Automated Theorem Proving for Natural Language Understanding

Patric Blackburn, Johan Bos, Michael Kohlhase
Universität des Saarlandes, Saarbrücken, Germany

Hans de Nivelle
Centrum voor Wiskunde en Informatica,
Universiteit van Amsterdam, Amsterdam, The Netherlands

1 Introduction

In this note we discuss an application of automated theorem proving techniques to natural-language processing. Semantic analysis – inference on the basis of semantic information and world knowledge – is one of the central cognitive tasks in NL processing. It is needed for situation-dependent disambiguation and for the coherent embedding of utterances into the discourse context. Humans obviously have at their disposal very efficient techniques for semantic analysis that allow them to infer the information that is relevant for communication.

In natural language processing, similarly powerful techniques have yet to be found. Early attempts from artificial intelligence [Win71,Cul78,Rie75], have had some limited success, but have failed to scale up to real-world examples and to current representation formalisms discussed in computational linguistics.

On the other hand, the field of automated theorem proving has had notable successes and is moving towards applications. However for several reasons it is not clear whether the systems can be used as off-the-shelf components for natural language understanding systems:

1. First-order predicate logic is not well-suited as a representation language for the semantical structures of natural language (see section 2),
2. Automated theorem provers are optimized towards finding deep combinatorially complex proofs of (mathematical) theorems rather than towards the rather shallow and straightforward proofs needed for semantical analysis,
3. Many of the inference problems necessary for semantical analysis are satisfiable and termination is not a priority goal of current automated theorem proving systems.

In this note we investigate the possibility of using a translation approach together with an automated theorem prover on several discourse inference problems encountered in semantic analysis, as suggested in [BB98]. The translation from dynamic logic (see the next section) to first-order logic (FOL) allows us to get around problem 1 and test objections 2 and 3.

In the experiment we have used the **`Bliksem`** theorem prover (see section 4) as a logic engine since it is terminating on a fragment (guarded logic, see section 4) that is close to the one generated by the translation procedure. The natural language front-end is built upon the computational semantics tools provided by [BB98]. This system constructs discourse representations for a small fragment of English, dealing with phenomena like scope ambiguities, pronoun resolution and presupposition projection, and eventually translates the result into first-order logic. The emphasis of the system is on the semantic analysis phase, where (spurious) ambiguities that are artefacts of the specific semantics construction process are analyzed and eliminated. For this, the system generates first-order deduction problems (see section 3) that are solved by passing them to the integrated theorem prover `Bliksem` that runs as a slave process. Thus in this application, the automated theorem prover is treated as a *logic engine* that answers satisfiability questions for the natural language system (its master).

Thus in this application, it is essential that the theorem prover performs well on so-called *junk theorems*, i.e. theorems that are of no intrinsic interest (as a consequence the proofs are irrelevant), which have to be tested efficiently and fully automatically.

2 Dynamic Representation Formalisms

One of the main problems with first-order predicate logic for representing natural language is that the accessibility of discourse referents (modeled as bound variables) is given by the logical scope induced by first-order quantification, which is insufficient to model phenomena like anaphoric references. Consider for instance in (1) the sentence (a) with the intuitive representation (b)

(1)a. A man walks.
b. $\exists x(man(x) \wedge walks(x))$
c. He whistles.
d. $\exists x(man(x) \wedge walks(x) \wedge wh(x))$

This representation makes a continuation of the discourse with (1.c) impossible, since the scope of $\exists x$ is closed and thus the bound variable x is inaccessible to the anaphoric reference with the pronoun *He*.

The so-called dynamic approaches to natural language semantics (Discourse Representation Theory (DRT, see e.g. [KR93]) or dynamic predicate logic (DPL [GS91])) have been developed to cope with this (and related) problems and are now well-established as representation formalisms for natural language semantics.

2.1 Discourse Representation Theory

We will concentrate on DRT in this paper. There, objects that are dynamically introduced in a discourse are not represented by bound variables but by so-called *discourse referents* in *discourse reperesentation structures* (DRS), which collect discourse referents and information about them. Let us now briefly present (some of) the ideas behind DRT by way of an example to make this paper self-contained. We refer the reader to [KR93] for details.

(1.a) is represented by the first box (DRS) and (1.c) by the second one. Both DRSes are then combined by the so-called *merge* operation $\otimes$ that combines two DRSes by collecting the discourse referents and the information about them into one DRS.

$$(2)\qquad \begin{array}{|c|}\hline u\\\hline man(u)\\\hline\end{array} \quad\otimes\quad \begin{array}{|c|}\hline \\\hline wh(u)\\\hline\end{array} \quad\longrightarrow\quad \begin{array}{|c|}\hline u\\\hline man(u)\\ wh(u)\\\hline\end{array}$$

It is important to note that the discourse referent u introduced in the first DRS is accessible (for anaphoric binding by *He*) in the second DRS. In fact, DRT comes with a specialized *accessibility relation* to model the scope properties of natural language. The semantics construction in DRT is modeled as a process, where a DRS is constructed for each new sentence that is then merged into the DRS for the whole discourse.

The semantics of DRSs is given by way of the following transformation $\overline{\cdot}$ into predicate logic,

$$\overline{\begin{array}{|c|}\hline u_1,\ldots,u_n\\\hline C_1,\ldots,C_m\\\hline\end{array}} = \exists u_1,\ldots,u_n.\overline{C_1}\wedge\ldots\wedge\overline{C_m}$$

$$\overline{\neg D} = \neg\overline{D}$$

$$\overline{\begin{array}{|c|}\hline u_1,\ldots,u_n\\\hline C_1,\ldots,C_m\\\hline\end{array} \rightarrow \begin{array}{|c|}\hline v_1,\ldots,n_l\\\hline D_1,\ldots,D_k\\\hline\end{array}} =$$
$$\forall u_1,\ldots,u_n.\overline{C_1}\wedge\ldots\wedge\overline{C_m}\Rightarrow\exists v_1,\ldots,v_l.\overline{D_1}\wedge\ldots\wedge\overline{D_k}$$

In our example discourse (1) we would obtain (1.d) as the translation of (2).

2.2 Deduction in Dynamic Logic

There are two approaches to inferencing in dynamic logics. The first – which we pursue in this note – is to use the (dynamic) deduction theorem to encode the (dynamic) entailment problem as a (dynamic) satisfiability problem (a DRS) and then translate that DRSs to FOL and test for satisfiabilty there. The second is paradigm is to develop a calculus for (dynamic) entailment or satisfiability that operates on the dynamic structures themselves (see [Sau93,RG94,MdR98] for calculi). While the second (more sepcialized) approach promises better results in the long run, the first approach allows to make use of the highly developed automated theorem proving systems that are available today.

The translation approach can also be varied in the translation that is employed. Jan van Eijck has developed an alternative (linear complexity) translation (see e.g. [vEK96]) using the weakest-precondition-calculus. It remains to be seen how the FOL fragment generated by this translation compares to that of our naive translation.

3 Inference Problems in Semantic Analysis

In this section, we will take a closer look a three classes of inference problems occurring during the semantic analysis phase of natural language processing. At this stage, the discourse has already undergone syntactic processing and semantic construction, which together have generated a set of discourse representation structures. This set can be quite large, due to ambiguities that arise from well known phenomena as quantifier scope and anaphora, and one way to deal with this is imposing conditions on the DRSs that decrease the number of readings.

We will begin with discussing three such conditions on discourses, and give examples that violate these three conditions. After that we discuss a number of (basic) examples that show the use of deduction in natural language analysis. All the examples are English sentences or sequences of sentences. Some of the sequences are not acceptable as a coherent discourse, because contributions to the discourse are non-informative or inconsistent. Further, there are examples that show how deduction can rule out readings generated by ambiguous input due to presuppositions or quantifier scope.

For each example we give the translation into discourse representation structures, and, using the translation to first-order logic, how we could use theorem provers to select readings or to rule out discourse.

Obeying basic conversational principles [Sta79], an assertion to a discourse should be informative and consistent, i.e. every utterance in a discourse should contribute something that is still unknown and it should not lead to obvious contradictions, since this is logically problematic. Clearly, these principles are only non-trivial if they are applied with respect to a given set of world knowledge so that checking these principles leads to inference problems.

3.1 Informativity

In [VdS92], Van der Sandt models the informativity principle as follows: a DRS B' is informative with respect to a DRS B, iff B does not entail B'.

With the background knowledge that someone who has a husband is married, the discourse

(3) Mia has a husband. She is married.

violates this principle. The DRSs after processing the first and second sentence respectively, are:

(4) a. $\begin{array}{|c|}\hline A,B \\ \hline A{=}mia \\ husband(B) \\ have(A,B) \\ \hline \end{array}$ b. $\begin{array}{|c|}\hline A,B \\ \hline A{=}mia \\ husband(B) \\ have(A,B) \\ married(A) \\ \hline \end{array}$

The background knowledge about marriage is coded into first-order logic:

(5) $woman(mia) \wedge \forall X(\exists Y.husband(Y) \wedge have(X,Y)) \Leftrightarrow married(X) \wedge woman(X)$

This translates as: everyone who has a husband is married and a woman, and every one who is a woman and married has a husband.

Note that in the approach advocated in this paper it is easy to integrate static background knowledge (given in FOL) with DRT, since the latter is translated to FOL anyway.[1]

In this situation, we can test informativity by checking whether

(6) $(5) \wedge \overline{(4.a)} \Rightarrow (4.b)$

is a theorem of first-order logic. In our example (see A.1 for the FOL representation), we should find a proof, as there is no new information conveyed by the second sentence.

3.2 Local Informativity and Consistency

Next we discuss a variation of informativity, the *local informativity* constraint. The local informativity constraint is that if one utters a phrase of the form: If A then B, then A should be not trivially satisfied. (So, here the if-then from natural language clearly differs from its logical counterpart). In the following example this condition is violated:

(7) Mia has a husband. If she is married, then Pete dances.

The DRSs belonging to these sentences are:

(8) *a.*

$$\begin{array}{|c|}\hline A,B \\ \hline A = mia \\ husband(B) \\ have(A,B) \\ \hline \end{array}$$

b.

$$\begin{array}{|c|}\hline C \\ \hline C = mia \\ \begin{array}{|c|}\hline \\ \hline married(C) \\ \hline \end{array} \rightarrow \begin{array}{|c|}\hline D \\ \hline D = pete \\ dances(D) \\ \hline \end{array} \\ \hline \end{array}$$

When these DRS's are combined the result equals:

(9)

$$\begin{array}{|c|}\hline A,B,C \\ \hline A = mia \\ C = A\ husband(B) \\ have(A,B) \\ \begin{array}{|c|}\hline \\ \hline married(C) \\ \hline \end{array} \rightarrow \begin{array}{|c|}\hline D \\ \hline D = pete \\ dances(D) \\ \hline \end{array} \\ \hline \end{array}$$

This last DRS violates the local informativity constraint, since

(10) $(5) \models (A = mia, have(A,B), husband(B)) \Rightarrow married(A).$

[1] In a dynamic deduction approach, the background knowledge would have to be formulated in DRT, or the approach would need to be extended to accomodate for FO reasoning.

3.3 Consistency

The last condition that we check is *consistency*. Say we had continued (4.a) with the utterance

(11) *a. She is not married.* *b.* $\neg$ [*married(A)*]

Clearly, the new information is inconsistent with the information that is already present (implicily) In this situation, we can check for informativity by checking whether

(12) $(5) \wedge \overline{(4.a) \otimes (11)}$

is unsatisfiable.

3.4 Presupposition Projection

Another problem, where inferences are needed in semantic analysis is the phenomenon of *presuppositions* in natural language. For instance, the sentence (13) is only sensible, if country XXX indeed has a king. If the presupposition is not satisfied, then the truth value of the sentence cannot be evaluated. In particular, the informativity and consistency principles discussed above lead to the hearer's assumption that XXX is not a republic and that there is a king. We say that (13) *presupposes* the existence of a king in XXX.

(13) *The king of XXX is bald*

This phenomenon is so widespread that natural language semantics has to accommodate for presupposed material and for inferences with presuppositions (see [Bea97] for a survey and [KK] for a mechanization using automated theorem proving techniques). The conversational principles are so strong that they lead to a process called *accommodation* when violated: If the presupposed material is inaccessible in the current conversational context, it is assumed to be true, in order to preserve the conversational principles. Since DRT offers a structured representation of meaning, it is not a priori clear, where the new information should be inserted (accomodated) in the structure: Elementary presuppositions can be linked to antecedents or accommodated to super-ordinated levels of discourse representation.

In our experiment, we use a generate-and-test algorithm for presupposition projection proposed by Rob van der Sandt in [VdS92]. This algorithm is set up in such a way that elementary presuppositions are tried to link to appropriate an antecedent, or, if that fails, to accommodate it to a superordinated level of discourse representation. The initial output of this algorithm is a set of DRSs (there could be more than one suitable antecedent or accommodation sites), from which all members that violate conversation principles, such as informativity or consistency (discussed in sections 3.1 to 3.3) are removed.

Before we take a look at the testing phase, let us look at an example discourse.

(14) Mia dances. If she is married, then her husband is out of town.

The consequent in the second (conditional) sentence of this discourse presupposes a context where Mia has a husband (the relevant presupposition trigger is her husband). The initial output of Van der Sandt's algorithm generates three readings, corresponding to the *global* (15), *intermediate* (16), and *local* (17) accommodation of the presupposition (we have highlighted the accomodated material):

(15) *global:*

A,B
dance(A)
A=mia
[| *married(A)*] → [| *outoftown(B)*]
of(B,A)
husband(B)

(16) *intermediate:*

A
dance(A)
A=mia
[*B* | *married(A)* **of(B,A)** **husband(B)**] → [| *outoftown(B)*]

(17) *local:*

A
dance(A)
A=mia
[| *married(A)*] → [*B* | *outoftown(B)* **of(B,A)** **husband(B)**]

In this situation, local informativity plays an important rôle. Each conditionally subordinated DRS should, if it were true, convey new information (with respect to the DRSs which embed them). Thus the global reading (15) (taking our knowledge about marriage of example 1 into account) is ruled out and therefore deleted from the set of possible readings of (14). This is correct, as the discourse does not presuppose that Mia has a husband. Why is that? Simply because the global DRS contains the information that Mia has a husband, and this is expressed in the DRS that builds the antecedent of the implicational condition. The intermediate and local reading (16 and 17) are locally informative, (the consequent DRS in the local reading adds the information that Mia's husband is out of town).[2]

[2] Actually, given the right background knowledge these readings are equivalent, so that the discourse is not ambiguous at all. In this we are not making any claim

How do we translate the local-informativity constraint into something suitable for a theorem prover? Let B_0 encode the background information, and B the new DRS. Then the DRS

(18) $$B_0 \models \overline{\begin{array}{|c|}\hline \\ \hline B' \rightarrow B_{Local} \\ \hline\end{array}}$$

with B_{Local} a DRS appearing in one of the conditions of B, and B′ the DRS B merged with all DRSs within B that subordinate B_{Local}. If the translation of this DRS to first-order logic is a theorem, then the local-informativity constraint is violated.

For (15), for instance, we get (among others):

(19) $$B_0 \models \overline{\begin{array}{|c|}\hline A,B \\ \hline dance(A) \\ A{=}mia \\ of(B,A) \\ husband(B) \\ \hline\end{array} \rightarrow \begin{array}{|c|}\hline \\ \hline married(A) \\ \hline\end{array}}$$

which is a theorem and therefore locally non-informative.

3.5 Quantifier Scope

The consisteny constraint on discourse can help us to resolve ambiguity. One of the well known cases of ambiguity are those introduced by quantifier scope, as in:

(20) Every car has an owner.

where the universally quantifying noun phrase every car can out-scope the existentially quantifying noun phrase an owner (this is the reading where there might be different owners for the different cars), or vice versa (all cars are owned by one and the same person).

However, using knowledge of the world, some readings induced by quantifier scope ambiguities are not available, as the following example shows:

(21) Every car has a radio.

Knowing that no radio can be in two different cars at the same time, the only reading that makes sense is the one where the universal quantifier out-scopes the existential one. That is, given the readings for this sentence encoded as (22a) and (22b).

(22) a. $$\begin{array}{|c|}\hline \\ \hline \begin{array}{|c|}\hline A \\ \hline car(A) \\ \hline\end{array} \rightarrow \begin{array}{|c|}\hline B \\ \hline radio(B) \\ have(A,B) \\ \hline\end{array} \\ \hline\end{array}$$ b. $$\begin{array}{|c|}\hline A \\ \hline radio(A) \\ \begin{array}{|c|}\hline B \\ \hline car(B) \\ \hline\end{array} \rightarrow \begin{array}{|c|}\hline \\ \hline have(B,A) \\ \hline\end{array} \\ \hline\end{array}$$

about the linguistic adequacy of the linguistic methods employed, but rather show that automated theorem provers can be employed naturally in such situations.

and the encoded world knowledge in

(23) $\forall X, Y, Z.radio(X) \wedge car(Y) \wedge car(Z) \wedge have(Y, X) \wedge have(Z, X) \Rightarrow X \neq Y$

we should test

(24) $(23) \wedge \overline{(22.b)}$

for unsatisfiability to rule out the unwanted reading.

4 The `Bliksem` Theorem Prover

`Bliksem` is a resolution based theorem prover which been designed with two objectives. First it should be a technically efficient prover which uses optimizations like state-of-the-art subsumption algorithms and indexing techniques. On the other hand **`Bliksem`** is designed to to be theoretically up to date. That means that **`Bliksem`** offers many different transformations to clausal normal form, and many different restrictions of resolution. In particular, **`Bliksem`** supports *resolution decision procedures*, i.e. restrictions of resolution which can be shown to terminate for certain subsets of first order logic, thus deciding the satisfiability problem for these classes (See [FLTZ93] for a survey).

This emphasis on decision procedures is the key reason why we have chosen to use **`Bliksem`** in our natutal language processing application, where termination is mandatory and deciding consistency is equally important as unsatisfiability. In particular, many of the examples encountered in our experiment are in (or close to) the so-called *guarded fragment* [AvBN96], which can be decided by the **`Bliksem`** theorem prover [dN98]. The reason for this is that new objects, when they are introduced, are related to the old objects through some relation. It is very unnatural in natural language to introduce a totally unrelated new object.

4.1 The Guarded Fragment and Translation

The *guarded fragment* is a subset of first order logic, that was introduced in [AvBN96] as the modal fragment of classical logic. It was shown there that the satisfiability problem and the universal validity problem are decidable. A formula is called guarded iff it does not contain function symbols, all universal quantifiers occur as $\forall\overline{x}.(a \rightarrow \Phi)$, and all existential quantifiers occur as $\exists\overline{x}.(a \wedge \Phi)$. Here a is an atom, containing at least the free variables of Φ. This atom a is called *guard*. Different occurrences of quantifiers may have different guards. $\overline{x}$ is a sequence of variables, which need not contain all free variables of Φ. The following formula is guarded:

$$\forall x.a(x) \rightarrow \forall y, z.(b(x, y, z) \rightarrow \neg(c(x, y) \wedge c(y, z)))$$

It has been observed in [Gra] and in [dN98] that the condition on the guard can be dropped for positively occurring $\forall$-quantifiers and negatively occurring

∃-quantifiers. (In the case that one is checking satisfiability). Unfortunately this is not going to help us here, because formulae are used both positively and negatively. Suppose that a new formula B is considered in a discource Γ. Then we want to check that Γ, B is consistent, and we want to know check that not Γ implies B, i.e. that $\Gamma, \neg B$ consistent. Since in the two tests, B is used with different polarities, we cannot drop any of the guard conditions. Nevertheless due to the flexibility it is still interesting to use `Bliksem`, and it is reasonable to assume that it can be tuned towards the underlying type of problem.

4.2 Experiments

Below we give a list of running times for the theorem provers SPASS and `Bliksem`. `Bliksem` was used in the automode, i.e. without a special decision strategy. The files contain the consistency and the informativity checks for Example 1 and Example 2. As `Bliksem` cannot accept DFG-syntax directly, the files had to be converted to `Bliksem` format. This was done with a separate program `dfg2blk`. The times used by `Bliksem` include these translation times.

name of problem	status	time(s) for bliksem	time(s) for SPASS
`d1r1cons`	consistent	0.25	0.26
`d1r1info`	consistent	0.28	0.33
`d2r1cons`	inconsistent	0.20	0.24
`d2r1info`	consistent	0.25	0.34

Below we give times for Example 7. The provers were given a time limit of $120.0s$.

name of problem	status	time(s) for bliksem	time(s) for SPASS
`d7r1cons`	consistent	1.87	> 120.0
`d7r1info`	consistent	19.27	> 120.0
`d7r2cons`	consistent	1.76	> 120.0
`d7r2info`	consistent	18.36	> 120.0
`d7r3cons`	consistent	1.55	> 120.0
`d7r3info`	consistent	18.30	> 120.0
`d7r4cons`	consistent	1.79	> 120.0
`d7r4info`	consistent	19.00	> 120.0
`d7r5cons`	consistent	1.81	> 120.0
`d7r5info`	consistent	18.54	> 120.0
`d7r6cons`	consistent	1.59	> 120.0
`d7r6info`	consistent	18.66	> 120.0

In the last table `Bliksem` outruns SPASS spectactularly but we don't understand why at this moment, as no special decision procedure was selected.

5 Conclusion

In this note we have presented an application for automated theorem provers in natural language processing. The theorem prover `Bliksem` has been successfully employed as an oracle for the NLP system to disambiguate multiple readings.

While the experiment has shown that the naive translation approach to dynamic reasoning is indeed feasible in this application, it is clear that in the presence of larger discourses (the ones tried out so far only consist of a couple of sentences), the techniques have to be refined both from the linguistic side as well as from the theorem proving side. For instance the set of formulae supplied to the automated theorem prover can be restricted by taking into account the discourse structure (see for instance [Gar97]). Theorem provers can be refined to this application by developing inference strategies that are particularly well-suited to the fragment generated by the respective translation algorithm. The observation pertaining to the guarded fragment discussed above is only a starting point here.

Availability

The `Bliksem` theorem prover and the translator from `DFG`-format to `Bliksem`-format are available at `http://www.cwi.nl/~nivelle`. A description of `DFG`-format can be obtained from `http://www.uni-koblenz.de/ag-ki/Deduktion` .
The code that make up the natural language front-end system is available on
`http://www.coli.uni-sb.de/~bos/comsem/`
The test sets can found at `http://www.coli.uni-sb.de/~bos/atp/`.

Acknowledgements

The authors thank Tim Geisler for comments on the example sets. Thanks also to Jan van Eijck for draft reading the paper.

This research of was carried out during a three month stay of Michael Kohlhase at CWI as a guest of project INS3. The CWI support is herewith gratefully acknowledged.

References

[AvBN96] H. Andréka, J. van Benthem, and I. Németi. Modal languages and bounded fragments of predicate logic. Technical Report ML-96-03, Institute for Logic, Language and Computation, 1996.

[BB98] Patrick Blackburn and Johan Bos. Representation and Inference for Natural Language. A First Course in Computational Semantics. Draft available at URL: `http://www.coli.uni-sb.de/~bos/comsem/`, July 1998.

[Bea97] David Ian Beaver. Presupposition. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*, chapter 17, pages 939–1008. Elsevier Science, 1997.

[Cul78] R.E. Cullingford. *Script Application: Computer Understanding of Newspaper Stories*. PhD thesis, Yale University, New Haven, CT, 1978.

[dN98] Hans de Nivelle. A resolution decision procedure for the guarded fragment. In Claude Kirchner and Hélène Kirchner, editors, *Proceedings of the 15th Conference on Automated Deduction*, number 1421 in LNAI, Lindau, , Germany, 1998. Springer Verlag.

[FLTZ93] Christian Fermüller, Alexander. Leitsch, Tannel. Tammet, and N. Zamov. *Resolution for the Decision Problem.* Number 679 in LNCS. Springer Verlag, 1993.

[Gar97] Claire Gardent. Discourse tree adjoining grammars. CLAUS Report 89, University of the Saarland, Saarbrücken, 1997.

[Gra] Ernst Graedel. On the restraining power of guards. To Appear.

[GS91] Jeroen Groenendijk and Martin Stokhof. Dynamic predicate logic. *Linguistics & Philosophy*, 14:39–100, 1991.

[KK] Manfred Kerber and Michael Kohlhase. A resolution calculus for presuppositions. pages 375–379.

[KR93] Hans Kamp and Uwe Reyle. *From Discourse to Logic.* Kluwer, Dordrecht, 1993.

[MdR98] C. Monz and M. de Rijke. Labeled resolution for discourse semantics. In *Proceedings of the First International Workshop on Labelled Deduction (LD'98).* Dept. of Computer Science, University of Freiburg, 1998.

[RG94] Uwe Reyle and Dov M. Gabbay. Direct deductive computation on discourse representation structures. *Linguistics & Philosophy*, 17:343–390, 1994.

[Rie75] Chuck Rieger. The commonsense algorithm as a basis for computer models of human memory, inference, belief and contextual language comprehension. In *Theoretical Issues in Natural Language Processing.* MIT Press, 1975.

[Sau93] Werner Saurer. A natural deduction system for discourse representation theory. *Journal of Philosophical Logic*, 22, 1993.

[Sta79] R. Stalnaker. Assertion. In P. Cole, editor, *Syntax and Semantics 9: Pragmatics*, pages 315–32. Academic Press, 1979.

[VdS92] Rob A. Van der Sandt. Presupposition projection as anaphora resolution. *Journal of Semantics*, 9(4):333–377, 1992.

[vEK96] Jan van Eijck and Hans Kamp. Representing discourse in context. In Johan van Benthem and Alice ter Meulen, editors, *Handbook of Logic and Language*, pages 179–238. Elsevier Science B.V., 1996.

[Win71] Terry Winograd. *Procedures as a representation for data in a computational program for understanding natural language.* PhD thesis, MIT, Cambridge, 1971.

A Theorem Prover Tests

In this appendix we will elaborate on the first-order inference problems posed to the automated theorem prover. This class of examples is posed as a challenge to the theorem prover community. The test sets can found at the repository `http://www.coli.uni-sb.de/~bos/atp/`. which will be updated with larger and more difficult tests reqularly.

A.1 Example 1: Informativity (simple)

The following discourse is not informative (the second sentence does not add new information, assuming that having a husband implies that you're married).

(25) Mia has a husband. She is married.

The following formula, in DFG syntax, codes the principle of informativity for this example.

```
formula(forall([A],forall([B],
               implies(and(forall([C],
                           implies(equal(C,mia),
                                   woman(C))),
                           and(forall([D],forall([E],
                                      implies(have(D,E),
                                              of(E,D)))),
                           and(forall([D],forall([E],
                                      implies(of(D,E),
                                              have(E,D)))),
                           and(not(exists([C],
                                   not(or(not(exists([D],
                                              and(husband(D),
                                                  have(C,D)))),
                                          and(woman(C),
                                              married(C)))))),
                           and(forall([C],
                               implies(and(married(C),woman(C)),
                                       exists([D],and(husband(D),
                                                      have(C,D))))),
                           and(husband(A),
                               and(have(B,A),
                                   equal(B,mia)))))))),
                        exists([A],exists([B],
                               and(husband(A),
                                   and(have(B,A),
                                       and(equal(B,mia),
                                           married(A)))))))))).
```

If this is a theorem (it is), then the second sentence does not contribute new information to the discourse. Note that, for this example, the equality conditions can be compiled out: they always appear between a variable and a constant.

A.2 Example 2: Consistency (simple)

The following discourse is inconsistent. The first sentence asserts that Mia has a husband, hence is married. The second sentence contradicts that.

(26) Mia has a husband. She does not have a husband.

And here is the translation into first-order logic (in DFG syntax).

```
formula(not(exists([A],
            and(forall([B],
                implies(equal(B,mia),
                        woman(B))),
            and(forall([C],forall([D],
                       implies(have(C,D),
                               of(D,C)))),
            and(forall([C],forall([D],
                       implies(of(C,D),
                               have(D,C)))),
            and(not(exists([B],
                    not(or(not(exists([C],
                               and(husband(C),
                                   have(B,C)))),
                           and(woman(B),
                               married(B)))))),
            and(forall([B],
                implies(and(married(B),woman(B)),
                        exists([C],and(husband(C),
                                       have(B,C))))),
            and(married(A),
                and(equal(A,mia),
                    not(exists([E],
                        and(husband(E),
                            have(A,E))))))))))))).
```

If this is a theorem (and it is) then the discourse is determined as inconsistent.

A.3 Example 3: Informativity (equality)

Consider the following discourse. The third sentence contributes no new information to the discourse.

(27) Every boxer is a criminal. Marsellus knows one criminal. Marsellus knows one boxer.

Here is the translation into first-order logic.

```
formula(forall([A],forall([B],
        implies(and(forall([D],implies(boxer(D),criminal(D))),
                    and(forall([E],implies(and(criminal(E),know(B,E)),
```

```
                              equal(A,E))),
        and(criminal(A),
            and(know(B,A),
                equal(B,marsellus)))))))))),
    exists([A],exists([B],exists([F],
           and(forall([D],
               implies(boxer(D),criminal(D))),
           and(forall([E],implies(and(criminal(E),
                                      know(B,E)),
                                  equal(A,E))),
               and(criminal(A),
                   and(know(B,A),
                       and(equal(B,marsellus),
           and(forall([G],implies(and(boxer(G),
                                      know(B,G)),
                                  equal(F,G))),
               and(boxer(F),know(B,F))))))))))))))).
```

Due to the translation of the determiner one, there are equality conditions between variables which cannot be eliminated (in contrast to examples 1 and 2).

A.4 Example 4: Consistency (equality)

With the background knowledge that Mia is a woman, Butch is a man, and that no person can be man and woman at the same time, the following discourse is inconsistent:

(28) Marsellus loves Mia and Butch. Mia is a boxer Butch is a boxer Marsellus loves one boxer

Here is the translation into first-order logic.

```
formula(not(exists([A],exists([B],exists([C],exists([D],
            and(forall([E],
                implies(man(E),not(woman(E)))),
            and(forall([E],
                implies(woman(E),not(man(E)))),
            and(forall([E],
                implies(equal(E,marsellus),man(E))),
            and(forall([E],
                implies(equal(E,butch),man(E))),
            and(forall([E],
                implies(equal(E,mia),woman(E))),
            and(love(A,C),
                and(love(A,B),
                    and(equal(A,marsellus),
                        and(equal(B,butch),
                            and(equal(C,mia),
                                and(boxer(C),
```

```
                              and(boxer(B),
          and(forall([F],
              implies(and(boxer(F),love(A,F)),
                      equal(D,F))),
          and(boxer(D),
              love(A,D))))))))))))))))))))))).
```

A.5 Example 5: Presupposition Projection

This example is discussed in the text extensively and need no further introduction:

(29) Mia dances. If she is married, then her husband is out of town.

The check for local-informativity is crucial her to exclude global accommodation of Mia's husband.

The initial output yields three readings, corresponding to the global, intermediate, and local accommodation of the presupposition.

The local-informativity constraint rules out reading 1 (global accommodation).

A.6 Example 6: Quantifier Scope

This example is discussed in the text too.

(30) Every car has a radio

There are two quantifiers and therefore two readings according to the different ways to assign scope. The second reading should be excluded, given the knowledge that no radio can be part of different cars. Here is coding in first-order logic for the reading where there is one radio for different cars:

```
formula(not(exists([A],
          and(not(exists([B],exists([C],exists([D],
              and(radio(B),
                  and(car(C),
                      and(car(D),
                          and(not(equal(C,D)),
                              and(have(C,B),
                                  have(D,B))))))))))),
          and(radio(A),
              forall([E],
                    implies(car(E),
                            have(E,A)))))))).
```

This input should be satisfiable. In contrast, there should be no proof for the following reading of the discourse:

```
formula(not(and(not(exists([A],exists([B],exists([C],
                    and(radio(A),
                        and(car(B),
                            and(car(C),
                                and(not(equal(B,C)),
                                    and(have(B,A),
                                        have(C,A))))))))),
            forall([D],
                   implies(car(D),
                           exists([E],
                                  and(radio(E),
                                      have(D,E)))))))).
```

A.7 Example 7: Informativity (long)

This is test for informativity on "long" discourses to see how provers fare with large input.

(31) Butch is a boxer. He works for a criminal who is out of town. Marsellus is a criminal Vincent or jules is a criminal. One criminal has a shotgun. Pumpkin is a robber. He has a gun Every robber is a criminal. If jules has a suitcase then pumpkin does not have the suitcase. Vincent is an owner of the suitcase. Vincent dates Mia. If Mia told a joke then Vincent enjoys the joke. If Mia has a husband then Marsellus is out of town. Every boxer works for one criminal. The criminal is out of town. Mia is married.

Even for relatively small texts like this one it is difficult for humans to find out if every contribution yields new information.

A.8 Example 8: Consistency (long)

A similar test as 7, but now on consistency.

(32) Butch is a boxer. He works for a criminal who is out of town. Marsellus is a criminal. Vincent or Jules has the suitcase. One criminal has a shotgun. Pumpkin is a robber, he has a gun. Every robber is a criminal. If Jules has the suitcase then Pumpkin does not have the suitcase. Jules is a owner of the suitcase. Vincent dates Mia. If mia told a joke then Vincent enjoys the joke. If mia has a husband then Marsellus is out of town. Every boxer works for one criminal. Pumpkin has the suitcase. Mia is married.

As for example 7, the translation of this discourse into first-order logic is not included, but to be found at the WWW-site mentioned in the text.