

# Two Accounts of Scope Availability and Semantic Underspecification

Alistair Willis and Suresh Manandhar,  
Department of Computer Science,  
University of York,  
York YO10 5DD, UK.  
{agw,suresh}@cs.york.ac.uk

## Abstract

We propose a formal system for representing the available readings of sentences displaying quantifier scope ambiguity, in which *partial* scopes may be expressed. We show that using a theory of scope availability based upon the function-argument structure of a sentence allows a deterministic, polynomial time test for the availability of a reading, while solving the same problem within theories based on the well-formedness of sentences in the meaning language has been shown to be NP-hard.

## 1 Introduction

The phenomenon of quantifier scope ambiguity has been discussed extensively within computational and theoretical linguistics. Given a sentence displaying quantifier scope ambiguity, such as *Every man loves a woman*, part of the problem of representing the sentence's meaning is to distinguish between the two possible meanings:

$$\forall x(\text{man}(x) \rightarrow \exists y(\text{woman}(y) \wedge \text{love}(x, y)))$$

where every man loves a (possibly) different woman, or

$$\exists y(\text{woman}(y) \wedge \forall x(\text{man}(x) \rightarrow \text{love}(x, y)))$$

where a single woman is loved by every man. One aspect of the problem is the generation of all available readings in a suitable representation language. Cooper (1983) described a system of "storing" the quantifiers as  $\lambda$ -expressions during the parsing process and retrieving them at the sentence level; different orders of quantifier retrieval generate different readings of the sentence. However, Cooper's method generates logical forms in which variables are not correctly

bound by their quantifiers, and so do not correspond to a correct sentence meaning. This problem is rectified by nested storage (Keller, 1986) and the Hobbs and Shieber (1987) algorithm. However, the linguistic assumptions underlying these approaches have recently been questioned. Park (1995) has argued that the availability of readings is determined not by the well-formedness of sentences in the meaning language, but by the function-argument relationships within the sentence. His theory proposes that only a subset of the well-formed sentences generated by nested storage are available to a speaker of English. Although the theories have different generative power, it is difficult to find linguistic data that convincingly proves either theory correct.

In the absence of persuasive linguistic data, it is reasonable to ask whether other grounds exist for choosing to work with either of the two theories. This paper considers the application of both theories to the problem of underspecified meaning representation, and the question of determining whether a set of constraints represents an available reading of an ambiguous sentence or not. We show that a constraint language based upon Park's linguistic theory (Willis and Manandhar, 1999) solves this problem in polynomial time, and contrast this with recent work based on dominance constraints which shows that using the more permissive theory of availability to solve the same problems leads to NP-hardness.

## 2 Underspecification

A recent area of interest has been with *underspecified* representations of an ambiguous sentence's meaning, for example, Quasi-Logical Form (QLF) (Alshawi and Crouch, 1992) and Underspecified Discourse Representation The-

ory (UDRT) (Reyle, 1995). We shall characterise the desirable properties of an underspecified meaning representation as:

1. the meaning of a sentence should be represented in a way that is not committed to any one of the possible (intended) meanings of the sentence, and
2. it should be possible to incrementally introduce *partial* information about the meaning, if such information is available, and without the need to undo work that has already been done.

A principal aim of systems providing an underspecified representation of quantifier scope is the ability to represent partial scopings. That is, it should be possible to state that some of the quantifiers have some scope relative to each other, while remaining uncommitted to the relative scope of the remaining quantifiers. However, representations which simply allow partial scopes to be stated without further analysis do not adequately capture the behaviour of quantifiers in a sentence. Consider the sentence *Every representative of a company saw most samples*, represented in the style of QLF:

```
_:see(<+i every x _:rep.of(x,
      <+j exists y co(y)>>),
      <+k most z sample(z)>>
```

A fully scoped logical form of this QLF is:

```
[+i,+k,+j]:see(<+i every x rep.of(x,
      <+j exists y co(y)>>),
      <+k most z sample(z)>>
```

where the list of quantifier labels indicates the relative scope of quantifiers at that point in the sentence. Although this formula is well formed in the QLF language, it does not correspond to a well formed sentence of logic, seeming closer to the formula:

```
every(x,rep.of(x,y), most(z,sample(z),
      exists(y, co(y), see(x, z))))
```

where the variable *y* does not appear in the scope of its quantifier. A language such as QLF will generally allow this scoping to be expressed, even though it does not correspond to a reading available to a speaker. In QLF semantics, a scoping which does not give rise to any well formed readings is considered “uninterpretable”; ie. there is no interpretation in which an evaluation function maps the QLF onto a truth value.

Our aim is to present a system in which there is a straightforward computational test of whether a well-formed reading of a sentence exists in which a partial scoping is satisfied, without requiring recourse to the final logical form. The language CLLS (Egg et al., 1998) has recently been developed which correctly generates the well-formed readings by using dominance constraints over trees. Readings of a sentence can be represented using a tree, where dominance represents outscoping, and quantifiers are represented using binary trees whose daughters correspond to the quantifiers’ restriction and scope. So for the current example, *Every representative of a company saw most samples*, the reading:

```
every(x, a(y, co(y), rep.of(x, y)),
      most(z, sample(z), see(x, z)))
```

can be represented by the tree in figure 1, where the restrictions of *a* and *most* have been omitted for clarity. Domination in the tree represents outscoping in the logical form.

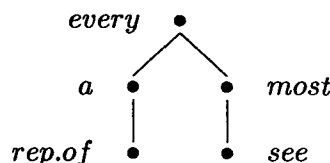


Figure 1: Representing relative scope as a tree

Underspecification can be captured by defining dominance constraints between nodes representing the quantifiers and relations in a sentence. Readings of the sentence with a free variable are avoided by asserting that each relation containing a variable must be dominated by that variable’s quantifier, and an available reading of the sentence is represented by a tree in which all the dominance constraints are satisfied. So the ill-formed readings of the sentence can be avoided by stating that the relation *rep.of* is dominated by the restriction of *every* and the scope of *a*, while *see* is dominated by the scopes of both *a* and *most*. This is represented in figure 2, where the dominance constraints are illustrated by dotted lines.

Further partial scope information can be introduced with additional dominance constraints. So the partial scope requirement that

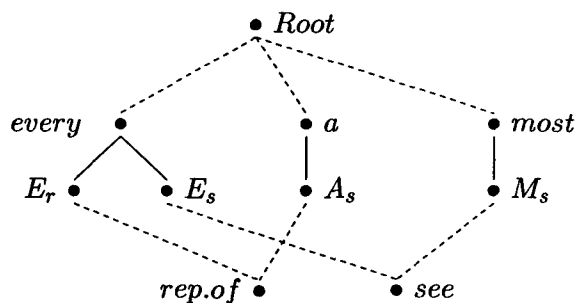


Figure 2: Representing available scopes with dominance constraints

*most* should outscope *every* would be captured by a constraint stating that the node representing *most* should dominate the node representing *every* in the constraints' solution.

It has been shown (Koller et al., 1998) that determining the consistency of these constraints is NP-hard. In the rest of this paper, we show that an alternative theory of scope availability yields a constraint system that can be solved in polynomial time.

### 3 Alternative Account of Availability

The NP-hardness result of the previous section arises from the assumption that the availability of scopings is determined by the well formedness of the associated logical forms. Park (1995) has proposed an alternative theory of scope availability which states that available scopes are accounted for by relative scopes of arguments around relations, whereby quantifiers may not move across NP boundaries. For example, consider the sentence *Every representative of a company saw most samples*, containing two relations, *saw* and *of*. Around *saw*, *every* (*representative of a company*) can outscope *most* (*samples*), or vice versa, and around *of*, *every* (*representative*) can outscope *a* (*company*), or vice versa. Park generalises this observation to the claim that for any  $n$ -ary relation in a sentence, there are  $n!$  possible orderings of quantified arguments around that relation. Other quantifiers in the sentence should not “intercalate” between those which are single arguments to a relation. So in the example sentence there are four possible scopes, because there are  $2! = 2$  scopings around *saw* and  $2! = 2$  scopings around *of*. What is not possible is a reading where *a*

outscores *most* which outscores *every*; although this can be represented by a well formed sentence of logic (with no unbound variables), it is not available to a speaker of English.

By using this theory as the basis of underspecification, we can say:

- underspecification is to be captured by allowing different possible relative scope assignments around the predicates, and
- partial scopes between arbitrary quantifiers in the sentence will be translated into the equivalent scoping of quantifiers around their predicates.

The chosen representation will be based upon a sentence's *quantifiers* and *relations* (for example, verbs and prepositions).

Quantifiers and the relations which determine their relative scope are represented by a set of elements under a strict partial order, where the ordering represents the relative scopes. A strict order will be taken to be transitive, antisymmetric and irreflexive. However, because the interaction between the predicates in the sentence has implications for possible scopings, it is also necessary to consider the relationships between the ordered sets.

Consider again the sentence *Every man loves a woman*. The quantifiers and relation in this sentence can be represented by a set of elements  $\{every, a, love\}$ . A strict partial order,  $\succ$ , is defined over the set which states that the relation *love* must be outscoped by both quantifiers:

$$\langle \{every, a, love\}, (every \succ love, a \succ love) \rangle$$

The partial order states that both quantifiers outscope the verb, but says nothing about their scopes relative to each other. This represents a completely underspecified meaning. An unambiguous reading of the sentence is represented when  $\succ$  defines a total order on the set. So if the relation  $every \succ a$  were added, the reading:

$$\forall x.man(x) \rightarrow \exists y.woman(y) \wedge love(x, y) \\ every \succ a \succ love$$

would be represented. Alternatively, adding  $a \succ every$  to the underspecified form would represent the reading:

$$\exists y.woman(y) \wedge \forall x.man(x) \rightarrow love(x, y) \\ a \succ every \succ love$$

The introduction of a further relation which does not lead to a well formed sentence (such as *love*  $\succ$  *every*) is shown by the irreflexivity of  $\succ$  being violated.

While using a single set of elements correctly accounts for the possible scopes of quantifiers in the sentences discussed so far, relative clauses and prepositional attachment to NPs are more complex. Consider the sentence *Every representative of a company saw most samples*. The presence of two binary relations, *of* and *saw*, implies that there should be  $2! \cdot 2! = 4$  readings. Continuing with the system developed so far, these possibilities could be represented by a pair of strictly partially ordered sets:

$$\langle \{every, most, see\}, (every \succ see, most \succ see) \rangle \\ \langle \{every, a, of\}, (every \succ' of, a \succ' of) \rangle$$

where the four possible ways of completing the strict orders on the sets correspond to the four available readings. To represent relative scope between arbitrary quantifiers in the sentence, a further transitive relation,  $\succ$ , is defined. Say that if  $\langle S, \succ \rangle$  is a strictly partially ordered set in the structure where  $x, y \in S$  and  $x \succ y$  then  $x \succ y$ . So for example, consider the pair of strictly partially ordered sets:

$$\langle \{every, most, see\}, (every \succ most \succ see) \rangle \\ \langle \{every, a, of\}, (a \succ' every \succ' of) \rangle$$

which would represent the reading (in a format similar to generalised quantifiers):

$$a(y, co(y), every(x, rep.of(x, y), \\ most(z, sample(z), see(x, z))))$$

The orders on the sets state that *every*  $\succ$  *most*  $\succ$  *see* and *a*  $\succ$  *every*  $\succ$  *of*, and from the transitivity of  $\succ$  it can be inferred (correctly) that *a*  $\succ$  *most*. Similarly, given the ambiguous sentence and the partial scope requirement that *a* should outscope *most*, the required partial scope can be obtained by adding the relations *a*  $\succ'$  *every* and *every*  $\succ$  *most*.

The transitivity of  $\succ$  is not enough to capture all the available scope information. Suppose it were required that *most* should outscope *a*. There are two readings of the sentence which satisfy this partial scope, those being:

$$most(z, sample(z), \\ every(x, a(y, co(y), rep.of(x, y)), see(x, z))) \\ \text{and} \\ most(z, sample(z), a(y, co(y), \\ every(x, rep.of(x, y), see(x, z))))).$$

These readings are precisely those for which the object of *see* outscopes its subject; the partial scope is captured by the pair:

$$\langle \{every, most, see\}, (most \succ every \succ see) \rangle \\ \langle \{every, a, of\}, (every \succ' of, a \succ' of) \rangle$$

where there is no additional information about the relative scope of *every* and *a*. However, the transitivity of  $\succ$  alone does not capture the fact that *most*  $\succ$  *a* follows from *most*  $\succ$  *every*.

We remedy this by defining a *domination* relation. In the current case, say that *every* dominates *a*, which means that *a* is nested within the QNP whose head quantifier is *every*. Then because quantifiers may not “intercalate” across NP boundaries, anything that outscopes *every* also outscopes anything that *every* dominates (here, *a*); if *most* outscopes one it must outscope both. We capture this behaviour by putting the sets into a tree structure, where each of the nodes is one of the strictly ordered sets representing the scopes around a relation. For any node, *N*, each of the daughter nodes has (exactly) one element in common with *N*, otherwise, any element appears only once in the structure. So, consider again the sentence *Every representative of a company saw most samples*. The scope information of the underspecified form is represented by the tree:

$$\langle \{every, most, see\}, (every \succ see, most \succ see) \rangle \\ | \\ \langle \{every, a, of\}, (every \succ' of, a \succ' of) \rangle$$

Now, say that an element *X* dominates another element *Y* (denoted as  $X \hookrightarrow Y$ ) if *X* and *Y* are (distinct) elements in a set at some node, and *X* is also in the parent node. Also,  $\hookrightarrow$  is transitive and irreflexive. So in the example given:

$$every \hookrightarrow a \text{ and } every \hookrightarrow of,$$

but  $every \not\hookrightarrow every$ .

We can now extend the definition of  $\succ$  by saying that:

if  $\langle P, \succ \rangle$  is a node in the tree, and  $x, y \in P$  and  $x \succ y$ , then  $x \succ y$  and  $x \succ z$  where  $z$  is any term that  $y$  dominates. Also,  $\succ$  is transitive and irreflexive.

This captures the scoping behaviour for nested quantifiers. So from the ambiguous representation of scopes:

$$\begin{array}{c} \langle \{every, most, see\}, (most \succ every \succ see) \rangle \\ | \\ \langle \{every, a, of\}, (every \succ' of, a \succ' of) \rangle \end{array}$$

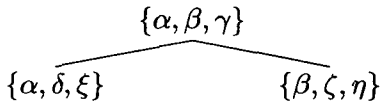
where  $most \succ every$  and  $every \hookrightarrow a$ , it is possible to infer correctly that  $most \succ a$ , whatever the relation is between  $every$  and  $a$ .

#### 4 Formal Definition of Scope Representations

We now provide a formal description of the structures described in section 3. The definition is divided into two parts. First a *scope structure* is defined, which is a tree structure whose nodes are sets under a strict order and describes the correct possible scopings of quantified arguments around their relations. Next, a *scope representation* is defined, which is the pair of a scope structure and an outscoping relation,  $\succ$ , which is defined over all the elements in the structure.

The analysis presented here differs from that of the previous section in that the nodes in the scope structures are sets under a strict *total* order, rather than under a partial order. The structures therefore represent unambiguous readings of the sentence. Underspecification will then be captured in the constraint language, rather than in the underlying structures, as discussed in section 5.

A scope structure is a finite tree, where each node of the tree is a finite, non-empty set of elements,  $P$ , taken from a set  $\mathcal{O} = \{\alpha, \beta, \gamma, \dots\}$  under a strict total order. For any node, each daughter node is also a strictly ordered set, such that each daughter set  $d_i$  has exactly one element in common with  $P$ , a different element for each of the  $d_i$ . An element can only appear once in the tree, unless it is the common node between a mother and a daughter. So:



is a correct scope structure, because no element appears twice except  $\alpha$  and  $\beta$ , which appear in mother/daughter pairs (the ordering relations have been omitted for clarity).

A scope structure is defined as a triple  $\langle P, \succ, \mathcal{D} \rangle$ , where  $P$  is a set of elements,  $\succ$  is a strict total order over  $P$  and  $\mathcal{D}$  is the set of daughters. We say that an element *occurs in* a scope structure if it is a member of the set at any node in the scope structure. If  $\mathcal{O}$  is a (countable) set of elements, then scope structures can be recursively defined as:

- If  $S = \langle P_S, \succ_S, \{\} \rangle$ , where  $P_S$  is a finite, non-empty subset of  $\mathcal{O}$  and  $\succ_S$  is a strict total order on  $P_S$ , then  $S$  is a scope structure, where:
  1. if  $x \in P_S$ , then  $x$  occurs in  $S$ ,
- If  $R$  and  $S$  are scope structures such that  $R = \langle P_R, \succ_R, \mathcal{D}_R \rangle$  and  $S = \langle P_S, \succ_S, \mathcal{D}_S \rangle$ , where no element occurs in both  $R$  and  $S$ , and there is some element  $\sigma$  such that  $\sigma \in P_R$ , then if  $T = \langle P_T, \succ_T, \mathcal{D}_T \rangle$ , where  $P_T = \{\sigma\} \cup P_S$ ,  $\mathcal{D}_T = \{\mathcal{D}_R\} \cup \mathcal{D}_S$  and  $\succ_T$  is a strict total order on  $P_T$  then  $T$  is a scope structure, where:
  1. If some element  $x$  occurs in either  $R$  or  $S$  then  $x$  occurs in  $T$
  2. If some element  $x$  occurs in  $R$  and  $x \neq \sigma$ , then  $\sigma$  dominates  $x$  in  $T$
  3. If  $x$  and  $y$  occur in  $R$  and  $x$  dominates  $y$  in  $R$  then  $x$  dominates  $y$  in  $T$
  4. If  $x$  and  $y$  occur in  $S$  and  $x$  dominates  $y$  in  $S$  then  $x$  dominates  $y$  in  $T$

If  $S$  is a scope structure, then a node in  $S$  is defined as:

- If  $S$  is a scope structure such that  $S = \langle P_S, \succ_S, \mathcal{D}_S \rangle$ , then:
  - $\langle P_S, \succ_S \rangle$  is a node in  $S$
  - if  $d_i \in \mathcal{D}_S$ , then any node in  $d_i$  is a node in  $S$ .

Having defined scope structures, we now define a *scope representation*, which is a pair  $\langle S, \succ_S \rangle$ , where  $S$  is a scope structure and  $\succ_S$  is a relation between pairs of elements which occur in  $S$ .  $\succ_S$  represents outscoping between any



elements which occur in  $S$  and wffs of the constraint language onto one of the pair of values  $\{t, f\}$ .  $I$  is a pair  $\langle \Phi, \mathcal{A} \rangle$ , where  $\Phi$  is a scope representation, such that  $\Phi = \langle S, \succ_S \rangle$ , and  $\mathcal{A}$  is a function mapping constants of the constraint language onto the set of elements which occur in  $S$ . The denotation of the constraints is then given by:

- $\llbracket X \rrbracket^I = \mathcal{A}(X)$  if  $X$  is a constant in the constraint language.
- $\llbracket X \circ Y \rrbracket^I = t$  if there is a node in  $S$ ,  $\langle P, \succ_P \rangle$ , such that  $\llbracket X \rrbracket^I \in P$  and  $\llbracket Y \rrbracket^I \in P$  and  $\llbracket X \rrbracket^I \neq \llbracket Y \rrbracket^I$ , otherwise  $\llbracket X \circ Y \rrbracket^I = f$ .
- $\llbracket X \hookrightarrow Y \rrbracket^I = t$  if  $\llbracket X \rrbracket^I$  dominates  $\llbracket Y \rrbracket^I$  in  $S$ , otherwise  $\llbracket X \hookrightarrow Y \rrbracket^I = f$ .
- $\llbracket X \triangleright Y \rrbracket^I = t$  if  $\llbracket X \rrbracket^I \succ_S \llbracket Y \rrbracket^I$ , otherwise  $\llbracket X \triangleright Y \rrbracket^I = f$ .
- $\llbracket \psi \wedge \phi \rrbracket^I = t$  if  $\llbracket \psi \rrbracket^I = t$  and  $\llbracket \phi \rrbracket^I = t$ , otherwise  $\llbracket \psi \wedge \phi \rrbracket^I = f$ .

**Satisfiability** A constraint set,  $\Delta$ , is satisfiable iff there is at least one  $I$  such that  $\llbracket \phi \rrbracket^I = t$  for all constraints  $\phi$  where  $\phi \in \Delta$ .

The satisfiability of a constraint set represents the existence of a reading of the sentence which respects the partial scoping.

## 6 Availability of Partial Scopes

We now turn to the question of determining whether a partial scoping is available. In section 3 it was stated that scope availability is accounted for by the relative scope of quantifiers around their predicates. It turns out (although we do not prove it here) that for any partial scoping, there is a necessary and sufficient set of scopings of quantifiers around their relations that gives the partial scoping. For example, we showed that for the sentence *Every representative of a company saw most samples*, the readings where *most* outscopes *a* are exactly those where the subject of *see* outscopes its object. Therefore, from the constraint  $most \triangleright a$ , it should be possible to infer  $most \triangleright every$ . The aim of the constraint solver is to determine what scopings of quantifiers about their relations are required to obtain the required partial scoping, and therefore to state whether the partial scope is available.

A set of rules is defined on the constraints, so that additional scope information may be inferred. The introduction of further scope constraints does not affect scope information already present (monotonicity). The rules are given in figure 3, where  $\Gamma$  represents any conjunction of literals and the associativity and commutativity of  $\wedge$  are assumed. The inference rules  $S1, S2$  and  $S3$  operate by recursively reducing the (arbitrary) outscoping constraint  $X \triangleright Z$  to  $X \triangleright Y \wedge Y \triangleright Y'$ , where  $Y$  and  $Y'$  represent arguments to a common relation, and  $Y'$  either dominates or is equal to  $Z$ . Repeated application of these constraints gives the set of scopes of quantifiers around their relations for the initial partial scoping. The rules *Trans* and *Dom* then generate the remaining possible scope constraints. If a scope is unavailable, then completing the transitive closure of  $\triangleright$  across the structure yields a constraint of the form  $X \triangleright X$ .

We then say that:

- A constraint set is in *normal form* iff applying the rules  $S1, S2, S3, Trans$  and *Dom* does not yield any new constraints.

If  $\Gamma$  is a constraint set in normal form then:

- $\Gamma$  represents an *available scoping* iff it does not contain a constraint of the form  $X \triangleright X$ .
- $\Gamma$  represents a *complete scoping* iff it represents an available scoping, and for every constraint of the form  $X \circ Y$  there is either a constraint  $X \triangleright Y$  or a constraint  $Y \triangleright X$ .

The condition for a scoping to be available follows from the irreflexivity of  $\succ$ . The condition for a scoping to be complete states that if two elements are arguments to a relation, or are a relation and one of its arguments, then they must have scope relative to each other. This corresponds to considering sets under a total order, rather than under a partial order.

**Complexity Issues** Let  $\Gamma$  be a constraint representing an available scoping of a sentence, and let  $X \triangleright Y$  be a constraint representing a partial scope between two terms in that sentence. Then the worst case of applying the inference rules to  $\Gamma \wedge X \triangleright Y$  to saturation turns out to be equivalent to completing the transitive closure of  $\triangleright$ , which is known to be soluble in better than  $O(n^3)$  time (Cormen et al., 1990), where  $n$  is the number of elements in the structure.

$$\begin{aligned}
S1: & \Gamma \wedge X \circ Y \wedge X \leftrightarrow X' \wedge X' \triangleright Y \vdash X \triangleright Y \wedge X' \triangleright X \\
S2: & \Gamma \wedge X \circ Y \wedge Y \leftrightarrow Y' \wedge X \triangleright Y' \vdash X \triangleright Y \\
S3: & \Gamma \wedge X \circ Y \wedge X \leftrightarrow X' \wedge Y \leftrightarrow Y' \wedge X' \triangleright Y' \vdash X' \triangleright X \wedge X \triangleright Y \\
\text{Trans:} & \Gamma \wedge X \triangleright Y \wedge Y \triangleright Z \vdash X \triangleright Z \\
\text{Dom:} & \Gamma \wedge X \circ Y \wedge X \triangleright Y \wedge Y \leftrightarrow Z \vdash X \triangleright Z \\
& \text{where } \Gamma \text{ is any conjunction of literals.}
\end{aligned}$$

Figure 3: Rules of inference

Application of rules  $S1$ ,  $S2$  and  $S3$  to completion can be completed in linear time; if  $X \triangleright Y$  is a constraint between two arbitrary quantifiers  $X$  and  $Y$  where  $X \not\circ Y$ , then exactly one of the rules  $S1$ ,  $S2$  or  $S3$  applies (lack of space prevents us proving this here). If  $X \circ Y$ , then none of these three rules applies. Application of  $S1$ ,  $S2$  or  $S3$  adds at most two new constraints, of which at most one is a scope constraint  $X \triangleright Y'$  where  $X \not\circ Y'$ . At most  $n - 1$  such constraints are generated.

Application of the rules  $S1$ ,  $S2$  and  $S3$  reduces an arbitrary partial scope into relative scopes of arguments around their relations. If a scoping is unavailable, this is represented by the irreflexivity of  $\triangleright$  being violated. Testing for this requires that the transitive closure of  $\triangleright$  be completed; this is known to be soluble in better than cubic time. We conclude that testing for the availability of a partial scope in this framework can be achieved in better than cubic time in the worst case.

## 7 Conclusion and Comments

A desirable property for an underspecified representation of quantifier scope ambiguity is that there should be a computationally efficient test for whether a partial scope is available or not. We have shown that accepting a theory of availability which states that scope availability is determined by the function-argument structure of a sentence allows the development of a test for availability which is polynomial in the number of quantifiers and relations in a sentence, while theories of availability based upon the logical well-formedness of meaning representations has been shown to be NP-hard.

**Acknowledgements** The authors would like to thank Alan Frisch, Mark Steedman and three anonymous reviewers for useful comments. The first author is funded by an EPSRC grant.

## References

- H. Alshawi and R. Crouch. 1992. Monotonic Semantic Interpretation. In *Proceedings of the 30th Annual Meeting of the ACL*, pages 32–39, Newark, Delaware.
- R. Cooper. 1983. *Quantification and Syntactic Theory*. Reidel.
- T. Cormen, C. Leiserson, and R. Rivest. 1990. *Introduction to Algorithms*. The MIT Press, Cambridge, Massachusetts.
- M. Egg, J. Niehren, P. Ruhrberg, and F. Xu. 1998. Constraints over lambda-structures in semantic underspecification. In *Proceedings of the 17th International Conference on Computational Linguistics and 36th Annual Meeting of the ACL*, Montreal, Canada.
- J. Hobbs and S. Shieber. 1987. An algorithm for generating quantifier scopings. *Computational Linguistics*, 13.
- W. Keller. 1986. Nested Cooper storage: The proper treatment of quantification in ordinary noun phrases. In U. Reyle and C. Rohrer, editors, *Natural Language Parsing and Linguistic Theory*, Studies in Linguistics and Philosophy, pages 432–437. Reidel.
- A. Koller, J. Niehren, and R. Treinen. 1998. Dominance constraints: Algorithms and complexity. In *Third International Conference on Logical Aspects of Computational Linguistics (LACL '98)*, Grenoble, France.
- J.C. Park. 1995. Quantifier scope and constituency. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 205–212. Cambridge, MA.
- U. Reyle. 1995. On reasoning with ambiguities. In *Proceedings of the EACL*, Dublin.
- A. Willis and S. Manandhar. 1999. The availability of partial scopings in an underspecified semantic representation. In *3rd International Workshop on Computational Semantics*, Tilburg, the Netherlands, January.