indexical sentence operator. After the deindexing step which introduces an episodic variable through tense operator *past*, we get ELF (c) below (with some simplifications).

(a) ((adv-e (during *Yesterday*)) [< ∃ mechanic> <past repair> Crack8])

(b) (past ((adv-e (during *Yesterday*)) ($\exists x_1$:[x_1 mechanic][x_1 repair Crack8])))

(c) ($\exists e_1$:[e_1 before *Now3*]
  [((adv-e (during (date (1993 2 23))))
    [($\exists x_1$:[x_1 mechanic][x_1 repair Crack8])) ** e_1])

The transformation from (b) to (c) is carried out by the tense-aspect deindexing rules of EL that use *tense trees* as components of discourse contexts. The mechanism is compositional in that operators *past, fut, perf, etc.,* contribute separately and uniformly to the meanings of their operand formulas, driving the generation and traversal of tense trees in deindexing. As an example, we show the Past-rule below.

Past: $(\text{past } \Phi)_{\mathtt{T}} \leftrightarrow (\exists e_{\mathtt{T}}{:}[[e_{\mathtt{T}} \text{ before } \mathtt{Emb}_{\mathtt{T}}] \wedge [\mathtt{Last}._{\mathtt{T}} \text{ orients } e_{\mathtt{T}}]]\ [\Phi_{\circ\ \mathtt{T}} ** e_{\mathtt{T}}])$

Tree transform: $(\text{past } \Phi) \cdot \mathtt{T} = \text{``}(\Phi \cdot (\circ \swarrow \mathtt{T}))$

T denotes a tense tree; $e_{\mathtt{T}}$ a "new" episode symbol. $\mathtt{Emb}_{\mathtt{T}}$, $\mathtt{Last}_{\mathtt{T}}$, $\swarrow \mathtt{T}$, $\circ\mathtt{T}$, etc., are easily computed functions on the tense tree T (see [5, 13] for details). The recursively deindexed Φ is taken to *characterize* the new episode $e_{\mathtt{T}}$, which is predicated to be *before* the embedding episode, e.g., *utterance episode*. Tense trees also provide the "points of orientation" (cf., [9]), such as the reference point for a perfect episode or the immediately preceding past episode in a succession of simple past-tensed sentences, and the "*orients*" predication captures this.

By meaning postulates, we can get from (c) the following (skolemizing $E7/e_1$):

[E7 before *Now3*]
[E7 during (date 1993 2 23)]
$[(\exists x_1{:}\,[x_1 \text{ mechanic}]\,[x_1 \text{ repair Crack8}]) * \text{E7}]$

This shows that the information in the *past* operator and in the time adverbial ("*yesterday*") are essentially interpreted *conjunctively*, much as in Dowty's system [4]. However, Dowty's system is not able to identify orienting episodes, and its final LFs still involve indexical operators like *past* while our ELFs are completely deindexed.

Conclusion

We think there is cause for optimism about the possibility of constructing theoretical and computational frameworks for full NLU. Our own efforts in that direction have led to a rather well-integrated conception of syntax, LF, knowledge representation, context, and inference, and of the interfaces linking these into an organic whole. The conception is not yet complete or fully "debugged," but it is sufficiently far along to have provided a basis for diverse start-up implementations. Unlike most past implemented NLU and inference systems, these implementations strenuously avoid cutting corners in syntax, LF computation, and most of all, knowledge representation and inference. Thus, we have reason to regard the theoretical framework and the implementations as a solid and extensible basis for further work toward the ultimate goal of general NLU.

References

[1] J. Allen and L. K. Schubert. The TRAINS project. Technical Report TR 382, Dept. of Comp. Sci., U. of Rochester, Rochester, NY, 1991.

[2] J. Barwise. *The Situation in Logic.* CSLI, Stanford, CA, 1989.

[3] K. Devlin. *Logic and Information.* Cambridge U. Press, Cambridge, UK, 1991.

[4] D. Dowty. Tense, time adverbs and compositional semantic theory. *Linguistics and Philosophy*, 5:23–55, 1982.

[5] C. H. Hwang and L. K. Schubert. Tense trees as the 'fine structure' of discourse. In *Proc. 30th Annual Meeting of the ACL*, pages 232–240. Newark, DE, June 29–July 2, 1992.

[6] C. H. Hwang and L. K. Schubert. Episodic Logic: A comprehensive semantic representation and knowledge representation for language understanding. Technical report, Dept. of Comp. Sci., U. of Rochester, Rochester, NY, 1993.

[7] C. H. Hwang and L. K. Schubert. Episodic Logic: A situational logic for natural language processing. In *Situation Theory and its Applications, V. 3*, Stanford, CA, In print. CSLI.

[8] H. Kamp. A theory of truth and semantic representation. In J. Groenendijk, T. Janssen, and M. Stokhof, editors, *Formal Methods in the Study of Language*. Mathematical Centre-tracts, U. Amsterdam, Amsterdam, Netherlands, 1981.

[9] G. Leech. *Meaning and the English Verb (2nd ed)*. Longman, London, UK, 1987.

[10] A. Namioka, C. H. Hwang, and S. Schaeffer. Using the inference tool EPILOG for a message processing application. *Internat. J. of Expert Systems*, In print.

[11] S. Schaeffer, C. H. Hwang, J. de Haan, and L. K. Schubert. *The User's Guide to* EPILOG (Prepared for Boeing Co. under Purchase Contract W-278258). Edmonton, Canada, 1991.

[12] L. K. Schubert and C. H. Hwang. An Episodic knowledge representation for narrative texts. In *Proc. 1st Inter. Conf. on Principles of Knowledge Representation and Reasoning (KR '89)*, pages 444–458, Toronto, Canada, May 15-18, 1989.

[13] L. K. Schubert and C. H. Hwang. Picking reference events from tense trees: A formal, implementable theory of English tense-aspect semantics. In *Proc. Speech and Natural Language, DARPA Workshop*, pages 34–41, Hidden Valley, PA, June 24-27, 1990.

Natural Language Semantics and Compiler Technology

John Nerbonne†, Joachim Laubsch‡, Abdel Kader Diagne†, Stephan Oepen†
†Deutsches Forschungszentrum für Künstliche Intelligenz
Stuhlsatzenhausweg 3
D-6600 Saarbrücken 11, Germany
{nerbonne,diagne,oe}@dfki.uni-sb.de
‡Hewlett-Packard Laboratories
1501 Page Mill Road
Palo Alto, CA 94303-0971, USA
laubsch@hplabs.hp.com

Keywords: Semantics, Software Tools

Precis Semantic representation languages (SRLs) and programming languages (PLs) share design goals: first, in both cases we seek modules in which not only conventional surface representations (print form) but also underlying data structures are important. Here we need general tools allowing the printing and reading of expressions. Second, these modules need to cooperate extensively with foreign modules, so that the importance of interface technology (compilation) is paramount; and third, both PL compilers and semantic modules need "inferential" facilities for transforming (simplifying) complex expressions in order to ease subsequent processing.

But the most important parallel is the need in both fields for tools which are useful in combination with a variety of concrete languages—general purpose parsers, printers, simplifiers (transformation facilities) and compilers. This arises in PL technology from (among other things) the need for experimentation in language design, which is again parallel to the case of SRLs.

Using a PL-based approach, we have implemented $\mathcal{NLL}$, a public domain software package for computational natural language semantics. Several interfaces exist for both grammar modules and applications, using a variety of interface technologies, including especially compilation.

1 Introduction: Design Goals

The focus of this paper is the design and IMPLEMENTATION of semantic representation languages (SRLs). Given the need of such modules to represent natural language meanings, we assume that they should apply linguistic semantics, the specialized study of natural language meaning. But because of the focus on design and implementation, we examine quite generally the uses to which such modules may be put, abstracting away from details which distinguish such superficially distinct approaches as generalized quantifier theory (GQT), discourse representation theory (DRT), situation theory, or dynamic logic.

The appropriate design for any module can only be determined by close analysis of the uses to which it is to be put. We do not consider applications for semantics in providing test tools for linguistic hypotheses or in adding constraints to recognition tasks ([YHW+89]), because these provide less clear design criteria. But the numerically most important group of applications, that of understanding and generation, give rise to fairly clear requirements. Independent semantics modules are used with these applications for semantic representation, inference, and in order to support meaning-related processing—disambiguation, resolution, and speech act management. The importance of these points confirms the good sense of current practice in the field—that of viewing the main task of the semantic module as the implementation of a linguistic semantic theory (with selected AI enhancements for resolution and disambiguation). But we suggest that insufficient attention is paid to the following four goals, which we focus on below:

- modularity—independence from syntax and application
- modifiability—for experimentation
- interface support—for mapping into and out of module
- support for independent use (reader, printer, tracer)

These prompt us to a comparison to programming language technology and compiler construction.

2 Analogy to Programming Languages

It is axiomatic that modern PLs should meet the last four goals listed in the design goals for SRLs. Standard introductions ([ASU86]) detail how a programming language syntax is specified in definitions independent of specific machines and environments (modularity) which are, moreover, easily modifiable given tools for parsing (parser-generators) and printing. The parsers automatically created from language specifications take well-formed strings as input and produce abstract syntax trees (like linguistic parse trees) as output. Modern tools also provide printers (unparsers) which reverse the process: given an abstract syntax tree, they produce a print form ([FWH92], 85ff).

Just as a modular SRL must interface to more than one application, a programming language needs to be able to run on different machines. In the latter case, this is accomplished by compiling: the abstract syntax trees produced by the PL's parser are transformed into (the abstract syntax trees representing) expressions of another lower-level language (often a machine-specific assembler language). While it is obvious how this scheme enables generality vis-à-vis translation targets, it may not be as immediately apparent that the level of abstract syntax likewise facilitates generality toward translations sources: in the case of a PL such as C, we not only compile FROM C into various assembler languages on the basis of transformations of abstract syntax, but we likewise compile other (normally more specialized languages) INTO C. The SRL correspondence is the use of compiler technology to translate into SRLs, viz., in syntax/semantics interfaces (in NLU) or application/semantics interfaces (in generation). Further details below.

Some of the transformations performed by compilers are not simple translations into target languages, but rather transformations to alternative structures in the source language (cf. [ASU86], 592ff), or immediate evaluation of parse structures (cf. "translation during parsing" [ASU86], 293-301). The use of these techniques suggests an implementation for some inference facilities for SRLs—those arising from equivalence rules.

Our core thesis: PL technology may profitably be applied to the design of SRL software. Figure 1 summarizes the points at which immediate borrowings from PL technology seem apt means to SRL goals. We turn now to a brief description of $\mathcal{NLL}$, an SRL implemented using PL technology. We then illustrate how SRLs profit from this approach using a concrete and fully implemented example.

Goals	PL	SRL
modularity	independent definition (BNF)	
tools	parser, printer	parser (%), printer
modifiability	parser-generator	
mappings in, out	compiler	
inference	program transformation	resolution, backward-chaining ...

Figure 1: Design goals common to PLs and SRLs plotted against "standard" solutions in the two areas. The analogy suggests filling the gaps for standard solutions for SRLs by using PL solutions: language specification tools for definition together with parser-generators to provide the SRL reader, and compiler technology for interfaces to semantics modules. Finally, program transformation techniques suggest a simple implementation for at least some inference rules.

3 $\mathcal{NLL}$

$\mathcal{NLL}$ is an SRL which borrows heavily from linguistic semantics in order to provide representational adequacy, using, e.g., on the one hand work from generalized quantifier theory and on the other from the logic of plurals. [LN91] and [Ner92] present an overview of $\mathcal{NLL}$ and the background linguistic and model-theoretic ideas, which will not be repeated here. For the sake of understanding examples below, we note that atomic formulas in $\mathcal{NLL}$ are composed of a predicate together with a set of role-argument pairs, e.g.

'Anterist ships to Hamburg' `ship(source:a goal:h)`

The $\mathcal{NLL}$ formula may also be read: '*a*' plays the role of agent and '*h*' that of goal in some shipping situation. An advantage of identifying arguments via roles rather than positions (as is customary in predicate logic) is that one can sensibly use the same predicate, e.g., '`ship`', with various numbers of arguments; thus even though something must also play a '`theme`' role in this situation (what is shipped), it need not be expressed in the role-coded set of arguments. Cf. [Ner92] for formal development.

3.1 Data Structures and Basic Tools

Following the PL lead, we begin with a formal syntactic specification of $\mathcal{NLL}$ in a form usable by a parser-generator.[1] We use Zebu ([Lau92b]), a public-domain tool in Common LISP.[2]

Zebu grammar specifications consist of a set of RULES, each of which specifies a SYNTAX for a grammatical category and an ACTION to be taken by the parser when the category is found. These specifications are easily modified in case extensions, variations or even substantial modifications of the language become interesting. In addition to syntax rules,

[1]We are concentrating here on the more recent $\mathcal{NLL}$ implementation; an earlier implementation in REFINE ([LN91]) is no longer the focus of our efforts, even though we continue to maintain it for its usefulness in rapid prototyping. REFINE is a trademark of Reasoning Systems, Palo Alto.

[2]Zebu was originally developed in Scheme by William Wells.

Zebu grammars may also contain lexical restrictions ([Lau92b],15) needed for generating a lexical analyzer (which, however, is not used in $\mathcal{NLL}$). ¿From the $\mathcal{NLL}$ grammar, Zebu generates an LALR(1) parser ([ASU86],§ 4), which is the $\mathcal{NLL}$ reader. The reader immediately supports experiments with the semantics module by easing the creation of semantic data structures. The Zebu grammar compilation process detects any inconsistencies or ambiguities in the grammar definition.

Zebu goes beyond the capabilities of parser-generators such as UNIX yacc in further optionally generating (automatically) the definition of a DOMAIN, a hierarchy of data structures (LISP structures) for abstract syntax. If this option is chosen, then Zebu defines a structure type for each expression type; the structure for a given expression has as many fields as the expression has subexpressions (e.g., Predicate and Role-Argument-Pairs). On the basis of the domain Zebu then also generates an "unparser", in this case the $\mathcal{NLL}$ printer (which in turn may be called by the LISP printer).

At this point we have implemented a representational system with thorough dual-access: we may process its elements through manipulations of either surface or abstract syntax. For example, $\mathcal{NLL}$ structures are created either from strings or from constructor functions (and occasionally in a mixture of approaches); similarly, one could specify logical inference rules or operations such as substitution either as a string operation or as an operation on LISP structures (or both). Important processing submodules have been implemented using both surface and abstract levels. We examine these now.

3.2 Interfaces and Compilations

As noted in § 1, an important SRL task is communication with a variety of NLP modules, including at least syntax, context (resolution), dialogue management, and application system. How can the PL approach to SRL design support interface construction?

The dual access provided by the PL approach already allows an interesting degree of freedom. For example, the opportunity to create $\mathcal{NLL}$ via constructor functions allows the implementation of a syntax-semantics interface of the sort suggested by [JK90]—in which the syntax/semantics interface is constituted by a set of generic constructor functions attached to syntactic rules (and therefore nonterminal nodes). $\mathcal{NLL}$ has been employed this way in a syntax/semantics interface in an extensive NLP system ([NP87]). This is appropriate when relatively complex structures are created in a series of simple increments. Alternatively, one may invoke the $\mathcal{NLL}$ reader to create $\mathcal{NLL}$, and an interface from the COSMA appointment manager (cf. below) to $\mathcal{NLL}$ (for generation) invokes the reader extensively. This made the single-step creation of complex structures much simpler.

But given the relatively easy access to abstract syntax trees provided by the Zebu reader, the construction of interfaces through genuine compilation (transformation based on abstract syntax) is also feasible. $\mathcal{NLL}$'s basic scheme of compilation is TREE REWRITING ([ASU86],572ff). An abstract syntax tree is traversed, and at each node, each of a sequence of REWRITE RULES is applied. A rewrite rule abstractly takes the form:[3]

[3]In the REFINE implementation of $\mathcal{NLL}$, rules are also concretely of this form (cf. [Rea90],§§ 3.7.5–3.7.7). In current work we are trying to incorporate abstract specifications of $\mathcal{NLL}$ rewrite rules in the Zebu-based version. In the current implementation, rewrite rules are LISP functions.

meta-syntactic-pattern $\Rightarrow$ *replacement-node*

A rewrite rule checks whether a *meta-syntactic-pattern* is satisfied at the current node, and returns the replacement node together with a boolean flag indicating whether the rule has fired: (*replacement-node*,δ?:bool). In case we are translating from one abstract syntax to another, then the meta-syntactic-pattern describes a node in the source language, while the replacement node belongs to the translation target language. The traversal routine replaces the current node with the replacement-node in case the rule has fired.

The top-down tree-rewriting algorithm PREORDER-TRANSFORM inputs a tree t and a sequence $\langle r_1 \ldots r_n \rangle$ of rewrite rules. It then traverses the tree in preorder ([AHU83],78-9), and at each node, attempts to rewrite using each of the rules r_i. If any rule in the sequence fires, then the entire sequence is tried again, until no rules fire. Then the traversal continues, until the leaf nodes of the tree. The algorithm is attractive because it reduces the tree-transformation problem to the specification of transformations on local subtrees. An analogous POSTORDER-TRANSFORM invokes sequences of rewrite rules in a bottom-up traversal of the abstract syntax tree.

In addition to optimized control routines for tree-transformation $\mathcal{NLL}$ provides a library of access and manipulation functions (for substitution, construction, simplification) to support the transformation process. [Lau92a] reports on the required transformations for $\mathcal{NLL}$ compilers to SQL and to the New Wave task language, and [Oep93] presents transformations needed in COSMA, a distributed system for appointment management ([NOeS92]). Cf. Figure 2 for a further example of a compilation to $\mathcal{NLL}$(NLL2FS), this time to feature description languages.

Compilation is normally an effective translation technique because it abstracts away from irrelevant details of the concrete syntaxes of target and source languages. It is especially appropriate: (i) when communication between modules is limited (e.g., when modules run on separate machines or in separate processes, so that communication is limited to strings); (ii) when the nature of target data structures is unknown or unspecified; or (iii) when there is minor variability in targets (e.g., different versions of the same programming language or query language).

3.3 Inference Rules

In § 3.2 above, we applied rewrite rules of the following form to compilation:

meta-syntactic-pattern $\Rightarrow$ *replacement-node*

where *meta-syntactic-pattern* describes a source language node, and *replacement node* a translation target language structure. But we may also formulate rewrite rules where both the metasyntactic pattern and replacement node refer to $\mathcal{NLL}$ structures, yielding inference rules for $\mathcal{NLL}$.

For example n-ary conjunctions are normalized according to the following patterns.

flatten	`AND{p AND{q r}} --> AND{p q r}`
identity element	`AND{p true} --> AND{p}`
constant	`AND{p false} --> false`
single element	`AND{p} --> p`

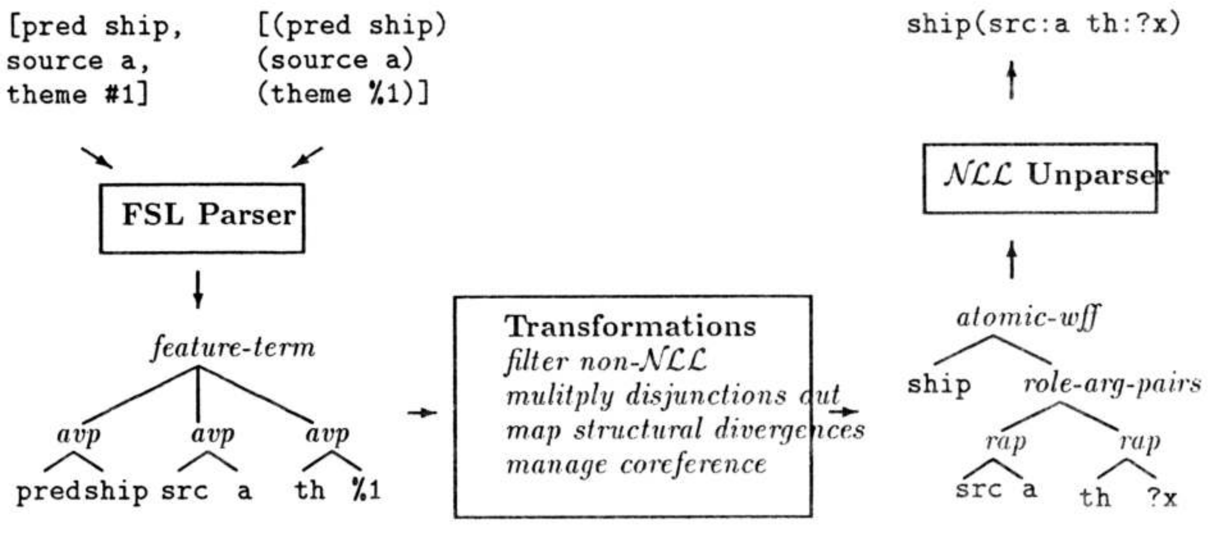

Figure 2: **Compiler-based approach to semantics modularity.** In order to employ $\mathcal{NLL}$ not only in the ASL speech understanding project (cf. Görz 1992) but also in the COSMA dialogue application (cf. Oepen 1993)—involving both different NL parsers and different feature description languages (ASL and UDiNe)—FS2NLL, a compiler from feature structure (fs) languages to $\mathcal{NLL}$ was developed. The two fs languages were parsed into the same abstract syntax, which was filtered and transformed into $\mathcal{NLL}$ abstract syntax, which is "unparsed" in a final step. A further borrowing from compiler technology, a symbol table, was needed to manage coreferences in the fs language ('%1,#1'). Diagne and Nerbonne 1992 report on the ASL interface. Several components FS2NLL are useful for reverse translation, and NLL2FS was implemented to provide semantic feedback to parsing and search in the ASL speech understanding system and for generation in the COSMA dialogue system. NLL2FS has just completed implementation.

Of course, as we indicated in § 3.2 above, these are not specified declaratively in the Zebu implementation, but instead implemented as LISP functions. A declarative (and easier to use) specification is the subject of ongoing work. [Lau89] presents many more inferences rules as these were used in a database interface from $\mathcal{NLL}$. There is no further inference mechanism in $\mathcal{NLL}$, so that the inference system is weak. [ENP92] sketches more powerful rewrite systems which proceed from the same basis, perhaps a direction for future work.

3.4 Compiling into $\mathcal{NLL}$

The examples of compilation above all involve compiling FROM $\mathcal{NLL}$ into another language; but it is natural to apply the same techniques when translating INTO $\mathcal{NLL}$, and similar advantages accrue here. Compilation is particularly cost-effective when one wishes to construct interfaces to several modules with similar representations. In this case relatively minor modifications in grammar specifications may be all that is required to obtain further interfaces. Cf. Figure 2 for an example in which two feature description languages were compiled into $\mathcal{NLL}$. These were used in different systems with a minimum of system-specific modification.

4 Conclusions and Prospects

In addition to the benefits noted above, the PL approach to SRLs enables relatively easy reverse translations, which are needed between components in dialogue systems, and which are useful in architectures emphasizing component feedback ([G̈92]). The ability to define interfaces on the basis of strings allows not only the tighter definition of interfaces noted above, but also flexibility in architectures—e.g., no assumptions need to be made about whether SRL modules run in the same address space or even on the same machine as their clients. (It even proves useful for modules in the same LISP image because it eliminates some packaging sensitivities.)

Perhaps the most interesting benefit has come in the ability to implement and therefore compare alternative SRLs relatively easily, and we have been able to conduct practical experiments compiling a language of situation semantics ([GP90],20)—which incorporates a mechanism for representing indefinites with scope (like DRT) but underspecified quantificational force, into the language of generalized quantifiers (from GQT, cf. [BC81])

The focus of our current work is the provision of a declarative (and easier) specification for transformation rules, both as a term-rewriting and as a general transformation system.

References

[AHU83] Alfred Aho, John Hopcroft, and Jeffrey Ullman. *Data Structures and Algorithms.* Addison Wesley, 1983.

[ASU86] Alfred Aho, Ravi Sethi, and Jeffrey Ullman. *Compilers: Principles, Techniques and Tools.* Addison Wesley, 1986.

[BC81] Jon Barwise and Robin Cooper. Generalized quantifiers and natural language. *Linguistics and Philosophy*, 4(2):159–219, 1981.

[ENP92] Norbert Eisinger, Andreas Nonnengart, and Axel Präcklein. Termersetzungssysteme. In K.H.Bläsius and H.-J.Bürckert, editors, *Deduktionssysteme*, chapter 3.4, pages 126–149. Oldenbourg, München, 1992. 2nd, Revised Edition.

[FWH92] Daniel Friedman, Mitchell Wand, and Christopher Haynes. *Essentials of Programming Languages.* McGraw-Hill, New York, 1992.

[G̈92] Günter Görz. Kognitiv orientierte architekturen für die sprachverarbeitung. Technical Report ASL-TR-39-92, Fachbereich Informatik, Universität Erlangen-Nürnberg, 1992.

[GP90] Jean Mark Gawron and Stanley Peters. *Anaphora and Quantification in Situation Semantics.* CSLI Lecture Notes, Stanford University, 1990.

[JK90] Mark Johnson and Martin Kay. Semantic abstraction and anaphora. In Hans Karlgren, editor, *Proceedings of COLING-90*, pages 17–27, Helsinki, 1990. COLING.

[Lau89] Joachim Laubsch. Logical form simplification. STL report, Hewlett-Packard, December 1989.

[Lau9 Joachim Laubsch. The semantics application interface. In Hans Haugeneder, editor, *Applied Natural Language Processing.* ??publisher, ??address, 1992.

[Lau9 Joachim Laubsch. **Zebu**: A tool for specifying reversible lalr(1) parsers. Technical report, Hewlett-Packard Laboratories, Palo Alto, CA, July 1992.

[LN91 Joachim Laubsch and John Nerbonne. An overview of $\mathcal{NLL}$. Technical report, Hewlett-Packard Laboratories, Palo Alto, July 1991.

[Ner9 John Nerbonne. $\mathcal{NLL}$ models. Research report, DFKI, Saarbrücken, 1992.

[NP8 John Nerbonne and Derek Proudian. The HP-NL system. Technical report, Hewlett-Packard Labs, 1987.

[NOe! Günter Neumann, Stephan Oepen and Stephen Spackman. Design and Implementation of the COSMA System. Research report, DFKI, Saarbrücken, 1993.

[Oep9 Stephan Oepen. Mapping from $\mathcal{NLL}$ to cosma-ir. In *EACL 93*, 1993. submitted to EACL student session.

[Rea9 Reasoning Systems Inc., Palo Alto. REFINE *User's Guide*, May 1990.

[YHW Sheryl Young, Alexander Hauptmann, Wayne Ward, Edward Smith, and Philip Werner. High level knowledge sources in usable speech recognition systems. *Communications of the ACM*, 32(2):183–194, 1989.

E-theory as a semantic tool for natural language reconstruction and processing

Bernd Mahr
Technische Universität Berlin

Much of the reconstruction of NL in logic terms is beyond conventional model- theoretic semantics. Using ϵ-structures and ϵ-logic (first order logic with two binary relations ϵ and $=$) a much richer model theory can be give. Namely it allows for non-wellfounded sets, the representation of circular phenomena naturally appearing in NL, such as self-application, self-reference, and self-similarity. Using ϵ-structures, model-theoretic semantics can be given to powerful type theoretic disciplines. In this way a uniform description technique using type propositions of the form $\alpha{:}\lambda$ together with model-theoretic semantics can be established.

The talk introduces the framework of ϵ-theory, discusses the construction of λ-models, the expressiveness and model-construction of ϵ_T-logic (first order logic withn intensional equality and truth predicates for propositions, formulas arte the individual terms), and self-similar models with NLR and NLP applications. Finally, the question is raised to what extent the approach applies to NL.

Belief Revision: Applications in NLP and Formal Properties

Bernhard Nebel
DFKI Saarbrücken

Espen Vestre
Universität des Saarlandes

The theory of belief revision is concerned with the following problem: If a given body of information K is given, and ϕ is a new piece of knowledge that is in conflict with K, how can we change K in order to incorporate ϕ?

Belief Revision has at least two applications in natural language processing. The most obvious application comes from the need to revise the knowledge base(s) of a natural language system. But Belief Revision may also be used to define the *semantics* of certain sentences.

Gärdenfors ([Gärdenfors 78], [Gärdenfors 88]) suggests a model where each K is a *theory* (closed under logical consequence). He then defines the semantics for conditionals through the "Ramsey Test":

$$A > B \in K \quad \textit{iff} \quad B \in K * A$$

In [Hansson 89] it is argued that K should rather be a *base* of a theory. The advantage of using bases are twofold. First, representing and manipulating theory bases is much easier than doing this on logically closed theories. Second, it becomes possible to distinguish between basic and derived beliefs. Take, for instance, the two bases $B_1 = \{a\}$ and $B_2 = \{a, a \vee b\}$. Although these bases are logically equivalent, i.e., they represent the same theory, base revision will usually lead different results. Revising the bases with $\neg a$ will lead to $\{\neg a\}$ for B_1 and to $\{\neg a, a \vee b\}$ for B_2 (under most base revision schemes).

A semantics for counterfactuals based on base revisions was introduced by [Veltman 76] and [Kratzer 79] and is known as *Premise Semantics*. A premise semantics is *based* on a *premise function* P, a function which for every possible world w gives a set of propositions (i.e. a *base*) $P(w)$. In belief revision terminology Premise Semantics can be reformulated in the following way:

$$i \in [\![A > B]\!] \quad \textit{iff} \quad \text{the } \textit{Simple Base Revision} \text{ of } P(i) \text{ with } A \text{ implies } B$$

[Ginsberg 86] reinvented this semantics in a syntactic version, but failed to define the acceptance criterions for formulas with embedded counterfactuals.

In natural language processing—and, more generally, in a computational context—base revision seems to be the more reasonable way to model belief revision. First of all, it seems to be more tractable from a computational point of view. Second, it has the above mentioned advantage that one can represent the difference between basic and derived beliefs. However, it has been argued that belief revision shall be an syntax-independent operation, i.e., not dependent on the syntactic form of the representation of the beliefs [Winslett 88, Katsuno and Mendelzon 89]. It is possible, however, to draw a very tight connection between base revision and theory revision [Nebel 92]. It can be shown that every base revision operation *generates* a theory revision operation. Further, these theory revision operations satisfy most of the rationality postulates that have been formulated for such operations [Alchourrón *et al.*85, Gärdenfors 88].

As has been argued, base revision is computationally more tractable than theory revision, since we do not have to take into account all derived propositions. Nevertheless, there is the question of how much costs we have to pay. As has been shown [Nebel 92, Eiter and Gottlob 92], even in the propositional case, base revision (in many forms) is more expensive than ordinary propositional derivability. In the general, the problem of solving whether a given proposition is derivable from a revised belief base is complete for Π_2^p, i.e., it is located on the second level of the polynomial hierarchy. This means that the problem contains two interacting sources of computational complexity, namely, propositional derivability and the problem of finding out whether a proposition is in one (or all) candidate solutions. Hence, special cases that can be solved in polynomial time (assuming P$\neq$NP) can only be obtained if both sources are eliminated, e.g., by restricting the logic to Horn logic and by simplifying the "identification" problem [Nebel 92, Eiter and Gottlob 92].

References

[Alchourrón *et al.*85] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50(2):510–530, June 1985.

[Eiter and Gottlob 92] Thomas Eiter and Georg Gottlob. On the complexity of propositional knowledge base revision, updates, and counterfactuals. *Artificial Intelligence*, 57:227–270, 1992.

[Gärdenfors 78] Peter Gärdenfors. Conditionals and changes of belief. In I. Niiniluoto and R. Tuomela, eds., *The Logic and Epistomology of Scientific Change (Acta Philosophica Fennica* 30 (2-4), pages 189–203, 1978.

[Gärdenfors 88] Peter Gärdenfors. *Knowledge in Flux.* The MIT Press, Cambridge(Mass.), 1988.

[Ginsberg 86] Matthew L. Ginsberg. Counterfactuals. *Artificial Intelligence*, 30:35–79.

[Hansson 89] Sven O. Hansson. New Operators For Theory Change. *Theoria*, 55:114–132, 1989.

[Katsuno and Mendelzon 89] Hirofumi Katsuno and Alberto O. Mendelzon. A unified view of propositional knowledge base updates. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, pages 1413–1419, Detroit, MI, August 1989. Morgan Kaufmann.

[Kratzer 79] Angelika Kratzer. Conditional Necessity and Possibility. In R. Bäuerle, U. Egli and A. von Stechow, eds., *Semantics from Different Points of View*, Springer-Verlag, Berlin, pages 117–147, 1979.

[Nebel 92] Bernhard Nebel. Syntax-based approaches to belief revision. In P. Gärdenfors, editor, *Belief Revision*, volume 29 of *Cambridge Tracts in Theoretical Computer Science*, pages 52–88. Cambridge University Press, Cambridge, UK, 1992.

[Veltman 76] Frank Veltman. Prejudices, Presuppositions and the Theory of Counterfactuals. In J. Groenendijk and M. Stokhof, eds., *Amsterdam Papers in Formal Grammar*, vol. 1, Centrale Interfaculteit, University of Amsterdam.

[Winslett 88] Marianne S. Winslett. Reasoning about action using a possible models approach. In *Proceedings of the 7th National Conference of the American Association for Artificial Intelligence*, pages 89–93, Saint Paul, MI, August 1988.

PSYCHOLOGICAL INVESTIGATIONS INTO DEFAULT LOGIC

Francis Jeffry Pelletier (jeffp@cs.ualberta.ca)
Departments of Philosophy and Computing Science

Renée Elio (ree@cs.ualberta.ca)
Department of Computing Science
The University of Alberta
Edmonton, Alberta, Canada
T6G 2H1

I. Introduction

1.1 Background

Default reasoning occurs whenever the evidence available to the reasoner does not guarantee the truth of the conclusion being drawn; that is, does not deductively *force* the reasoner to draw the conclusion under consideration. ('Force' in the sense of being required to do it *if* the reasoner is to be logically correct). For example, from the statements 'Linguists typically speak more than three languages' and 'Kim is a linguist', one might draw the conclusion, by default, 'Kim speaks more than three languages'. What is meant by the phrase 'by default' is that we are justified in making this inference because we have no information which would make us doubt that Kim was covered by the generalization concerning linguists or would make us think that Kim was an abnormal linguist in this regard. Of course, the inference is not *deductively* valid: it is *possible* that the premises could be true and the conclusion false. So, one is not *forced* to draw this conclusion in order to be logically correct. Rather, it is a conclusion that we draw "by default"—the type of conclusion we draw in the ordinary world and ordinary circumstances in which we find ourselves.

Formally speaking, the term 'non-monotonic reasoning' refers to argumentation in which one uses certain information (the *premises* of the argument) to reach a conclusion, but where it is possible that later adding some further information to those very same premises (i.e., adding another premise to the existing premises of the argument) could make one want to *retract* the original conclusion. (Sometimes this might even make us wish to conclude the opposite of the original conclusion). Symbolically, it is a case of nonmonotonic reasoning if one is willing to make the inference $\{P_1,P_2,\ldots P_n\}\vdash C$ but is unwilling to make the inference $\{P_1,P_2,\ldots P_n,P_{n+1}\}\vdash C$. The catch-phrase of non-monotonic reasoning is "that new information makes one withdraw previously-made inferences without withdrawing any background premises".

It is easily seen that the informal notion of default reasoning manifests a type of non-monotonic reasoning. In the above example, for instance, we concluded that Kim spoke at least three languages. But were we to add to our list of premises the further fact that Kim graduated from NewWave University, which we know has revoked all language requirements, we then would wish to withdraw the earlier conclusion. Thus, default reasoning is a species of non-monotonic reasoning.

1.2: Default Reasoning in Artificial Intelligence

A certain trend in artificial intelligence (AI) has been to investigate the construction of "knowledge bases" -- which is envisaged as a type of database, but where there is somehow a lot of common-sense reasoning ability built in. (See Reiter 1987 for a summary). The analogy being pursued by such investigators is with actual human information storage. In humans, not all of our information about the world is stored in an explicit form in our minds; instead much of it is *implicit* in that it is a consequence of, or follows from, other information which *is* stored explicitly. And we can "call it up" by performing inferences on what we *have* explicitly stored. One of the ways we infer these implicit pieces of knowledge is by default reasoning. Most researchers in this area think this is the most pervasive form of reasoning in our everyday life. In AI there are two general schools of thought as to how to characterize formally this default reasoning. (i) Our background information is associated with a "likelihood" parameter and our new conclusions are modulated accordingly. The most common version of this type is to assign our beliefs or information states a "probability", and to draw conclusions in accord with a probabilistic logic. Another version of this type employs "fuzzy logic." (ii) Our background information is characterized as being "typically true", and we draw conclusions that are treated as 'true', or 'true in the absence of information to the contrary'. The difference between the two versions of default reasoning amounts to whether we explicitly represent our lack of deductive conclusiveness in some *quantitative* way, always attaching some evaluation to each of our beliefs and propagating evaluations to our newly-drawn conclusions. Method (i) enjoins us to do so; whereas method (ii) instead tells us to treat each belief as *qualitatively* true but to be prepared to withdraw conclusions in the face of new information.

In the AI literature, drawing conclusions in accordance with method (i) is usually called "uncertain inference", whereas drawing them in accordance with method (ii) is usually called "nonmonotonic reasoning". Uncertain Inference has been quite thoroughly studied in Philosophy, Psychology, and Management Science, both as a theory and as an account of the extent to which people actually follow those inferences claimed to be correct by the theory. The same cannot be said about the qualitative method (ii), nonmonotonic reasoning. Here the theoretical foundations have been investigated mostly in Computer Science, but without a consensus on what is the correct underlying logical structure. Indeed, there is even much doubt as to which inferences *ought* to be sanctioned and which *ought* to be disallowed.

1.3: The Non-Monotonic Benchmark Problems

To ameliorate the problem of not knowing which non-monotonic inferences should be considered valid, Lifschitz (1989) published a list of 25 "Nonmonotonic Benchmark Problems" which gave the answers generally accepted by researchers in the area. All future formal accounts of nonmonotonic reasoning were supposed to be able to yield these answers. There are different types of Benchmark Problems in Lifschitz's list, corresponding to the different areas in which default reasoning is seen as useful to the AI community. In this paper we will report results of our empirical investigation of one group of these problems, the "Basic Default Inference" problems; and will mention some pilot data surrounding a second group of problems, the "Inheritance Inference" problems.

1.4: The Justification of Non-Monotonic Formalisms in AI

Non-monotonic theoreticians believe that it is *correct* to make default inferences. It is *not* a mistake on people's part, nor is it a matter of "having to do *something*, anything, in the face of insufficient information." Rather, it is right and proper to make such inferences: not only is this what people *in fact* do, but it is what people (and computer simulations of them) *ought* to do.

Indeed, much of the initial motivation for investigating non-monotonic reasoning was to characterize more accurately how people in fact reason. The background idea was that people use their reasoning abilities "to get along in the world" very well; if computers could only emulate people in this regard they too would be able to live up to their promise. Here is a typical quotation from the non-monotonic reasoning literature on this viewpoint:

> In everyday life, it seems clear that we, human beings, draw sensible conclusions from what we know and that, on the face of new information, we often have to take back previous conclusions, even when the new information we gathered in no way makes us want to take back our previous assumptions ... It is most probable that intelligent automated systems will have to perform the same kind of (nonmonotonic) inferences. [Kraus, *et al* 1990]

The point we wish to emphasize and to which we wish to draw the reader's attention is this: *Despite the acknowledgement by the artificial intelligence community that the goal of developing non-monotonic systems owes its justification to the success that ordinary people have in dealing with default reasoning, there has been no investigation into what sorts of default reasoning ordinary people in fact employ.* Instead, artificial intelligence researchers rely on their introspective abilities to determine whether or not their system ought to embody such-and-so inference. And even the 25 Benchmark Problems of Lifschitz were formulated with absolutely no regard to whether ordinary people in fact do reason in the way prescribed! Given the central place that the Benchmark Problems occupy in the field—they are the minimal abilities that any artificial system must embody—it certainly behooves the Cognitive Science community to investigate whether or not non-monotonic reasoning as conceived by the formal non-monotonic reasoning community actually conforms to the promises and goals initially held out for it, especially those promises that it would mirror actual processes that people engage in. Principal to this is the question of the extent to which the non-monotonic community has accurately characterized "ordinary", "common-sensical" reasoning. After all, the example non-monotonic arguments cited in the literature were all invented *ex nihilo* by theorists. None of them empirically investigated the extent to which *real* "ordinary, commonsensical reasoning" agreed with the examples. Yet, it was precisely such an agreement that was the entire *raison d'etre* for the enterprise. We therefore pose the question: Do people actually reason in the manner prescribed by the non-monotonic logic community?

2. Results

In this paper, we present empirical results on the plausible conclusions people actually draw, given a Benchmark problem to solve. According to non-monotonic theories, the existence of, and information about, an an exception object for a default rule (an object which does not obey the rule) should have no bearing on conclusions drawn about any *other* object when using that rule. Our results indicate this is not true for human reasoners: their plausible conclusions are influenced by specificity of information available about exception objects, and how similar an exception is to the object about which they are asked to reason. (A side result of our experiments suggests an "Asimov Effect": people are unwilling to allow robots to draw default conclusions they allow themselves to make.) The inheritance of default properties by some object also seems influenced by the knowledge of other subclasses and their relation to the default property. One interpretation of our findings is that people do not reason about defaults and exceptions as formal rules to be manipulated: they will put themselves in "problem-solving mode" and integrate *all* the information presented in some way to generate a plausible conclusion. This suggests that it may be difficult to develop robust models of non-monontonic reasoning without some goal-directed component, that in turn determines what kind of information is relevant to the application of a default rule.

REFERENCES

Doyle, J. (1979) "A Truth Maintenance System" *Artificial Intelligence* **12**: 231-272.

Ginsberg, M.L. (1987) *Readings in Nonmonotonic Reasoning* (Los Altos, CA: Morgan Kaufmann).

Johnson-Laird, P.N. & R.M.J. Byrne (1991) *Deduction* (Hillsdale,NJ: Lawrence Erlbaum).

Kraus, S., D.Lehmann, M.Magidor (1990) "Nonmonotonic Reasoning, Preferential Models, and Cumulative Logics" *Artificial Intelligence* **44**: 167-207.

Lifschitz, V. (1989) "25 Benchmark Problems in Nonmonotonic Reasoning, v.2.0" in M.Reinfrank, J. de Kleer, M.Ginsberg *NonMontonoic Reasoning* (Berlin: Springer) 202-219.

McCarthy, J. (1977) "Epistemological Problems of Artificial Intelligence" *Proceedings of the 5th International Joint Conference on Artificial Intelligence* (Cambridge, MA) 1038-1044.

McCarthy, J. (1980) "Circumscription -- a Form of Non-Monotonic Logic" *Artificial Intelligence* **13**: 27-39.

McCarthy, J. (1986) "Applications of Circumscription to Formalizing Common Sense Knowledge" *Artificial Intelligence* **28**: 89-116.

McDermott, D. & J.Doyle (1980) "Non-monotonic Logic, I" *Artificial Intelligence* **13**: 41-72.

Moore, R.C. (1985) "Semantical Considerations on Nonmonotonic Logic" *Artificial Intelligence* **25**: 75-94.

Pollock, J.(1990) "A Theory of Defeasible Reasoning" *International Journal of Intelligent Systems*

Reiter, R. (1978) "On Reasoning by Default" *Proceedings of TINLAP-2* (Univ. Illinois: Assoc. for Computational Linguistics) 210-218.

Reiter, R. (1980) "A Logic for Default Reasoning" *Artificial Intelligence* **13**: 81-132.

Reiter, R. (1987) "Nonmonotonic Reasoning" *Annual Reviews of Computer Science* **2**: 147-187.

Rosch, E. (1978) "Principles of Categorization" in E.Rosch & B.B.Lloyd (eds.) *Cognition and Categorization* (Hillsdale,NJ: Lawrence Erlbaum) 27-48.

Tversky, A. & D. Kahneman (1983) "Extensional Versus Intuitive Reasoning: The Conjunction Fallacy in Probability Judgment" *Psychological Review* **90**: 293-315.

Non-Monotonic Formalisms
for Natural Language Semantics

Richmond H. Thomason
Intelligent Systems Program
University of Pittsburgh
Pittsburgh, PA 15260
U.S.A.

August 10, 1993

Extended Abstract of a Paper to be Presented at

The Dagstuhl Seminar on Semantic Formalisms
in Natural Language Processing

February 23–26, 1993

This is a working draft:
version of <u>August 10, 1993</u>.
The material is volatile; do not quote.
Comments welcome.

Non-Monotonic Formalisms for Natural Language Semantics

1. Introduction

I will try to do three things in this paper. First, I want to situate certain problems in natural language semantics with respect to larger trends in logicism, including:

(i) Attempts by positivist philosophers earlier in this century to provide a logical basis for the physical sciences;

(ii) Attempts by linguists and logicians to develop a "natural language ontology" (and, presumably, a logical language that is related to this ontology by formally explicit rules) that would serve as a framework for natural language semantics;

(iii) Attempts in artificial intelligence to formalize common sense knowledge.

Second, I want to propose an extension of Montague's framework, and to illustrate some of its applications in the semantics of words. Third, I'll distinguish between the problem of specifying a phenomenon like natural language semantics (or providing conditions on an adequate solution), and providing an account of the relevant procedures. I'll try to explain briefly why I think that we are a long way from an adequate, unified account of semantics for the purposes of natural language processing.

All of this is actually a tall order. The pressure of trying to fill it given limits of space, and—even more—my present state of ignorance or confusion on many of the central issues will affect the quality of the exposition and the views themselves. But I hope that you will bear with me, since I think that a partially successful and fragmentary attempt at the larger project may convince some members of my audience that the natural language semantics community and the subgroup of the AI community interested in formalizing common sense knowledge have a great deal in common, and much to learn from one another. To a large extent, the rhetorical goal of this paper is to make this message seem plausible and exciting, and to provide semanticists with a number of relevant references from the literature in AI.

2. Logicism

The material in this and the subsequent two sections of this abstract is lifted from [Thomason 91].

Let X be a topic of inquiry. X logicism is the view that X should be presented as an axiomatic theory from which the rest can be deduced by logic. *Science logicism* is expressed as an ideal in Aristotle's *Organon*. But Aristotle's logic is far too weak to serve as a means of representing Aristotelian science, and logicism was in effect remained impracticable until the 17th century, when a separation of theoretical science from common sense simplified the task of designing an underlying logic.[1]

[1] Despite the simplification, of course, a workable formalism did not begin to emerge until the 19th century.

There is a moral here about logicism. X logicism imposes a program: the project of actually presenting X in the required form. But for the project to be feasible, we have to choose a logic that is adequate to the demands of the topic. If a logic must involve explicit formal patterns of valid reasoning, the central problem for X logicism is then to articulate formal patterns that will be adequate for formalizing X.

The fact that very little progress was made for over two millennia on a problem that can be made to seem urgent to anyone who has studied Aristotle indicates the difficulty of finding the right match of topic and formal principles of reasoning. Though some philosophers (Leibniz, for one) saw the problem clearly, the first instance of a full solution is Frege's choice of mathematical analysis as the topic, and his development of the *Begriffschrift* as the logical vehicle. It is a large part of Frege's achievement to have discovered a choice that yields a logicist project that is neither impossible nor easy.

I will summarize some morals. (1) Successful logicism requires a combination of a formally presented logic and a topic that can be formalized so that its inferences become logical consequences. (2) When logicist projects fail, we may need to seek ways to develop the logic. (3) Logic development can be difficult and protracted.

3. Extensions to the empirical world

The project of extending Frege's achievement to the empirical sciences has not fared so well. Of course, the mathematical parts of sciences such as physics can be formalized in much the same way as mathematics. Though the metamathematical payoffs of formalization are most apparent in mathematics, they can occasionally be extended to other sciences.[2] But what of the empirical character of sciences like physics? One wants to relate the systems described by these sciences to observations.

Rudolph Carnap's *Aufbau*[3] was an explicit and ambitious attempt to extend mathematics logicism to science logicism, by providing a basis for formalizing the empirical sciences. The *Aufbau* begins by postulating elementary units of subjective experience, and attempts to build the physical world from these primitives in a way that is modeled on the constructions used in Frege's mathematics logicism.

Carnap believed strongly in progress in philosophy through cooperative research. In this sense, and certainly compared with Frege's achievement, the *Aufbau* was a failure. Nelson Goodman, one of the few philosophers who attempted to build on the *Aufbau*, calls it "a crystallization of much that is widely regarded as worst in 20th century philosophy."[4]

After the *Aufbau*, the philosophical development of logicism becomes somewhat fragmented. The reason for this may have been a general recognition, in the relatively small community of philosophers who saw this as a strategically important line of research, that the underlying logic stood in need of fairly drastic revisions.[5]

This fragmentation emerges in Carnap's later work, as in the research of many other logically minded philosophers. Deciding after the *Aufbau* to take a more direct, high-

[2] See [Montague 62].

[3] [Carnap 28].

[4] [Goodman 63], page 545.

[5] I can vouch for this as far as I am concerned.

level approach to the physical world, in which it was unnecessary to construct it from phenomenal primitives, Carnap noticed that many observation predicates, used not only in the sciences but in common sense, are "dispositional"—they express expectations about how things will behave under certain conditions. A malleable material will deform under relatively light pressure; a flammable material will burn when heated sufficiently. It is natural to use the word 'if' in defining such predicates; but the "material conditional" of Frege's logic gives incorrect results in formalizing such definitions. Much of [Carnap 36-37] is devoted to presenting and examining this problem.

Rather than devising an extension of Frege's logic capable of solving this problem, Carnap suggests dropping the requirement that these predicates should be explicated by definitions. This relaxation makes it harder to carry out the logicist program, because a natural way of formalizing dispositionals is forfeited. But it also postpones a difficult logical problem, which was not, I think, solved adequately even by later conditional logics in [Stalnaker & Thomason 70] and [Lewis 73]. Such theories do not capture the notion of normality that is built into dispositionals: a more accurate definition of 'flammable', for instance, is 'what will *normally* burn when heated sufficiently'. Thus, logical constructions that deal with normality offer some hope of a solution to Carnap's problem of defining dispositionals. Such constructions have only become available with the development of nonmonotonic logics.

4. Linguistic logicism

Though work in philosophical logic and its applications continues the logicist tradition to some extent, logicist projects are largely out of fashion in philosophy, and much of the work on projects of this sort is being carried on in other disciplines.

In linguistics, a clear logicist tradition emerged from the work of Richard Montague, who (building to a large extent on Carnap's work in [Carnap 56]) developed a logic he presented as appropriate for *philosophy logicism*. Montague's extreme logicist position is stated most clearly in a passage in [Montague 69].

> It has for fifteen years been possible for at least one philosopher (myself) to maintain that philosophy, at least at this stage in history, has as its proper theoretical framework set theory with individuals and the possible addition of empirical predicates. ... [But] philosophy is always capable of enlarging itself; that is, by metamathematical or model-theoretical means—means available within set theory—one can "justify" a language or theory that transcends set theory, and then proceed to transact a new branch of philosophy within the new language. It is now time to take such a step and to lay the foundations of intensional languages.[6]

Montague's motivation for expanding his logical framework is the need to relate empirical predicates like 'red' to their nominalizations, like 'redness'. He argued that many such nominalizations denote properties, that terms like 'event', 'obligation', and 'pain' denote

[6] [Montague 74], pages 156–157.

properties of properties, and that properties should be treated as functions taking possible worlds into extensions. The justification of this logical framework consists in its ability to formalize certain sentences in a way that allows their inferential relations with other sentences to be captured by the underlying logic.

Philosophers other than Montague—not only Frege, but Carnap in [Carnap 56] and Church in [Church 51]—had resorted informally to this methodology. But Montague was the first to see the task of *natural language logicism* as a formal challenge. By actually formalizing the syntax of a natural language, the relation between the natural language and the logical framework could be made explicit, and systematically tested for accuracy. Montague developed such formalizations of several ambitious fragments of English syntax in several papers, of which [Montague 73] was the most influential.

The impact of this work has been more extensive in linguistics than in philosophy.[7] Formal theories of syntax were well developed in the early 70s, and linguists were used to using semantic arguments to support syntactic conclusions, but there was no theory of semantics to match the informal arguments. "Montague grammar" quickly became a paradigm for some linguists, and Montague's ideas and methodology have influenced the semantic work of all the subsequent approaches that take formal theories seriously.

As practiced by linguistic semanticists, language logicism would attempt to formalize a logical theory capable of providing translations for natural language sentences so that sentences will entail one another if and only if the translation of the entailed sentence follows logically from the translation of the entailing sentence and a set of "meaning postulates" of the semantic theory. It is usually considered appropriate to provide a model-theoretic account of the primitives that appear in the meaning postulates.

This methodology gives rise naturally to the idea of "natural language metaphysics," which tries to model the high-level knowledge that is involved in analyzing systematic relations between linguistic expressions. For instance, the pattern relating the transitive verb 'bend' to the adjective 'bendable' is a common one that is productive not only in English but in many languages. So a system for generating derived lexical meanings should include an operator ABLE that would take the meaning of 'bend' into the meaning of 'bendable'.

To provide a theory of the system of lexical operators and to explain logical interactions (for instance, to derive the relationship between 'bendable' and 'deformable' from the relationship between 'bend' and 'deform'), it is important to provide a model theory of the lexical operators. So, for instance, this approach to lexical semantics leads naturally to a model-theoretic investigation of ability,[8] a project that is also suggested by a natural train

[7]It is hard to explain the lack of philosophical interest in the project. Recent linguistic work in natural language metaphysics is loosely connected to earlier attempts to exploit language as a source of insights into the nature of distinctively human patterns of thought about what might be called the common sense world. I am thinking here of works like [Cassirer 55] and of [Jespersen 65]. Both of these projects grew out of a rich philosophical tradition: Cassirer's work, in particular, is firmly rooted in the European Kantian tradition. And, of course, there has been much work in the phenomenological tradition—which, however, has been much less formal.

[8]That the core concept that needs to be clarified here is ability rather than the bare conditional 'if' is suggested by cases like 'drinkable'. 'This water is drinkable' doesn't mean 'If you drink this water it will have been consumed'. (Of course, ability and the conditional are related in deep ways.) I will return

of thought in logicist AI.[9]

Theories of natural language meaning that, like Montague's, grew out of theories of mathematical language, are well suited to dealing with quantificational expressions, as in

4.1. *Every boy gave two books to some girl.*

In practice, despite the original motivation of his theory in the semantics of word formation, Montague devoted most of his attention to the problems of quantification, and its interaction with the intensional and higher-order apparatus of his logical framework.

But those who developed Montague's framework soon turned their attention to these problems, and much of the later research in Montague semantics—especially David Dowty's early work in [Dowty 79] and the work that derives from it—concentrates on semantic problems of word formation, which of course is an important part of lexical semantics.[10]

5. Formalizing common sense

To a certain extent, the motives of the common sense logicists overlap with Carnap's reasons for the *Aufbau.* The idea is that the theoretical component of science is only part of the overall scientific project, which involve situating science in the world of experience for purposes of testing and application; see [McCarthy 84] for explicit motivation of this sort. For extended projects in the formalization of common sense reasoning, see [Hobbs & Moore 90] and [Davis 90].

The project of developing a broadly successful logic-based account of semantic interrelationships among the lexical items of a natural language is roughly comparable in scope with the project of developing a high-level theory of common sense knowledge. Linguists are mainly interested in explanations, and computer scientists are (ultimately, at any rate) interested in implementations. But for logicist computer scientists who have followed McCarthy's advice of seeking understanding before implementing, the immediate goals of the linguistic and AI projects are not that different.

And—at the outset at least—the subject matter of the linguistic and the computational enterprise are remarkably similar. The linguistic research motivated by lexical decomposition beginning in [Dowty 79] and the computational research motivated largely by problems in planning (or practical reasoning) both lead naturally to a focus on the problems of representing change, causal notions, and ability.

6. Formalizing nonmonotonic reasoning

briefly to the general problem of ability in Section 7.5, below.

[9]See, for example, [Thomas *et al.* 90].

[10]This emphasis on compositionality in the interpretation of lexical items is similar to the policy that Montague advocated in syntax, and it has a similar effect of shifting attention from representing the content of individual lexical items to operators on types of contents. But this research program seems to require a much deeper investigation of "natural language metaphysics" or "common sense knowledge" than the syntactic program, and one can hope that it will build bridges between the more or less pure logic with which Montague worked and a system that may be more genuinely helpful in applications that involve representation of and reasoning with linguistic meaning.

See [Ginsberg 88b] for an a good guide to the field of nonmonotonic reasoning and its development.

Among the available theories of defeasible reasoning that could be applied in lexical semantics, I find circumscription the most congenial to use in attempting to apply these theories to problems of natural language semantics, for the following reasons.

(i) Circumscription is based on second-order logic, and this second-order foundation can easily be generalized to Intensional Logic.[11] Montague's apparatus does conflict with preferences that McCarthy has expressed from time to time about how to formalize intensional constructions, but this philosophical disagreement does not seem to be an obstacle to the absorption of circumscription into the framework of Intensional Logic.

(ii) The more sophisticated versions of circumscription provide an explicit formalism for dealing with abnormalities.[12] I believe that such a formalism is needed in the linguistic applications.

7. Case studies

I'll illustrate the use of nonmonotonic formalisms in semantics with several case studies. In these studies, I'm merely trying to motivate the use of a nonmonotonic formalism in the semantics of words, and to suggest how it might be applied to some of the immediate problems that arise in this area. At the date of this version, I have not tried to work out the details. At this point, the abstract will become much more sketchy.

7.1. The *-able* suffix

The natural way to define 'x is water-soluble' is

7.1. If x were put in some water, then x would dissolve in the water.

So at first glance, it may seem that the resources for carrying out the definition that Carnap found problematic will be available in a logic with a subjunctive conditional. But suppose it so happens that if one were to put this salt in some water, it would be this water, and this water is saturated with salt. The fact that the salt would not then dissolve is no reason why the salt should count as not water-soluble. This and other such thought experiments indicate that what is wanted is not the bare subjunctive conditional, but a "conditional normality" of the sort that is used in some nonmonotonic formalisms.[13]

In a circumscriptive framework, normality is obtained by conditions on a number of abnormality predicates, which are then circumscribed, or minimized relative to certain background assumptions, in obtaining models of the nonmonotonic theory. Events are an appropriate locus for organizing these abnormality predicates not only in the case of dissolving, but in many other cases of interest for purposes of lexical decomposition.

It is convenient to think of events as classified by a system of event types, from which abnormalities and other features are inherited. In treating the dissolving example, I will make the following assumptions.

[11]See [Thomason 90] for a brief description and application of the combined theory (to discourse, rather than to lexical semantics).

[12]See [Lifschitz 89].

[13]See [Boutilier 92], [Asher & Morreau 91].

7.i. There is an event type ϕ of *put-in events*.[14] Associated with this type (and, by inheritance, with events falling under it) there is a container *container*(ϕ) and a thing moved *movee*(ϕ).

7.ii. The event type ϕ has a subtype ϕ_1, in which *container*(ϕ) is a quantity of water and *movee*(ϕ) is a quantity of salt. There is an abnormality predicate associated with ϕ_1.

7.iii. There is an event type ψ of *dissolving events*. I assume that associated with this event type (and, by inheritance, with events falling under it) there is an inception, a body, and a culmination (where the first two are events and the last is a state); also, an associated medium *medium*(ψ) and a thing dissolved *dissolvee*(e); also an abnormality predicate.

It will follow from general considerations about the event type ψ that if a ψ-normal event of this type occurs, its associated culmination state will also occur. (See the remarks below on telicity.)

[14]This event type itself has a decompositional analysis, but we can ignore that for purposes of the example.

Given this information about event types, the sort of analysis that I currently favor for dissolving amounts to this.

7.iv. Every ϕ_1-normal ϕ_1-event e_1 is also the inception of a ϕ-event e_2 such that

$$container(e_1) = medium(e_2) \text{ and } movee(e_1) = disolvee(e_2).$$

This analysis invokes notions that have come to light in accounting for other phenomena in the analysis of word meaning. I will pass directly to these other phenomena, but will return to the problem of dispositionals briefly later, when I discuss ability.

7.2. Telicity

I am abstracting here away from all problems having to do with time and the progressive, and concentrating on the relation between a telic event and its culminating state.[15] The most important feature of the type of telic events is that these events have three associated parts: the inception, the body, and the culmination. The inception is an initiating event. The culmination is the state that normally results. (Since the beginning, the theory of planning has concentrated on features of culminations, since these represent properties of the state that can be assumed to result if the agent performs an action.) The body is the process that normally leads to the culmination; often (as in closing a door or filling a glass), the body will consist of stages in which the goal is progressively achieved.[16] We can lay it down as a general default on telic events that the culmination of such an event will occur if its inception and body occurs. In many cases (like dissolving, or filling a glass from a tap, but not like filling a glass from a pitcher) the body will also normally occur if the inception occurs. Thus, I am likening unfulfilled telic events to Manx cats—they are objects that belong to a type that normally has a certain part, but that for some *ad hoc* reason happen to lack this part.

7.3. Agency

Some formalisms of agency in AI involve a separation of events into those that are in an agents' immediate repertoire and those that are not.[17] If such a division is adopted for linguistic purposes, we can capture agency—at least, for telic events that normally follow from their inceptions—as follows.

7.v. $Do(x, e)$ holds iff the inception $e^{\mathbf{0}}$ of e is identical to an immediate action $e^{\boldsymbol{\varpi}}$ that is performed by x, provided that the body of $e^{\boldsymbol{\varpi}}$ occurs.[18]

For example, it follows from this account that in case someone puts a piece of salt in water and it then dissolves in the ordinary way, them this person has also performed the action of dissolving it in water, assuming that putting the salt in water is immediate an immediate action. Moreover, the action of putting the salt in water will be the inception of the dissolving event.

---

[15] It should be clear, though, that I have in mind an account that would relate the truth of a progressive sentence to the occurrence of the body of an event.

[16] For ideas that are in some respects similar to these, see [Steedman & Moens 87].

[17] See especially [Moore 90].

[18] I want to say that the body of a telic event occurs even if it is partial or incomplete.

On this treatment, we dispense with an explicit use of any causal notions in the analysis of agency—though causal notions are certainly implicit if we believe that there is a connection between sequences of events conforming to patterns of normality and causal sequences. Since, despite the contribution of [Shoham 88], an explicit theory of common sense causality is not likely to be easy, I prefer such eliminative accounts. However, I'm not sure if explicit causality can be eliminated in general from the theory of agency.

7.4. Causality

The notion of causality is usually left unanalyzed in linguistic treatments of lexical decomposition. But theories of nonmonotonic reasoning offer some hope of either providing an account of causality in terms of defaults governing sequences of events, or at least of providing systematic relations between such defaults and causality. For the most extended treatment, see [Shoham 88]. This work provides another, independent reason for incorporating a theory of default reasoning in an account of the compositional semantics of words.

7.5. Ability

'This water is drinkable' doesn't mean

7.2. *An attempt to drink this water will normally culminate in its being drunk.*

Rather, the meaning is

7.3. *Normally, one can drink this water.*

This, linguistic examples, as well as obvious motivations in AI that relate to the needs of planning, lead to a need for an account of practical ability. I don't think that such an account can be given without an extended background theory of practical reasoning. For that reason, the account that I'll sketch here may seem circular or trivial. The reason (I hope) is that the background hasn't been filled in.

Let's suppose that there is a propositional constant *practical_abnormality* that is used in practical reasoning to reject alternatives because of utility considerations. That is, if a contemplated practical alternative is shown to lead (perhaps with the aid of defaults) to this constant, the alternative has thereby been shown to one that can be ruled out of consideration. A qualitative account of practical reasoning would have to relate this constant to desires and contingent circumstances.

The definition of practical ability would then be the following, where $\Box$ represents temporal necessity.

7.vi. $can(\phi) \leftrightarrow \neg\Box[\phi \rightarrow practical\_abnormality]$

7.6. Artifacts

Many artifacts are defined in terms of their normal uses. This suggests decompositional analyses such as the following example.

7.vii. A *fastener* is an object x such that, where ϕ is the event type of using x, every ϕ-normal occurrence of an event e of type ϕ is such that $purpose_e$ is to fasten an object to another object.

8. The problem of reasoning

Many researchers in the area of knowledge representation have advocated a close association between the development of systems of representation and reasoning systems. For many practitioners, this may mean only that attention must be paid to the complexity of some of the algorithms associated with the representations; for others, it may mean that the representations should be deployed in a working system whose performance has been tested in practical terms.[19] On the other hand, John McCarthy, one of the founders of AI and of the field of knowledge representation, seems to have steadfastly maintained a platonic research agenda, according to which we first clarify in logical terms the knowledge that we are trying to build into a reasoning system, without distracting ourselves with heuristics and implementation details.

Since it seems to me that we may not yet have adequate theories in which to describe our goals, and that attention to the design of successful systems is a likely source of good theories, and a good way of ensuring that they will ultimately have some reasonable relation to technology, I am a moderate on this issue.

However, I would not have been able to write this paper without pretending to be a platonist. Natural language semantics is such a large domain that it is very hard to think about it in the general way that linguistics seems to require while maintaining a good relation to some feasible reasoning task. The task of axiomatizing the theories that I have been contemplating in this theory is forbidding, and, if the axiomatization could be carried out, it is hardly likely that it would provide a promising path to any workable implementation.

Perhaps the best hope of closing the gap between theory and technology would be the development of compilers for higher-level planning languages that use the kinds of concepts that we have found useful for lexical decomposition. Some computer scientists are beginning to work in this direction, and compilers have even been written for simple languages of this sort.[20] But this line of research will need a lot of development before we can hope to get much help from it in figuring out how to assign procedural significance to parts of lexical semantics, and at best it is pretty slender at the moment.

[19] See [Brachman 90].

[20] See [Shoham forthcoming].

References

[Asher & Morreau 91] Nicholas Asher and Michael Morreau. "Commonsense entailment: a modal theory of nonmonotonic reasoning." In J. Mylopoulos and R. Reiter, eds., *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence,* Morgan Kaufmann, Los Altos, California, 1991, pp. 387–392.

[Bach 89] Emmon Bach. *Informal Lectures on Formal Semantics.* State University of New York Press, Albany, NY, 1989.

[Boutilier 92] Craig Boutilier. *Conditional logics for default reasoning and belief revision.* Technical Report KRR-TR-92-1, Computer Science Department, University of Toronto, Toronto, Ontario, 1992.

[Brachman 90] Ronald Brachman. "The future of knowledge representation." In *AAAI-90: Proceedings of the eighth national conference on artificial intelligence.* AAAI Press and MIT Press, Menlo Park, CA and Cambridge, MA, respectively, pages 1082–1092, 1990.

[Carnap 28] Rudolph Carnap. *Der logische Aufbau der Welt.* Weltkreis-Verlag, Berlin-Schlactensee, 1928.

[Carnap 36-37] Rudolph Carnap. "Testability and meaning." *Philosophy of Science 3,* pages 419–471, 1936, and *Philosophy of Science 4,* pages 1–40, 1937.

[Carnap 56] Rudolph Carnap. *Meaning and necessity,* 2nd edition. Chicago University Press, Chicago, 1956.

[Cassirer 55] Ernst Cassirer. *The Philosophy of Symbolic Forms,* volumes 1–3. Yale University Press, New Haven, CT, 1955. [These works were first published (in German) in 1923–29.]

[Church 51] Alonzo Church. The Need for Abstract Entities in Semantic Analysis. *Proceedings of the American Academy of Arts and Sciences 80,* pages 100–112, 1951.

[Davis 90] Ernest Davis. *Representations of commonsense knowledge.* Morgan Kaufmann, Los Altos, California, 1990.

[Dowty 79] David Dowty. *Word meaning and montague grammar.* D. Reidel, Dordrecht, Holland, 1979.

[Ginsberg 90] M. Ginsberg, ed. *Nonmonotonic reasoning.* Morgan Kaufmann, Los Altos, 1988.

[Ginsberg 88b] "Introduction." In [Ginsberg 88b], pp. 1–23.

[Goodman 63] Nelson Goodman. "The significance of *Der logische Aufbau der Welt.* In P. Schilpp, ed., *The philosophy of Rudolph Carnap,* Open Court, LaSalle, Illinois, pp. 545–558, 1963.

[Hobbs & Moore 90] Jerry Hobbs and Robert Moore. *Formal theories of the commonsense world.* Ablex, Norwood, New Jersey, 1988.

[Horty & Thomason forthcoming] John Horty and Richmond Thomason. "Conditionals and artificial intelligence." *Fundamenta informaticae 15* (1991), pp. 301–324.

[Jespersen 65] Otto Jespersen. *The philosophy of grammar.* W.W. Norton, New York, 1965. [This work was originally published in 1924.]

[Lewis 73] David Lewis. *Counterfactuals.* Harvard University Press, Cambridge MA, 1973.

[Lifschitz 89] Vladimir Lifschitz. "Circumscriptive theories." In R. Thomason, ed., *Philosophical logic and artificial intelligence,* Reidel, 1989, pp. 109–159.

[McCarthy80] John McCarthy. "Circumscription—A form of non-monotonic reasoning." *Artificial Intelligence 13,* 1980.

[McCarthy 84] John McCarthy. "Some expert systems need common sense." In H. Pagels, ed., *Computer culture: the scientific, intellectual and social impact of the computer.* Volume 426, Annals of the New York Academy of Sciences, pages 129–137, 1984.

[McCarthy 86] John McCarthy. "Applications of circumscription to formalizing common sense." *Artificial Intelligence,* 1986.

[McCarthy 89] John McCarthy. "Artificial intelligence and logic." In R. Thomason, ed., *Philosophical logic and artificial intelligence,* Kluwer Publishing Co., Dordrecht, Holland, pages 161–190, 1989.

[Montague 62] Richard Montague. "Deterministic theories." In *Decisions, Values, and Groups,* Volume 2. Pergamon Press, Oxford, pages 325–370, 1962. Reprinted in [Montague 74], pages 303–359.

[Montague 69] Richard Montague. "On the nature of certain philosophical entities. " *The Monist 53,* pages 159–194, 1969. Reprinted in [Montague 74], pages 148–187.

[Montague 73] Richard Montague. "The Proper treatment of quantification in ordinary English." In J. Hintikka, J. Moravcsik, and P. Suppes (eds.), *Approaches to natural language: proceedings of the 1970 Stanford workshop on grammar and semantics.* D. Reidel, Dordrecht, Holland, pages 221–242, 1973. Reprinted in [Montague 74], pages 247–270.

[Montague 74] Richard Montague. *Formal philosophy: selected papers of Richard Montague.* Edited by R. Thomason. Yale University Press, New Haven, CT, 1974.

[Moore 90] Robert Moore. "A formal theory of knowledge and action." In [Hobbs & Moore 90], pp. 319–358.

[Shoham 88] Yoav Shoham. *Reasoning about change.* MIT Press, Cambridge, Massachusetts, 1988.

[Shoham forthcoming] Yoav Shoham. "Agent oriented programming." *Artificial intelligence,* forthcoming.

[Stalnaker & Thomason 70] Robert Stalnaker and Richmond Thomason. "A semantic analysis of conditional logic." *Theoria 36,* pages 23–42, 1970.

[Steedman & Moens 87] Mark Steedman and Marc Moens. "Temporal ontology in natural language." In *Proceedings of the 25th annual conference of the Asscoiation for Computational Linguistics.* Association for Computational Linguistics, 1987, pp. 1–7.

[Thomason 90] Richmond Thomason. "Propagating epistemic coordination through mutual defaults I." In Rohit Parikh, ed., *Proceedings of the third conference on theoretical aspects of reasoning about knowledge.* Morgan Kaufmann, San Mateo, CA, pages 29–39, 1990.

[Thomason 91] Richmond Thomason. "Logicism, artificial intelligence, and common sense: John McCarthy's program in philosophical perspective." In *Artificial intelligence and mathematical theory of computation,* Vladimir Lifschitz, ed., Academic Press, San Diego, 1991, pp. 449–466.

[Thomas *et al.* 90] Becky Thomas, Yoav Shoham, and Anton Schwartz. "Modalities in agent-oriented programming." Unpublished manuscript, Computer Science Department, Stanford University, Stanford, CA, 1990.

THE FRAMEWORK OF UPDATE SEMANTICS

FRANK VELTMAN

UNIVERSITY OF AMSTERDAM

1. The notion of an update system.

The standard definition of logical validity runs as follows: An argument is valid if its premises cannot all be true without its conclusion being true as well. By far the most logical theories developed so far have taken this definition of validity as their starting point. Consequently, the heart of such theories consists in a specification of truth conditions.

The heart of the theories developed within the framework I want to discuss does not consist of truth conditions but of update conditions. Within the framework of update semantics the slogan 'You know the meaning of a sentence if you know the conditions under which it is true' is replaced by this one: 'You know the meaning of a sentence if you know the change it brings about in the information state of anyone who accepts the news conveyed by it'. Thus, meaning becomes a dynamic notion: the meaning $[\phi]$ of a sentence ϕ is an operation on information states.

Let σ be an information state and ϕ a sentence with meaning $[\phi]$. We write $\sigma\ [\phi]$ for the information state that results when σ is updated with ϕ. In most cases $\sigma\ [\phi]$ will be different from σ, but every now and then it may happen that $\sigma\ [\phi] = \sigma$. Then the information conveyed by ϕ is already subsumed by σ: if σ is updated with ϕ, the resulting information state turns out to be σ again. In such a case, i.e. when $\sigma\ [\phi] = \sigma$, we write σ ª ϕ and say that ϕ *is accepted in* σ. To specify an update semantics for a given language L, one has to specify a set Σ of relevant (information) states, and a function [ ] that assigns some state $\sigma\ [\phi]$ to each sentence ϕ of L and each state σ in Σ. Any such triple $< L, \Sigma, [\ ] >$ will be called an update system.

DEFINITION

(i) Let $< L, \Sigma, [\ ] >$ be an update system, and let $\sigma, \tau \sum \Sigma$. Then $\sigma \leq \tau$ iff $\sigma = \tau$ or there are $\phi_1, ..., \phi_n$ in L such that $\tau = \sigma\ [\phi_1]...[\phi_n]$. Whenever $\sigma \leq \tau$, we say that τ is *accessible from* σ.

(ii) An update system $< L, \Sigma, [\ ] >$ is *expressively complete* iff there exists a state $\mathbf{0} \sum \Sigma$, the *minimal* state, such that $\mathbf{0} \leq \sigma$ for every state σ.

Below, we will restrict our attention to systems that are expressively complete in the sense of (ii). (Assuming that there is such a thing as a minimal information state this seems a sensible thing to do. We are not interested in information states *per se,* we are interested in the impact that language has on them. There is no need for an update system to tell us more about information states than is required in order to understand the processing of the language for which the system is developed.)

2. Constraints

There are several global constraints one might want to put on update systems. For one thing, one might wonder whether the operation ϕ will always be total. In general the answer is no. It is not difficult to think of a situation in which for a given ϕ and a given information state σ, σ [ϕ] will be undefined. Take the case of a pronoun desperately looking for a referent:

'He is just joking'

If it is not clear to whom the speaker is refering, the addressee will not know what to do with this statement. Or take the case of presupposition. As is shown in Beaver(1991), the framework of update semantics offers a natural explanation of the notion of this notion:

ϕ *presupposes* ψ iff for every state σ, σ [ϕ] is defined only if σ ª ψ.

Another constraint that does not generally hold, is this:

Idempotence: For any state σ and sentence ϕ, σ [ϕ] ª ϕ

At first sight this principle goes without saying — what would 'updating your information with ϕ' mean if not at least 'changing your information in such a manner that you come to accept ϕ'? Still, there are sentences of natural language for which the Principle of Idempotence is questionable. Here paradoxical sentences like 'This sentence is false' are a point in case. Asis shown in Groeneveld[to appear] shows there is no such thing as a successful update with the Liar, and that is where its paradoxality resides.

A third principle worth looking at is the Principle of Permutation:

Permutation: σ [ϕ] [ψ] = σ [ψ] [ϕ]

Here, too, it is not difficult to find counterexamples. The clearest ones are provided by sentences in which modal qualifications like 'presumably', 'probably', 'must', 'may' or 'might' occur. Consider:

— Somebody is knocking at the door... Presumably, it's John... It's Mary.

— Somebody is knocking at the door... It's Mary ... Presumably, it's John.

These two sequences consist of the same sentences. Only the order differs. Nevertheless processing the first sequence does not cause any problems, but processing the second sequence does. Explanation: it is quite normal for one's expectations to be overruled by the facts —that is what is going on in the first sequence. But if you already know something, it is a bit silly to pretend that you still expect something else, which is what is going on in the second.

One of the advantages of the dynamic approach is that these differences can be accounted for. The set-up enables us to deal with sequences of sentences, whole texts. Let ϕ_1 = 'Somebody is knocking at the door', ϕ_2 = 'Presumably, it's John', and ϕ_3 = 'It's Mary'. If we want, we can compare σ [ϕ_1] [ϕ_2] [ϕ_3] with σ [ϕ_1] [ϕ_3] [ϕ_2] for any information state σ, and see if there are any differences.

The phrase update semantics is a bit misleading. 'Updating your information state with ϕ' suggests that what you have to do is to just add the informational content of ϕ to the information you already have. But, as the next proposition shows, this picture only holds as long we are dealing with a system in which [ ] is idempotent and permutation invariant.

DEFINITION

An expressively complete update system $< L, \Sigma, [\,] >$ is *additive* iff there exist a binary operation + on Σ, and a function " ' which assigns to every sentence ϕ of L a state "ϕ' in Σ such that the following holds:

(i) the operation + has all the properties of a 'join' operation:

$$\mathbf{0} + \sigma = \sigma$$

$$\sigma + \sigma = \sigma$$

$$\sigma + \tau = \tau + \sigma$$

$$(\rho + \sigma) + \tau = \rho + (\sigma + \tau)$$

(ii) for every sentence ϕ and state σ, $\sigma\,[\phi] = \sigma +$ "ϕ'

PROPOSITION

An expressively complete system $< L, \Sigma, [\,] >$ is additive iff the principles of Idempotence and Permutation hold.

As soon as we are dealing with an additive system, the dynamic approach has little to offer over and above the static approach. In such a case one can just as well associate with every sentence ϕ of L a basic, static meaning — $\mathbf{0}\,[\phi]$, standing for the information contained in ϕ— and define the dynamic notion in terms of it. Only if one interested in describing phenomena for which Idempotence or Permutation do not hold, one has to go dynamic.

3. Notions of validity

Various explications of logical validity suggest themselves:

- An argument is valid$_1$ iff updating the minimal information state '$\mathbf{0}$' with the premises $\psi_1, \ldots, \psi_n$ in that order, yields an information state σ in which the conclusion ϕ is accepted. Formally:

$$\psi_1, \ldots, \psi_n \models_1 \phi \text{ iff } \mathbf{0}\,[\psi_1]\ldots[\psi_n] \models \phi.$$

- An argument is valid$_2$ iff updating any information state σ with the premises $\psi_1,\ldots, \psi_n$ in that order, yields an information state in which the conclusion ϕ is accepted. Formally:

$$\psi_1, \ldots, \psi_n \models_2 \phi \text{ iff for every } \sigma, \sigma\,[\psi_1]\ldots[\psi_n] \models \phi.$$

- An argument is valid3 if one cannot accept all its premises without having to accept the conclusion as well. More formally:

$$\psi_1, \ldots, \psi_n \models_3 \phi \text{ iff } \sigma \models \phi \text{ for every } \sigma \text{ such that } \sigma \models \psi_1, \ldots, \sigma \models \psi_n$$

PROPOSITION

In any expressively complete update system the next two statements are equivalent.

(i) The principles of Idempotence and Permutation hold.

(ii) $\psi_1, \ldots, \psi_n \models_1 \phi$ iff $\psi_1, \ldots, \psi_n \models_2 \phi$ iff $\psi_1, \ldots, \psi_n \models_3 \phi$.

In the absence of *Permutation* the three explications of validity will give rise to different logics. And all three have their use as is argued in Veltman[to appear}. The third notion of validity is monotonic. If an argument with premises $\psi_1,\ldots,\psi_n$ and conclusion ϕ is valid3, then it remains valid3 if you add more premises to $\psi_1,\ldots,\psi_n$. The second notion will be non-monotonic. Note however that the second notion is at least left-monotonic:

$$\text{If } \psi_1, \ldots, \psi_n \models_2 \phi, \text{ then } \chi, \psi_1, \ldots, \psi_n \models_2 \phi.$$

What fails is right-monotonicity:

$$\text{If } \psi_1, \ldots, \psi_n \models_2 \phi, \text{ then } \psi_1, \ldots, \psi_n, \chi \models_2 \phi.$$

The first notion is neither right nor left-monotic. But it it is easy to verify that it conforms to the following principle of *Cautious Monotonicity*:

$$\text{If } \psi_1, \ldots, \psi_n \models_1 \phi \text{ and } \psi_1, \ldots, \psi_n, \theta_1, \ldots, \theta_k \models_1 \chi, \text{ then}$$
$$\psi_1, \ldots, \psi_n, \phi, \theta_1, \ldots, \theta_k \models_1 \chi$$

Moreover, it complies with the following version of the principle of *Cut Elimination*, which we shall call *Cautious Cut*:

$$\psi_1, \ldots, \psi_n \models_1 \phi \text{ and } \psi_1, \ldots, \psi_n, \phi, \theta_1, \ldots, \theta_k \models_1 \chi, \text{ then}$$
$$\psi_1, \ldots, \psi_n, \theta_1, \ldots, \theta_k \models_1 \chi$$

Given the principle of idempotence, we also find that this notion of validity is *Reflexive*.

$$\psi_1, \ldots, \psi_n, \phi \models_1 \phi$$

PROPOSITION

(i) Let º be any consequence relation for which the three principles mentioned above hold. Then there is an expressively complete update system in which the principle of Idempotence holds such that

$$\psi_1, \ldots, \psi_n \text{ º } \phi \text{ iff } \psi_1, \ldots, \psi_n \models_1 \phi$$

(ii) Let º be any consequence relation for which in addition to the three principles mentioned above is *left-monotonic*. Then there is an expressively complete update system in which the principle of Idempotence holds such that

$$\psi_1, \ldots, \psi_n \text{ º } \phi \text{ iff } \psi_1, \ldots, \psi_n \models_2 \phi$$

In other words, the principles mentioned completely characterize the dynamic notions of validity.

References

Groeneveld, W.: 1989, 'Dynamic Semantics and Circular Propositions', To appear in *The Journal of Philosophical Logic*.

Beaver, D: 1991, 'The kinematics of presupposition', Proceedings of the 8th A'dam Colloquium.

Veltman, F: 'Defaults in Update semantics', *Journal of Semantics* **9** (), To appear in *The Journal of Philosophical Logic*.

Zeevat, H: 1992, 'Presupposition and Accomodation in Update semantics', *Journal of Semantics* **9**, pp 379-412.

Update Semantics and Discourse Grammar
Abstract

Henk Zeevat
Computational Linguistics
University of Amsterdam

The starting point of this talk is the version of Discourse Grammar developed by Pruest and Scha in recent years from the older proposals of Scha and Polanyi and the question what the relationship is between this kind of grammar and the developing account of pragmatics by means of update semantics and other dynamical approaches to interpretation.

I will set out by presenting a brief introduction to both topics and will continue by an attempt to integrate these two theories.

1 Discourse Grammar and the MSCD

In the theory of Pruest, VP-ellipsis and important subcases of anaphora are treated by means of the construction of the MSCD. If we have two clauses (or longer constructions) the MSCD is the maximal structure that combines all the information that is available in one of two structures where the information is compatible and takes the generalisation where they are not compatible. The following is a definition for prolog terms.

1. MSCD(A,B)=C if C is the unification of A and B

2. MSCD(r(A1,...,Ak),r(B1,..,Bk)= r(MSCD(A1,B1),...,MSCD(An,Bn) otherwise

3. MSCD(A,B) is an anonymous variable if neither 1 nor 2 apply.

Under the natural assumption of assigning a variable —but possibly typed— meaning to pronouns and omitted constituents, this notion allows us to treat simple cases of anaphora and ellipsis, by unifying the second of two clauses with the MSCD obtained from from the two clauses:

John ran	p(run,j)
Bill did too	p(X,b)
MSCD	p(run,Y)
result	p(run,b)

John kissed Mary	p(kiss,j,m)
Bill embraced her.	p(embrace,b,X)
MSCD	p(R,V,m)
result	p(embrace,b,m)

The functor *p* represents here various predication predicates.

Pruest generalises this too simple treatment to more complicated cases by developing a theory of structured semantic representations. Moreover, the knowledge which makes clauses submit to this process is guided by discourse grammar rules. Therefore, successful resolution or MSCDing in turn guides the parsing and interpretation process for discourses.

Nevertheless, the process fails to deal with important cases of anaphora as it stands. Notably, anaphora within the same clause cannot be treated and the theory gives up on anaphorical patterns where parallelism does not directly apply, such as contrast and elaboration. Thus the theory only applies to list structures, although in practice for many cases of elaboration and contrast the right predictions are obtained. Another feature that is not adequately dealt with are anaphorical processes where parallelism does apply, but where the reference of pronoun and antecedent are not the same. Such cases are found in temporal reference and by certain uses of definites.

Next to the considerable empirical coverage of VP-ellipsis achieved by Pruest an additional advantage is the explanation of parallellism in pronoun resolution. It is the only (partial) account of pronoun resolution that makes parallelism a starting point rather than a heuristic technique. Nevertheless, the computation of MSCDs stands itself in need of further explanation. A first explanation can be sought in perceptual psychology and particularly in the observed impossibility of not remarking similarities in two perceptual inputs that are offered in one after the other. Though important there is another explanation strategy here — not necessarily inconsistent with the psychological one— where MSCDs are explained in terms of information strategy. This view interprets the MSCD as the topic of the two sentences.

2 Topics

I will follow *Van Kuppevelt 1991* in the assumption that all topics are questions. Sometimes in the course of a dialogue the topic is explicitly set by a question, more frequently the first assertion on a topic both sets an implicit question and answers it by new information. (This can be either the theme of the utterance or its focus. In the following I will equivocate over themes and foci). This view has slightly more empirical bite as topics may derive from other sources than a process of comparing two sentences. It are these extra constraints that will be exploited below to give an improved, but still partial— account of anaphora. If we can think of an MSCD as being given externally and not as fully determined by formal relationships between clauses we also lose a correct feature of Pruests theory. It does no longer follow that the topic is the maximal common denominator. This is a problem which we can only deal with by stipulation in the current context. It would appear that discourse is like language in general and that we always assume the strongest interpretation. Intonational facts may of course help us out.

The two extra constraints that I am interested in are the following. In the first place, a chunk of discourse may be the reply to a question. In that case the topic can identical to the question (in a suitable format) and the question may fill in gaps and pronoun antecedents. As is well known, questions are not always literally replied to. The answer may go beyond what was actually asked by formally being the reply to a stronger question. Also a question may be underanswered. If we would apply the method of MSCDs here, we get in the first case a topic that is partially answered by an answer to the literal question. In the second case, we get a topic that is a subquestion of the question that was asked. So in this case we get a question that has a formal relationship to the question that was put. This relationship is easily describable in terms of a unification like ordering over questions.

The second constraint that seems important has to do with elaborations. Elaborations typically break parallellism in the resolution of pronouns but typically in a minimal way: as much remains parallel as can remain so. In fact, Pruest's thesis contains a number of cases where MSCDs do fine in dealing with anaphora over elaborations. The point here is that the topic of elaborations is closely related to the content of the element that is elaborated. Elaborations either specify participants or adjuncts of that element or they further specify the kernel event of the utterance. The following table offers some possible questions after an assertion that could be the topic of an elaboration.

Bill hit Suzy	
Why did Bill hit Suzy	He got angry/She called him a liar.
Who is Bill	He is Suzy's former boyfriend.
Who is Suzy	She is my neighbour's daughter.
With what?	With a newspaper.
When?	At 4/ When she called him a liar
Where?	In the kitchen./ On the head.
How did it happen?	She came into the room. He had a newspaper and hit her on head.

Specific answering strategies are sometimes responsible for more specific questions. Again we are in a position to say that the topic of an elaboration must bear a relationship to the element that it elaborates. The process of proving that it has will attach a discourse relation as well as antecedents to the elaboration.

So far we have seen relationships where there is formal content. The closest is between a question and its direct answer. A weaker relationship is found with topic specialisation and generalisation. Still weaker is the formal relationship in elaborations. No formal content at all remains when we are dealing with contrastive relationship. The reason for this is obvious. In a concessive statement we concede an element of which the truth is expected to be contradicted by the truth of the assertion we are currently making. What this statement is depends not on the form of the assertion but on the content of the system of beliefs and expectations. Formal relationships may be present or absent depending on the concession made and the reasoning that establishes a link between the conceded element and the statement that prompts its concession.

3 Updating with questions.

An assertion is processed by updating its contribution to the information state that is updated by means of the question. The assertion update is then not so much the update with its logical content as the update with the unification of the question and the content of the answer. (A refinement allows for undoing the question update in the case of negative answers like *No* to a *WH*-question or *I do not know* to an arbitrary question.) The unification to which the answers are submitted resolves pronouns and VP-ellipsis and is responsible for exhaustification effects such as Evans' E-type readings and scalar implicatures.

Within the framework of a discourse grammar, this complex update procedure involves a number of constraints.

a. The topic must be unifiable with the discourse element which the discourse grammar tries to integrate into the parse.

b. For elaborations the elaboration topic must be a question that arises from the element from which the elaboration starts.

c. Before any element, its topic must be a proper question on the information state. This means that the information state allows two alternative answers and that the information state satsifies the question presupposition if the question has one.

d. After the element, it must have ceased to be a proper question. This means that the question is now answered completely.

For the purposes at hand, WH-questions will be formalised as open formulae to which a question operator is attached. A question update then does supply or rather construct some information. It supplies a pointer to the answer. As this is a new pointer, the information state will be restricted to those values of the pointer that are true and exhaustive with respect to the index at hand. The truth derives from the standard interpretation of variables, exhaustivity is due to the question operator.

Properly answering a question is asserting the identity between the question variable (or the question as such) and its answer.

4 The Question Ordering

Having open formulas as the interpretation of questions entails having a partial order over questions, which is in essence the unification order.

Who does what?
Who does sleep?
Which boy sleeps?
Does Bill sleep?

Logically, the order is a specialisation of the relation between questions: whenever A answers x, A answers y.

The order can be exploited for redefining answers to a question x: answering x equals answering each of the questions x_i. What we want is some constraint $x = sum(X)$, with x a question and X a set of of questions. For this to be true, it must hold that every $y \in X$ is a subquestion of x (a formal constraint) and that the reference of the WH-variables in x, obtained after the update, is the sum of the references of the corresponding elements in the subquestions with respect to the information state at hand.

What did Bill do last Friday.

He went out.
He had a few beers.
He saw a film.
He came home at midnight.

Who sleeps

John sleeps.
Bill sleeps.
Harry sleeps.

5 A Grammar

I am assuming a syntax for basic clauses, which analyses them as feature structures with slots info-in, info-out, syntax and phonology. A clause is enriched to be a dcu by having slots for topic (value as for clause). In the syntax we incorporate indexes into elements of the evolving information state.

There are constraints here that the topic subsumes the clause slot after the clause slot is unified with a copy of the topic, that the topic holds exhaustively over info-in, and that info-in does not equal info-out. (This latter constraint can be stated alternatively as the requirement that the question is not answered already by info-in).

The discourse grammar rules are now there to make larger dcus out of the basic ones. We have two constructions.

A. A list construction that adds to the semantics of each of the elements the statement that the union of the topics equals the topic.

B. An elaboration construction that sets up a topic in a way constrained by the topic of the element on which elaboration takes place.

```
LIST:phon = [DCU:phon,CONT:phon]
DCU:info-in = LIST:info-in[LIST:topic][LIST:topic=DCU:topic+CONT:topic]
DCU:topic := LIST:topic
CONT:topic = LIST:topic
DCU:info-out = CONT:info-in
LIST:info-out = CONT:info-out
question(LIST:topic,LIST:info-in)
answered(LIST:topic,LIST:info-out)

CONT = REST or VOID

REST:phon= [DCU:phon, CONT:phon]
DCU:info-in = REST:info-in
DCU:topic := REST:topic
CONT:topic = LIST:topic
DCU:info-out = CONT:info-in
LIST:info-out = CONT:info-out

ELAB:phon = [DCU:phon,LIST:phon]
ELAB:topic = DCU:topic
question-about(LIST:topic,DCU:topic)

VOID:phon = []
VOID:info-in=VOID:info-out

ELEM:syntax:= ELEM:topic
ELEM:syntax:info-in = ELEM:info-in[ELEM:topic]
question(ELEM:topic,ELEM:info-in),
answered(ELEM:topic,ELEM:info-out)
```

Next to normal unification, we require one-way unification :=.

Constraints

question(QUESTION,INFOSTATE)

is the negation of

answered(QUESTION,INFOSTATE)

A question is a syntactic-semantic representation of a question. A question is answered on an information state intuitively expresses that the information state already has an answer to the question. It depends on the nature of the information state (database, set of possible worlds, partial model) how this relation is to be defined. In standard update semantics, the following definition is a possible one: the question update is proper and limits the information state to elements which are determinate for the value of the WH-variables.

answered(QUESTION,INFOSTATE)

The other constraint

question-about(ASSERTION,QUESTION)

is of an entirely different nature and it is not clear that semantics is involved. It concerns the focus of the ASSERTION. The elements there are new and involve new referents. Concerning each of the referents (including the event marker) questions of identity can be put which in the case of eventmarkers can be questions with respect to cause and effect.

6 Appendix: the question update

The logical form of a question will be here that of a DRS within a question operator $q(\varphi)$, with discourse markers associated to φ. Updates proceed in a number of steps. q builds a copy of the current information state and computes the exhaustive reading for the discourse markers of φ. On the copy φ is true and the reference of the discourse markers are maximal. A refusal to answer deletes the copy, a denial subtracts the closure of the updated copy from the original and a positive reply intersects the original with the updated copy after a further update with the answer.

The basic information carriers are possible worlds. For these purposes it is practical to see variables as constants (with just a formal distinction). Ontological operations (set formation or mereology) are characterised by a set of axioms which each world meets. Also we assume that the domains of the objects are taken from a common set.

Two worlds can be variants of each other with respect to a set of variables: all the values they assign are the same except for the values they assign to the variables in the set. Similarly worlds can be object-identical by assigning precisely the same values to all constants and variables (they can still differ in what is the case in these worlds). Finally worlds can be object-identical but for the value of a set of variables.

In terms of these relations and normal updates, it is possible to define the notion of a question update and the relation of a question being unanswered over a given information state.

A world w is exhaustive within a set of worlds W with respect to a set of variables X iff $\forall w^{0} \in W \forall w^{00} \in W(w =_X w^{0} \wedge objectidentical(w, w^{00})) \rightarrow (\exists w^{000} \in W objectidentical(w^{000}, w^{0}) \wedge w^{000} =_X w^{00}))$

A question update for a question $q(\varphi)$ with discourse markers DM is then the set of exhaustive worlds in $\sigma[\varphi]$ with respect to DM.

References:

Hub Pruest. **On Discourse Structuring, VP Anaphora and Gapping.** PhD Dissertation, University of Amsterdam, 1992.

Jan van Kuppevelt. Topic en Comment. Impliciete en expliciete vraagstelling in discourse. PhD Dissertation. University of Nijmegen 1991.

Session Summary

Manfred Krifka
Christopher Habel

Friday, 26-02-93

The common theme of the morning talks was non-monotonic reasoning and semantic formalisms. Various versions of non-monotonic reasoning have been developed over the last decade to facilitate knowledge representation tasks, but it also has became obvious that such techniques will be crucial for the semantic representation of a wide range of NL constructions (generic sentences, conditionals, and aspectual information, to name just a few). Beyond the sentence level, non-monotonic reasoning is needed in discourse semantics (especially to establish coherence relations).

Rich Thomason (Pittsburgh) gave a talk about "Non-monotonic formalisms for NL semantics", using his network representation containing "defeasible arrows". He discussed various applications, among others lexical semantics and in particular thematic role assignment (e.g., a transitive verb will have an agent/patient structure by default). Then he sketched the development of non-monotonic logics as an attempt to specify the underlying logic in informal reasoning systems as developed within AI. He stressed that it will be an important task to uncover the "NL metaphysics", and that non-monotonic rules will be essential to describe it (for example, that telic events typically, but not always have a culmination point). In the discussion, Bob Moore suggested to treat the well-known problem of the proper fine-grainedness of propositions in belief contexts by non-monotonic inferencing.

Espen Vestre and Bernd Nebel (Saarbruecken) talked about "Belief revision methods in NL semantics". Vestre discussed the relation between belief revision and the semantics of conditionals, following Gaerdenfors, showing that the simple possible world representation of beliefs runs into problems with examples like Hansson's "Man with the hamburger", and pointed out that the crucial problem is updating a knowledge state with a conditional. Nebel proposed to represent beliefs by finite sets of sentences (socalled bases), not possible worlds or deductively closed sets, and proposed a way to revise such bases with simple sentences and conditionals. The discussion centered on the distinction of indicative and counterfactual conditionals and on how to make sense of the intuitive notion of various degrees of epistemic entrenchment of belief contents.

Jeff Pelletier (Alberta), in his talk "Psychologically oriented studies in belief revision", argued that the study of non-monotonic reasoning differs from traditional branches of logic, as it is motivated by the actual reasoning of people. Hence it should be regarded as a psychological enterprise. He presented a series of experiments designed to test whether

Lifschitz's "benchmark tests" for nonmononotic formalisms actually predict the behavior of persons, reported on a number of differences, and traced these effects back to possible factors. For example, it seemed to matter whether exceptions have been explicitly mentioned and whether they show an essential similarity to the object about which a conclusion should be drawn. In the discussion, David Dowty questioned the design of the experiments, as it did not take into account possible effects arising through conversational implicatures (e.g., mentioning a related exceptional case implicates that the case in question is also an exception). Frank Veltman contended that although psychological research may be very valuable, nonmonotonic reasing should still be considered a branch of pure logic, as people can convince themselves that certain inference patterns are problematic.

Dagstuhl-Seminar 9308: List of Participants

Dipl.-Inform. Jürgen **Allgayer**
IBM Germany
Software Architectures & Technologies
Dept. 2300 Bldg. 7030-91
Hanns-Klemm Str. 45
W-7030 BÖblingen
allgayer@vnet.ibm.com
tel.: +49-7031-16 6191
fax.: +49-7031-16-6440

Dr. Douglas E. **Appelt**
SRI International
333 Ravenswood Ave.
Menlo Park CA 94025
USA
appelt@ai.sri.com
tel.: +1-415-859-61 50
fax.: +1-415-859-37 35

Martin H. **van den Berg**
University of Amsterdam
Computational Linguistics
Spuistraat 134
NL-1012 VB Amsterdam
The Netherlands
vdberg@alf.let.uva.nl

Alan **Black**
University of Edinburgh
Centre for Cognitive Science
2 Bucclench Place
Edinburgh EH8 9LW
Great Britain
awb@ed.ac.uk
tel.: +44-31-650-4627
fax.: +44-31-650 4587

Dr. Robin **Cooper**
University of Edinburgh
Center for Cognitive Science
2 Buccleuch Place
Edinburgh EH8 9LW
Great Britain
cooper@cogsci.ed.ac.uk
tel.: +44-31-650-44 07
fax.: +44-31-650-45 87

Prof. David **Dowty**
Ohio State University
Department of Linguisties
Oxley Hall 222
1712 Neil Avenue
Columbus OH 43 210
USA
dowty@ling.ohio-state.edu
tel.: 614-292-5400

Prof. Dr. Jens Erik **Fenstad**
Universitetet i Oslo
Department of Mathematics
Postboks 1053 Blindern
N-0316 Oslo
Norway
jfenstad@math.uio.no
tel.: +47-22-85 58 65
fax.: +47-22-85 43 49

Prof. Dr. Nissim **Francez**
Israel Institute of Technology
Computer Science Departement
Technion City
Haifa 32000
Israel
francez@cs.technion.ac.il
tel.: +972-4-294369 / +972-4-928444
fax.: +972-4-294353

Prof. Dr. Christopher **Habel**
Universität Hamburg
FB Informatik - WSV
Bodenstedtstr. 16
W-2000 Hamburg 50
habel@informatik.uni-hamburg.de
tel.: +49-40-4123 6107
fax.: +49-40-4123 6159

Prof. Dr. Erhard **Hinrichs**
Universität Tübingen
Seminar für Sprachwissenschaft
Kleine Wilhelmstr. 113
W-7400 Tübingen
hinrichs@mailserv.zdv.uni-tuebingen.d
tel.: +49-7071-29 54 46

Dr. Chung Hee **Hwang**
University of Rochester
Department of Computer-Science
Rochester NY 14627-0226
USA
hwang@cs.rochester.edu
tel.: +1-716-275-5766
fax.: +1-716-461-2018

Prof. Dr. Hans **Kamp**
Universität Stuttgart
IMSV-Institut für masch.
Sprachverarbeitung
Azenbergstr. 12
W-7000 Stuttgart
kamp@ims.uni-stuttgart.de
tel.: +49-711-121-1362
fax.: +49-711-121-1366

Dr. Walter **Kasper**
DFKI Saarbrücken
Stuhlsatzenhausweg 3
W-6600 Saarbrücken 11
Germany
kasper@dfki.uni-sb.de
tel.: +49-681-302 5284
fax.: +49-681-302 5341

Dr. Ewan **Klein**
University of Edinburgh
Centre for Cognitive Science
2 Buccleuch Place
Edinburgh EH8 9LW
Great Britain
klein@ed.ac.uk oder ewan@cogsci.ed.ac.nk
tel.: +44-31-650-44 28 oder +44 31 650 4594
fax.: +44-31-650-45 87

Dr. Manfred **Krifka**
University of Texas at Austin
Dept. of Linguistics
Calhoun Hall
Austin TX 78712 - 1196
USA
ligk417@utxvms.cc.utexas.edu
tel.: +1-512-471 1701

Noor **van Leusen**
University of Amsterdam
Computational Linguistics
Spuistraat 134
NL-1012 VB Amsterdam
The Netherlands
noor@alf.let.uva.nl
tel.: +31-20 525 2072 / +31-20 6274787

Marc **Light**
University of Rochester
Department of Computer-Science
Rochester NY 14627
USA
light@cs.rochester.edu
tel.: +1-716-275-25 69

Dr. Jan Tore **Lonning**
University of Oslo
Department of Linguistics
P.O. Box 1102 Blindern
N-0317 Oslo 3
Norway
jtl@ulrik.uio.no
tel.: +47 22 85 69 71
fax.: +47 22 85 69 19

Prof. Dr. Bernd **Mahr**
FB Informatik ISTI FR 6-10
Franklinstr. 28-29
W-1000 Berlin 10
Germany
zippke@cs.tu-berlin.de / ww@cs.tu-berlin.de
tel.: +49-30-314-7 35 41/0
fax.: +49-30-314-7 36 22

Dipl.-Inform. Sebastian **Millies**
Universität des Saarlandes
FR 8.7 Computerlinguistik
Im Stadtwald 15
W-6600 Saarbrücken 11
Germany
millies@coli.uni-sb.de
tel.: +49-681-302 4497
fax.: +49-681-302-4351

Dr. Robert **Moore**
SRI International
333 Ravenswood Ave.
Menlo Park CA 94025
USA
bmoore@ai.sri.com
tel.: +1-415-859 3448
fax.: +1-415-859 3735

Dr. Hideyuki **Nakashima**
Electrotechnical Laboratories
1-1-4 Umezono
Tsukuba Ibariki 305
Japan
nakashim@etl.go.jp
tel.: +81-298-58-58 53 / 51 83
fax.: +81-298-58-51 84

Dr. Bernhard **Nebel**
DFKI Saarbrücken
Stuhlsatzenhausweg 3
W-6600 Saarbrücken 11
Germany
nebel@dfki.uni-sb.de
tel.: +49-681-302-5254
fax.: +49-681-302-53 41

Prof. Dr. John **Nerbonne**
Rijksuniversiteit Groningen
P. O. Box 716
NL-9700 AS Groningen
The Netherlands
nerbonne@let.rug.nl
fax.: +31-50-634 900

Prof. Jeffry **Pelletier**
University of Alberta
Computer Science Dept.
Edmonton Alberta T6G 2H1
Canada
jeffp@cs.ualberta.ca

Harold José **Peredes-Frigolet**
AITEC
Alter Hellweg 50
W-4600 Dortmund 70
Germany

Stanley **Peters**
Stanford University
CSLI
Ventura Hall
Stanford CA 94305-4115
USA
peters@csli.stanford.edu
tel.: +1-415-723 2212
fax.: +1-415-723 0758

Prof. Dr. Manfred **Pinkal**
Universität des Saarlandes
FR 8.7 Computerlinguistik
Im Stadtwald 15
W-6600 Saarbrücken 11
pinkal@coli.uni-sb.de / sek-mp@coli.uni-sb.de
tel.: +49-681-302 4344
fax.: +49-681-302 4351

Massimo **Poesio**
University of Rochester
Department of Computer Science
Rochester NY 14627-0226
USA
poesio@cs.rochester.edu
tel.: +1-716-275-5605

Dr. Stephen G. **Pulman**
SRI International
Suite 23
Millers Yard
Mill Lane
Cambridge CB2 1RQ
Great Britain
pulman@cam.sri.com
tel.: +44-223-32 41 46
fax.: +44-223-46 04 02

Dr. Uwe **Reyle**
Universität Stuttgart
IMSV-Institut für masch. Sprachverarbeitung
Azenbergstr. 12
W-7000 Stuttgart
ims@rusvx2.rus.uni-stuttgart.dbp.de

Prof. Remko J. **Scha**
University of Amsterdam
Computational Linguistics
Spuistraat 134
NL-1012 VB Amsterdam
The Netherlands
scha@alf.let.uva.nl

Prof. Dr. Lenhart **Schubert**
University of Rochester
Department of Computer-Science
Rochester NY 14627-0226
USA
schubert@cs.rochester.edu
fax.: +1-716-461-2018

Prof. Mark **Steedman**
University of Pennsylvania
Dept. of Computer and Information Science
220 South 33rd Street
Philadelphia PA 19104-6389
USA
steedman@linc.cis.upenn.edu
tel.: +1-215-8 98-20 12

Prof. Richmond H. **Thomason**
University of Pittsburgh
Intelligent Systems Program
Pittsburgh PA 15260
USA
thomason@pogo.isp.pitt.edu
tel.: +1-412-624-5791
fax.: +1-412-624-6089

Dr. Frank **Veltman**
University of Amsterdam
Nieuwe Doelenstraat 15
NL-1012 CP Amsterdam
The Netherlands
veltman@illc.uva.nl
tel.: +31-20-525 4564
fax.: +31-20-525 4503

Espen **Vestre**
Universität des Saarlandes
FR 8.7 Computerlinguistik
Im Stadtwald 15
W-6600 Saarbrücken 11
espen@coli.uni-sb.de
tel.: +49-681-302 4501

Prof. Dr. Wolfgang **Wahlster**
DFKI Saarbrücken
Stuhlsatzenhausweg 3
W-6600 Saarbrücken 11
Germany
wahlster@dfki.uni-sb.de
tel.: +49-681-302 5252
fax.: +49-681-302 5341

Dr. Henk **Zeevat**
University of Amsterdam
Computational Linguistics
Spuistraat 134
NL-1012 VB Amsterdam
The Netherlands
henk@mars.let.uva.nl
tel.: +31-20-525 46 38

Prof. Dr. Frans **Zwarts**
Rijksuniversiteit Groningen
Institute for Behavioral Cognitive and
Neuro-Sciences
Postbus 716
NL-9700 AS Groningen
The Netherlands
frans@let.rug.nl
tel.: +31-50-63 58 59
fax.: +31-50-63 59 79

Zuletzt erschienene und geplante Titel:

B. Booß, W. Coy, J.-M. Pflüger (editors):
Limits of Information-technological Models, Dagstuhl-Seminar-Report; 31, 10.-14.2.92 (9207)

N. Habermann, W.F. Tichy (editors):
Future Directions in Software Engineering, Dagstuhl-Seminar-Report; 32; 17.2.-21.2.92 (9208)

R. Cole, E.W. Mayr, F. Meyer auf der Heide (editors):
Parallel and Distributed Algorithms; Dagstuhl-Seminar-Report; 33; 2.3.-6.3.92 (9210)

P. Klint, T. Reps, G. Snelting (editors):
Programming Environments; Dagstuhl-Seminar-Report; 34; 9.3.-13.3.92 (9211)

H.-D. Ehrich, J.A. Goguen, A. Sernadas (editors):
Foundations of Information Systems Specification and Design; Dagstuhl-Seminar-Report; 35; 16.3.-19.3.9 (9212)

W. Damm, Ch. Hankin, J. Hughes (editors):
Functional Languages:
Compiler Technology and Parallelism; Dagstuhl-Seminar-Report; 36; 23.3.-27.3.92 (9213)

Th. Beth, W. Diffie, G.J. Simmons (editors):
System Security; Dagstuhl-Seminar-Report; 37; 30.3.-3.4.92 (9214)

C.A. Ellis, M. Jarke (editors):
Distributed Cooperation in Integrated Information Systems; Dagstuhl-Seminar-Report; 38; 5.4.-9.4.92 (9215)

J. Buchmann, H. Niederreiter, A.M. Odlyzko, H.G. Zimmer (editors):
Algorithms and Number Theory, Dagstuhl-Seminar-Report; 39; 22.06.-26.06.92 (9226)

E. Börger, Y. Gurevich, H. Kleine-Büning, M.M. Richter (editors):
Computer Science Logic, Dagstuhl-Seminar-Report; 40; 13.07.-17.07.92 (9229)

J. von zur Gathen, M. Karpinski, D. Kozen (editors):
Algebraic Complexity and Parallelism, Dagstuhl-Seminar-Report; 41; 20.07.-24.07.92 (9230)

F. Baader, J. Siekmann, W. Snyder (editors):
6th International Workshop on Unification, Dagstuhl-Seminar-Report; 42; 29.07.-31.07.92 (9231)

J.W. Davenport, F. Krückeberg, R.E. Moore, S. Rump (editors):
Symbolic, algebraic and validated numerical Computation, Dagstuhl-Seminar-Report; 43; 03.08.-07.08.92 (9232)

R. Cohen, R. Kass, C. Paris, W. Wahlster (editors):
Third International Workshop on User Modeling (UM'92), Dagstuhl-Seminar-Report; 44; 10.-13.8.92 (9233)

R. Reischuk, D. Uhlig (editors):
Complexity and Realization of Boolean Functions, Dagstuhl-Seminar-Report; 45; 24.08.-28.08.92 (9235)

Th. Lengauer, D. Schomburg, M.S. Waterman (editors):
Molecular Bioinformatics, Dagstuhl-Seminar-Report; 46; 07.09.-11.09.92 (9237)

V.R. Basili, H.D. Rombach, R.W. Selby (editors):
Experimental Software Engineering Issues, Dagstuhl-Seminar-Report; 47; 14.-18.09.92 (9238)

Y. Dittrich, H. Hastedt, P. Schefe (editors):
Computer Science and Philosophy, Dagstuhl-Seminar-Report; 48; 21.09.-25.09.92 (9239)

R.P. Daley, U. Furbach, K.P. Jantke (editors):
Analogical and Inductive Inference 1992 , Dagstuhl-Seminar-Report; 49; 05.10.-09.10.92 (9241)

E. Novak, St. Smale, J.F. Traub (editors):
Algorithms and Complexity for Continuous Problems, Dagstuhl-Seminar-Report; 50; 12.10.-16.10.92 (9242)

J. Encarnação, J. Foley (editors):
Multimedia - System Architectures and Applications, Dagstuhl-Seminar-Report; 51; 02.11.-06.11.92 (9245)

F.J. Rammig, J. Staunstrup, G. Zimmermann (editors):
Self-Timed Design, Dagstuhl-Seminar-Report; 52; 30.11.-04.12.92 (9249)

B. Courcelle, H. Ehrig, G. Rozenberg, H.J. Schneider (editors):
Graph-Transformations in Computer Science, Dagstuhl-Seminar-Report; 53; 04.01.-08.01.93 (9301)

A. Arnold, L. Priese, R. Vollmar (editors):
Automata Theory: Distributed Models, Dagstuhl-Seminar-Report; 54; 11.01.-15.01.93 (9302)

W. Cellary, K. Vidyasankar , G. Vossen (editors):
Versioning in Database Management Systems, Dagstuhl-Seminar-Report; 55; 01.02.-05.02.93 (9305)

B. Becker, R. Bryant, Ch. Meinel (editors):
Computer Aided Design and Test , Dagstuhl-Seminar-Report; 56; 15.02.-19.02.93 (9307)

M. Pinkal, R. Scha, L. Schubert (editors):
Semantic Formalisms in Natural Language Processing, Dagstuhl-Seminar-Report; 57; 23.02.-26.02.93 (9308)

W. Bibel, K. Furukawa, M. Stickel (editors):
Deduction , Dagstuhl-Seminar-Report; 58; 08.03.-12.03.93 (9310)

H. Alt, B. Chazelle, E. Welzl (editors):
Computational Geometry, Dagstuhl-Seminar-Report; 59; 22.03.-26.03.93 (9312)

H. Kamp, J. Pustejovsky (editors):
Universals in the Lexicon: At the Intersection of Lexical Semantic Theories, Dagstuhl-Seminar-Report; 60; 29.03.-02.04.93 (9313)

W. Strasser, F. Wahl (editors):
Graphics & Robotics, Dagstuhl-Seminar-Report; 61; 19.04.-22.04.93 (9316)

C. Beeri, A. Heuer, G. Saake, S. Urban (editors):
Formal Aspects of Object Base Dynamics , Dagstuhl-Seminar-Report; 62; 26.04.-30.04.93 (9317)

R. V. Book, E. Pednault, D. Wotschke (editors):
Descriptional Complexity, Dagstuhl-Seminar-Report; 63; 03.05.-07.05.93 (9318)

H.-D. Ehrig, F. von Henke, J. Meseguer, M. Wirsing (editors):
Specification and Semantics, Dagstuhl-Seminar-Report; 64; 24.05.-28.05.93 (9321)

M. Droste, Y. Gurevich (editors):
Semantics of Programming Languages and Algebra, Dagstuhl-Seminar-Report; 65; 07.06.-11.06.93 (9323)

Ch. Lengauer, P. Quinton, Y. Robert, L. Thiele (editors):
Parallelization Techniques for Uniform Algorithms, Dagstuhl-Seminar-Report; 66; 21.06.-25.06.93 (9325)

G. Farin, H. Hagen, H. Noltemeier (editors):
Geometric Modelling, Dagstuhl-Seminar-Report; 67; 28.06.-02.07.93 (9326)

Ph. Flajolet, R. Kemp, H. Prodinger (editors):
"Average-Case"-Analysis of Algorithms, Dagstuhl-Seminar-Report; 68; 12.07.-16.07.93 (9328)

J.W. Gray, A.M. Pitts, K. Sieber (editors):
Interactions between Category Theory and Computer Science, Dagstuhl-Seminar-Report; 69; 19.07.-23.07.93 (9329)

D. Gabbay, H.-J. Ohlbach (editors):
Automated Practical Reasoning and Argumentation, Dagstuhl-Seminar-Report; 70; 23.08.-27.08.93 (9334)