Non-Redundant Scope Disambiguation in Underspecified Semantics

Rui Pedro Chaves
Centro de Linguística da Universidade de Lisboa (CLUL), Av. Gama Pinto, n°2, Lisboa, Portugal
rui.chaves@clul.ul.pt

Abstract. This paper presents a strategy that aims to efficiently avoid the generation of logically equivalent formulas that arise in scope processing applications. Constraint based semantic underspecification formalisms may be extended to include an additional scoping restriction that constrains the set of possible disambiguations, straightforwardly avoiding the generation of redundant quantifier scopings. Such a restriction, in principle valid to any logic, is formalized within Hole Semantics (Bos 1996), a general semantic underspecification framework.

1 Introduction

Semantic Underspecification frameworks (such as QLF (Alshawi and Crouch 1992), UDRSs (Reyle 1993), UMRS (Egg and Lebeth 1995), Hole Semantics (Bos 1996) and CLLS (Egg et al. 1998)) are able to cope efficiently with the combinatorial explosion of highly ambiguous Natural Language phenomena such as scope ambiguity.

Classic approaches to scope processing (such as Cooper (1983), Hobbs and Shieber (1987) and Keller (1988)) are inefficient in the sense that an exponential number of formulas is always generated (worse case entails generating $n!$ formulas, for n quantifiers). In underspecified semantics, the set of possible readings is described via a unique, compact partial representation which may be reduced in a straightforward fashion, simply by the incremental specification of additional constraints (e.g. from prosody, discourse context or world knowledge). Furthermore, underspecification is able to deal with other scope bearing constituents other than quantification.

However, the existing methods of scope processing typically overgenerate in the sense that some or even all generated formulas may be logically equivalent. We present a new scoping restriction capable of reducing the set of scopal disambiguations in underspecified semantics, avoiding the generation of logical redundancies that result from quantifier scope (a subset of formulas with the same *prenex normal form*) and therefore greatly reducing

Proceedings of the Eighth ESSLLI Student Session
Balder ten Cate (editor)

the need for theorem provers. Section 2 briefly reviews the existing methods for the elimination of redundant formulas. Section 3 formalizes a scope disambiguation restriction within Hole Semantics and applies it to Discourse Representation Theory (Kamp and Reyle 1993) formulas. Finally, practical implementation results and some extensions are discussed.

2 Logical Overgeneration: Redundant Scopings

It is well known that the relative orderings of identical first order quantifiers do not result in distinct truth conditions. For instance, sentence (1) only has a single reading, yet two equivalent formulas are typically generated:

(1) Every boy saw every girl.
(1a) $(\forall x)(boy(x) \rightarrow (\forall y)(girl(y) \rightarrow saw(x,y)))$
(1b) $(\forall y)(girl(y) \rightarrow (\forall x)(boy(x) \rightarrow saw(x,y)))$

A more complex case is visible in sentence (2), which typically receives up to 60 equivalent formulas[1] but only has a single reading:

(2) A boy that sings in a choir gave a flower with a velvet lace to a girl.

The formulas generated for each of the examples above have the same *prenex normal form*[2] and hence are equivalent. We are presently concerned with equivalences that result from the Laws of Quantifier Movement (pulling nested quantifiers out of formulas: $(\varphi \rightarrow (\forall x)\psi(x)) \Leftrightarrow (\forall x)(\psi \rightarrow \psi(x))$ and $(\varphi \rightarrow (\exists x)\psi(x)) \Leftrightarrow (\exists x)(\psi \rightarrow \psi(x))$ where x is not free in φ and from the Laws of Quantifier Independence (interchanging quantifiers: $(\forall x)(\forall y)\varphi(x,y) \Leftrightarrow (\forall y)(\forall x)\varphi(x,y)$ and $(\exists x)(\exists y)\varphi(x,y) \Leftrightarrow (\exists y)(\exists x)\varphi(x,y)$).

Much more subtle cases involve (possibly many) different sets of logically redundant scopings, as sentence (3) below illustrates. Here, a distinct scopal operator (negation) induces partial non-redundancy: 6 possible permutations but only 4 logically distinct readings.

(3) A cat doesn't like a dog.
(3a) $(\exists y)(dog(y) \wedge (\exists x)(cat(x) \wedge \neg like(x,y)))$
(3b) $(\exists y)(dog(y) \wedge \neg(\exists x)(cat(x) \wedge like(x,y)))$
(3c) $\neg(\exists y)(dog(y) \wedge (\exists x)(cat(x) \wedge like(x,y)))$
(3d) $(\exists x)(cat(x) \wedge (\exists y)(dog(y) \wedge \neg like(x,y)))$
(3e) $(\exists x)(cat(x) \wedge \neg(\exists y)(dog(y) \wedge like(x,y)))$
(3f) $\neg(\exists x)(cat(x) \wedge (\exists y)(dog(y) \wedge like(x,y)))$

[1] The total number of scopings is not factorial (i.e. $5! = 120$) because nested NPs cannot have arbitrary semantic scope beyond their syntactical local domain, e.g. '*a choir*' may not simultaneously outscope '*a velvet lace*' and be outscoped by '*a flower*'.

[2] $(\mathrm{Q}_0\ \mathrm{x}_0)\ldots(\mathrm{Q}_{n-1}\ \mathrm{x}_{n-1})\psi$ where Q_i $(0 < i < n)$ is a quantifier and ψ is quantifier-free.

Clearly, formula (3a) is equivalent to (3d) and formula (3c) is equivalent to (3f). One solution to this problem is to use a theorem prover with a search time limit to help decide the equivalence of each pair of formulas generated: formulas P and Q are equivalent *iff* (P⇔Q) is provable. For n generated formulas a theorem prover requires $n(n-1)$ proofs in the worse case (i.e. equivalent formulas do not exist), for pairwise choices of P and Q.

Vestre (1991) presents a method that avoids quantifier scope redundancy by limiting the selection of determiners in an enumerative algorithm for scope processing that exhaustively generates and evaluates all possible scopings for a given quantifier. In contrast, we present a method that does not search an exponential number of scopings. Rather, the proposed strategy strictly generates irredundant quantifier scopings by keeping track of the scope-bearing constituent undergoing disambiguation and prohibiting certain disambiguation patterns. Because this method is formalized within a semantic underspecification framework it is able to consider partially disambiguated structures and the interaction of other scope-bearing structures besides quantification, such as negation, modality and indirect discourse.

Gabsdil and Striegnitz (1999) proposes and implements a general method that orders all the formulas outputted by a scoping algorithm into a graph structure in which logically equivalent readings are collapsed into a single structure. This strategy requires the exhaustive enumeration of possible formulas, the very problem that underspecification originally aimed to solve. The next section formalizes a method, independent from the logic of choice, that aborts (sets of) disambiguations that describe redundant scopings.

3 Non-Redundant Scopings in Hole Semantics

Hole Semantics (Bos 1996) is a general underspecification framework where there is a clear distinction between the underspecification metalanguage and the object language. In this framework, scopal ambiguity is represented via partial subordination constraints in an upper-semilattice. What follows is a brief excursion to Hole Semantics.

An Underspecified Representation (UR) is a triple ⟨H,L,C⟩ where H is a set of metavariables (*holes*) over formulas; L is a set of labelled formulas; and C is a set of subordination constraints expressed via a relation "$\leq$" that establishes a partial order (i.e. is reflexive, transitive and antisymmetric) over labels and metavariables. This partial order is also an upper-semilattice given that a special element, the *supremum*, subsumes any pair of elements in the structure. An admissible disambiguation (henceforth a *plugging*) is a bijection from holes to labelled formulas that does not violate the set of constraints, operating recursively from a special *top* hole h_0. When a hole is plugged (i.e. $P(h)=l$), the hole variable is substituted by the formula identified by the correspondent label (e.g. "$l{:}\varphi$"). Bos (1996) shows that both

Dynamic Predicate Logic and Discourse Representation Structures (DRSs (Kamp and Reyle 1993)) can be used as object language in Hole Semantics.

We now formalize a general method within Hole Semantics, in principle applicable to any logic and originally suggested in Chaves (2002a), that consists in a restriction that constrains the set of *possible pluggings* (i.e. disambiguations that do not violate any subordination constraints). In essence, pluggings shall be associated to a special unrestricted scope operator, initialised as "$\top$", and as the disambiguation proceeds this operator may be *updated*, *percolated* or forced to *abort* ("$\perp$") the plugging (in the latter case, plugging a given hole with a given formula would result in a non-empty set of formulas with logically equivalent scopings, if disambiguation is completed).

It must be expressly noted that we assume the numerical indexes associated to each label are unique and lexically attributed according to *surface order*. The *Distinct Scope* restriction will be endowed with a 'short-term memory' of the disambiguation process and shall impose decreasing order on the label indexes between formulas that may induce logical redundancy. More generally, scope disambiguation in constraint based underspecification formalisms can be further restricted in order to efficiently avoid the symmetric scoping counterpart between specific operators capable of inducing logical redundancy.

3.1 Metalanguage Definitions

Firstly, we shall define the set of metalanguage formula schemata that can result in redundant disambiguations. Secondly, we provide a basis for detecting formulas with identical outermost scopal operators and finally, we formalize a general *Distinct Scope* function and the restrictions therein.

Definition 1: Scopal Schemata
Let S be the set of formula schemata {Op(ψ): Op(ψ) $\in$ U}, defined in a given metalanguage U, under the scope of an operator Op that can potentially induce redundant disambiguations. For instance, in an underspecified account of predicate logic such as Bos (1996), one would have S = $\{(\exists x)(k), (\forall y)(k)\}$; where k is either a metavariable or a metalanguage formula.

Definition 2: Scopal Operator Equivalence
Formulas φ and ψ have an *equivalent syntactical scope* ($\varphi \equiv \psi$) *iff* φ and ψ are of the form $\mathrm{Op}_i(\varphi')$ and $\mathrm{Op}_j(\psi')$ respectively, where Op_i and Op_j are identical scope-bearing operators. E.g. : '$(\exists x)(\alpha)$' $\equiv$ '$(\exists y)(\beta)$' $\not\equiv$ '$(\forall z)(\tau)$'.

Definition 3: Distinct Scope Restriction
Let D be the function which is defined as follows D: $\Gamma \times \Gamma \rightarrow \Gamma$, where Γ is a set of ϕ::= $(l : \varphi)|\top|\perp$. This function establishes a mapping between a ordered pair of *main scope operators* (either some labelled metaformula 'l:φ',

or the special symbols "⊤" (*verum*) and "⊥" (*falsum*)) and a new *main scope operator*. For n equivalent readings, the ordering constraints enforced by D are able to interrupt pluggings linearly, *at most n* times (i.e. depending on the plugging strategy). Below, φ and ψ are metalanguage formulas:

$$D(\top, (l_i : \varphi)) = (l_i : \varphi) \qquad (initial\ scope\ domain\ update)$$

$$D((l_i : \varphi), (l_j : \psi)) = \begin{cases} (l_i : \varphi) & if \ \ \psi \notin S & (percolate) \\ (l_j : \psi) & if \ \ \psi \in S \ \wedge \ \varphi \not\equiv \psi & (update) \\ (l_j : \psi) & if \ \ \varphi, \psi \in S \ \wedge \ \varphi \equiv \psi \ \wedge \ i > j & (update) \\ \bot & otherwise & (abort) \end{cases}$$

The crucial step takes place when both syntactically equivalent formulas are in set S and have *decreasing* labelling indexes. Function D aborts this specific plug, not allowing orderings where the numerical indexes increase, thus eliminating the full permutation of logically identical quantificational operators. For instance "every$_1$ student$_2$ read$_3$ every$_4$ poem$_5$ to$_6$ every$_7$ girl$_8$" would only have the disambiguation with the scoping order: 7>4>1.

Next, the main scope operator slot and D must be embedded into the plugging procedure itself. To illustrate this we shall adopt *unplugged* DRSs (Bos 1996) as a description language, extended in order to explicitly consider duplex conditions and a basic account of indirect discourse.

3.2 Extended Plugging Procedure for DRTU

Typical disambiguation algorithms for DRT formulas will also be able to overgenerate by building identical DRSs, given different scoping choices:

Syntax of DRTU formulas (adapted from Bos (1996))

1. If h_i and h_j are holes and p is a propositional discourse marker then $\langle\{\},\{h_i \Rightarrow h_j\}\rangle$, $\langle\{\},\{\neg h_i\}\rangle$, $\langle\{\},\{h_i \vee h_j\}\rangle$, and $\langle\{p\},\{p : h\}\rangle$ are DRTU formulas.
2. If h is a hole, Q is a generalized quantifier and $k_1 \dots k_n$ are holes or DRTU formulas then $\otimes\{k_1, \dots, k_n\}$ and $\langle\{\},\{Q(k,h)\}\rangle$ are DRTU formulas.
3. If $x_1 \dots x_n$ are discourse markers and P is a symbol for an n-place predicate, then $\langle\{x\},\{\}\rangle$ and $\langle\{\},\{P(x_1, \dots, x_n)\}\rangle$ are DRTU formulas.

Nothing else is a DRTU formula.

The *merge* operator "⊗" denotes the union of DRSs (Universes and Conditions), e.g. $\otimes\{\langle U_1, C_1\rangle, \dots, \langle U_n, C_n\rangle\} = \langle U_1 \cup \dots \cup U_n,\ C_1 \cup \dots \cup C_n\rangle$. By definition, the label of the formula subordinates every *hole variable* introduced by that same formula. In these terms, a single underspecified representation is assigned to an ambiguous sentence such as (4):

(4) A boy didn't read a book.

$$\text{(5)} \quad \text{UR} = \left\langle \left\{ \begin{array}{c} h_0 \\ h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \end{array} \right\}, \left\{ \begin{array}{l} l_1 : \otimes\{\langle\{x\},\{\}\rangle, h_1, h_2\} \\ l_2 : \langle\{\},\{boy(x)\}\rangle \\ l_3 : \langle\{\},\{\neg h_3\}\rangle \\ l_4 : \langle\{\},\{read(x,y)\}\rangle \\ l_5 : \otimes\{\langle\{y\},\{\}\rangle, h_4, h_5\} \\ l_6 : \langle\{\},\{book(y)\}\rangle \end{array} \right\}, \left\{ \begin{array}{c} l_1 \le h_0 \\ l_3 \le h_0 \\ l_5 \le h_0 \\ l_2 \le h_1 \\ h_1 \le l_2 \\ l_4 \le h_2 \\ l_4 \le h_3 \\ l_6 \le h_4 \\ h_4 \le l_6 \\ l_4 \le h_5 \end{array} \right\} \right\rangle$$

Graphically, subordination constraints are represented by arrows:

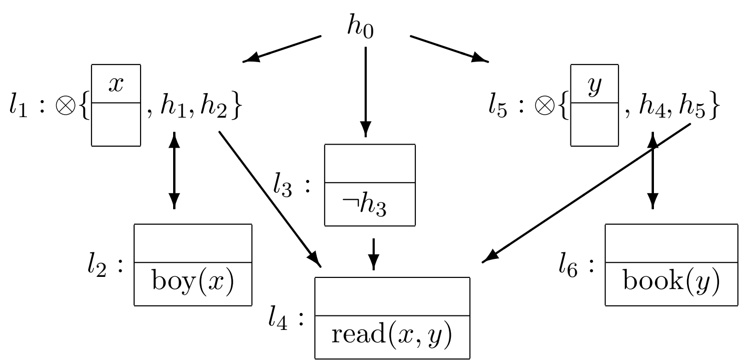

Figure 1: Graphical representation of UR (5).

Note that the scope between the metavariables in the subject NP indefinite (h_2), negation (h_3) and complement NP indefinite (h_5) is unspecified, and therefore a total of 6 pluggings are possible:

$$\begin{array}{l} \text{P}_1 : \{h_0 = l_1, h_1 = l_2, h_2 = l_5, h_4 = l_6, h_5 = l_3, h_3 = l_4\} \\ \text{P}_2 : \{h_0 = l_1, h_1 = l_2, h_2 = l_3, h_3 = l_5, h_4 = l_6, h_5 = l_4\} \\ \text{P}_3 : \{h_0 = l_3, h_3 = l_1, h_1 = l_2, h_2 = l_5, h_4 = l_6, h_5 = l_4\} \\ \text{P}_4 : \{h_0 = l_3, h_3 = l_5, h_4 = l_6, h_5 = l_1, h_1 = l_2, h_2 = l_4\} \\ \text{P}_5 : \{h_0 = l_5, h_4 = l_6, h_5 = l_3, h_3 = l_1, h_1 = l_2, h_2 = l_4\} \\ \text{P}_6 : \{h_0 = l_5, h_4 = l_6, h_5 = l_1, h_1 = l_2, h_2 = l_3, h_3 = l_4\} \end{array}$$

Note also that P_1 and P_6 are redundant (i.e yield the same DRS), and so are P_3 and P_4. In order to avoid this overgeneration, the labels of the formulas lexically introduced will be used to limit the range of possible pluggings as outlined by function D. An extended plugging algorithm that incorporates this function is defined below as a partial function $Plug$, restricting the set of possible disambiguations. Firstly, assume $\text{S} = \{\otimes\{k, h, h\}, \langle\{\}, \{Q_x(k, h)\}\rangle\}$ for the schemata of DRTU formulas, where Q is a given generalized quantifier such as '*every*' or '*few*' (but not like '*most*' or '*many*', for the consecutive interchanging of such quantifiers does not always preserve the same truth conditions; see McCawley (1981: 53-54) for a brief discussion).

Definition 4: Extended Plugging Procedure
Plugging corresponds to the function $Plug : (H \cup L) \times \Gamma \times$ SR $\rightarrow \Sigma$, where H is the set of holes and L the set of labelled metalanguage formulas, Γ is the set of *main scope operators* ϕ as defined before, SR is the set of *solved* Underspecified Representations $\langle H, L, C\rangle$, and finally, Σ is the set of disambiguated object-language formulas.

The set of constraints C in a solved Underspecified Representation explicitly describes a solution to the original underspecified structure. In other words, the constraints in C describe a tree of the formulas in L (i.e. subordinations $l \leq h$ and $l \leq h'$ where $h \neq h'$ do not exist). Althaus et al. (2003) presents an efficient method for enumerating the described solutions as *forests* in constraint graphs (a general framework for the partial description of trees) and these results can be used in several other underspecification formalisms, including Hole Semantics (Koller et al. 2003).

The initial call to this function is $Plug(h_0, \top, \langle H, L, C\rangle)$ where H is a set of holes, L is a set of labelled unplugged formulas and C a set of solved subordination constraints. The return shall be a disambiguated formula in the object-language. The crucial step of avoiding spurious scopings takes place when a consistent plug $\mathrm{P}(h) = l$ occurs and function D is applied:

$$Plug(h, \phi, \langle\, H \cup \{h\}\,,\ L \cup \{l : \psi\}\,,\ C \cup \{(l \leq h)\}\,\rangle) = \\ \quad Plug(\psi, D(\phi,\ l : \psi), \langle H, L, C\rangle)$$

Above, hole h is identified with a given outscoped formula ψ. Next, a recursive call attempts to plug formula ψ *iff* $D(\phi, l : \psi) \neq \bot$. More explicitly, $Plug(\psi,\ \bot,\ \langle H, L, C\rangle)$ always fails. Note that D is compatible with more straightforward, though less efficient, disambiguation strategies: in Blackburn and Bos (1999) the original set C of the UR is updated after each plug $\mathrm{P}(h) = l$ (i.e. every occurrence of h in C is replaced by l) and is checked for consistency. Similarly, D applies to the next plug and the remaining cases are identical to the presented below (Chaves 2002b: 325).

In duplex conditions the *restrictor* hole remains unconstrained ("$\top$") while the *scope* hole does not, because nested generalized quantifiers within a restrictor yield distinct readings, as illustrated below in sentence (6):

$$Plug(\langle\{\}, \{Q_x(k, h)\}\rangle, \phi, \langle H, L, C\rangle) = \\ \quad \langle\{\}, \{\, Q_x(\, Plug(k, \top, \langle H, L, C\rangle),\ Plug(h, \phi, \langle H, L, C\rangle)\,\}\rangle$$

(6) Every representative of every company protested.

At least two readings are available for (6): companies with possibly different representatives protested; representatives that simultaneously represent all companies protested. Note that if D imposed an *increasing* order to the numerical indexes, nested quantifiers would not get wide scope readings over the main quantifier (perhaps indexes might be able to reflect scopal preferences via an underspecified partial order). Conditionals are similar:

$$Plug(\langle\{\},\{\, h_i \Rightarrow h_j \,\}\rangle, \phi, \langle H, L, C\rangle) = \\ \langle\{\},\{\, Plug(h_i, \top, \langle H, L, C\rangle) \;\Rightarrow Plug(h_j, \phi, \langle H, L, C\rangle) \,\}\rangle$$

$$Plug(\otimes\{k_1, \ldots, k_n\}, \phi, \langle H, L, C\rangle) = \\ \otimes\{Plug(k_1, \phi, \langle H, L, C\rangle), \ldots, Plug(k_n, \phi, \langle H, L, C\rangle)\}$$

Indirect discourse, disjunction, negation (as well as modal operators) and n-place predicates do not induce *scopal* logical redundancy, and have unrestricted ("$\top$") pluggings (where α is a propositional discourse referent):

$$Plug(\langle\{\alpha\},\{\alpha : h\}\rangle, \phi, \langle H, L, C\rangle) = \langle\{\alpha\},\{\alpha : (Plug(h, \top, \langle H, L, C\rangle))\}\rangle$$

$$Plug(\langle\{\},\{\, h_i \vee h_j\}\rangle\,, \phi, \langle H, L, C\rangle) = \\ \langle\{\},\{Plug(h_i, \top, \langle H, L, C\rangle) \vee Plug(h_j, \top, \langle H, L, C\rangle)\}\rangle$$

$$Plug(\langle\{\},\{\neg h\}\rangle, \phi, \langle H, L, C\rangle) = \langle\{\},\{\neg\, Plug(h, \top, \langle H, L, C\rangle)\,\}\rangle$$

$$Plug(\langle\{\},\{R(x_1, \ldots, x_n)\}\rangle, \phi, \langle H, L, C\rangle) = \langle\{\},\{R(x_1, \ldots, x_n)\}\rangle$$

Take for instance sentence (7) and the corresponding UR depicted in (8) below, which has 6 distinct readings but a total of 18 possible pluggings:

(7) A girl that a teacher mentioned didn't read a book.

(8)
$$\left\langle \left\{ \begin{array}{c} h_0 \\ h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{array} \right\}, \left\{ \begin{array}{l} l_1 : \otimes\{\langle\{x\},\{\}\rangle, h_1, h_2\} \\ l_2 : \langle\{\},\{girl(x)\}\rangle \\ l_3 : \otimes\{h_3, h_4\} \\ l_4 : \otimes\{\langle\{y\},\{\}\rangle, h_5, h_6\} \\ l_5 : \langle\{\},\{teacher(y)\}\rangle \\ l_6 : \langle\{\},\{mentioned(y,x)\}\rangle \\ l_7 : \langle\{\},\{\neg h_7\}\rangle \\ l_8 : \langle\{\},\{read(x,z)\}\rangle \\ l_9 : \otimes\{\langle\{z\},\{\}\rangle, h_8, h_9\} \\ l_{10} : \langle\{\},\{book(z)\}\rangle \end{array} \right\}, \left\{ \begin{array}{c} l_1 \leq h_0 \\ l_4 \leq h_0 \\ l_7 \leq h_0 \\ l_9 \leq h_0 \\ l_3 \leq h_1 \\ h_1 \leq l_3 \\ l_2 \leq h_3 \\ h_3 \leq l_2 \\ l_6 \leq h_4 \\ l_8 \leq h_2 \\ l_5 \leq h_5 \\ h_5 \leq l_5 \\ l_6 \leq h_6 \\ l_{10} \leq h_8 \\ h_8 \leq l_{10} \\ l_8 \leq h_9 \end{array} \right\} \right\rangle$$

The 6 available readings are obtained through the following pluggings:

$\{h_0{=}l_4, h_5 = l_5, h_6 = l_1, h_1 = l_3, h_3 = l_2, h_4 = l_6, h_2 = l_7, h_7 = l_9, h_8 = l_{10}, h_9 = l_8\}$
(8a) $\langle\{y, x\}, \{\mathit{teacher}(y), \mathit{girl}(x), \mathit{mentioned}(y, x), \neg\langle\{z\}, \{\mathit{book}(z), \mathit{read}(x, z)\}\rangle\}\rangle$

$\{h_0{=}l_4, h_5 = l_5, h_6 = l_7, h_7 = l_9, h_8 = l_{10}, h_9 = l_1, h_1 = l_3, h_3 = l_2, h_4 = l_6, h_2 = l_8\}$
(8b) $\langle\{y\}, \{\mathit{teacher}(y), \neg\langle\{z, x\}, \{\mathit{book}(z), \mathit{girl}(x), \mathit{mentioned}(y, x), \mathit{read}(x, z)\}\rangle\}\rangle$

$\{h_0{=}l_9, h_8 = l_{10}, h_9 = l_4, h_5 = l_5, h_6 = l_1, h_1 = l_3, h_3 = l_2, h_4 = l_6, h_2 = l_7, h_7 = l_8\}$
(8c) $\langle\{z, y, x\}, \{\mathit{book}(z), \mathit{teacher}(y), \mathit{girl}(x), \mathit{mentioned}(y, x), \neg\langle\{\}, \{\mathit{read}(x, z)\}\rangle\}\rangle$

$\{h_0{=}l_9, h_8 = l_{10}, h_9 = l_4, h_5 = l_5, h_6 = l_7, h_7 = l_1, h_3 = l_3, h_3 = l_2, h_4 = l_6, h_2 = l_8\}$
(8d) $\langle\{z, y\}, \{\mathit{book}(z), \mathit{teacher}(y), \neg\langle\{x\}, \{\mathit{girl}(x), \mathit{mentioned}(y, x), \mathit{read}(x, z)\}\rangle\}\rangle$

$\{h_0{=}l_9, h_8 = l_{10}, h_9 = l_7, h_7 = l_4, h_5 = l_5, h_6 = l_1, h_1 = l_3, h_3 = l_2, h_4 = l_6, h_2 = l_8\}$
(8e) $\langle\{z\}, \{\mathit{book}(z), \neg\langle\{y, x\}, \{\mathit{teacher}(y), \mathit{girl}(x), \mathit{mentioned}(y, x), \mathit{read}(x, z)\}\rangle\}\rangle$

$\{h_0{=}l_7, h_7 = l_9, h_8 = l_{10}, h_9 = l_4, h_5 = l_5, h_6 = l_1, h_1 = l_3, h_3 = l_2, h_4 = l_6, h_2 = l_8\}$
(8f) $\langle\{\}, \{\neg\langle\{z, y, x\}, \{\mathit{book}(z), \mathit{teacher}(y), \mathit{girl}(x), \mathit{mentioned}(y, x), \mathit{read}(x, z)\}\rangle\}\rangle$

The remaining 12 pluggings are aborted by the ordering constraints in D. For instance, plug $\mathrm{P}(h_6) = l_9$ visible below (where $\otimes\{\langle\{z\}, \{\}\rangle, h_8, h_9\}$ is identified with h_6 in $\otimes\{\langle\{y\}, \{\}\rangle, h_5, h_6\}$) is unsuccessful because $4 \not\succ 9$:

$\{h_0 = l_4,\ h_5 = l_5,\ h_6 = l_9\}$ (*plugging aborted*)

$D(\top,\ \ l_4\colon \otimes\{\langle\{y\}, \{\}\rangle, h_5, h_6\}) =\ l_4\colon \otimes\{\langle\{y\}, \{\}\rangle, h_5, h_6\}$
$D(l_4{:}\otimes\{\langle\{y\}, \{\}\rangle, h_5, h_6\},\ \ l_5\colon \langle\{\}, \{\mathit{teacher}(y)\}\rangle) =\ l_4\colon \otimes\{\langle\{y\}, \{\}\rangle, h_5, h_6\}$
$D(l_4{:}\otimes\{\langle\{y\}, \{\}\rangle, h_5, h_6\},\ \ l_9\colon \otimes\{\langle\{z\}, \{\}\rangle, h_8, h_9\}) =\ \bot$

If continued, this plugging would be equivalent to either (8c) or (8d) above:

$$\{h_0{=}l_4, h_5{=}l_5, \mathbf{h}_6 = \mathbf{l}_9,\ h_8{=}l_{10}, \begin{array}{l} \nearrow h_9{=}\ l_1, h_1{=}\ l_3, h_3{=}\ l_2, h_4 = l_6, h_2 = l_7, h_7 = l_8\} \\ \searrow h_9{=}\ l_7, h_7{=}\ l_1, h_1{=}\ l_3, h_3 = l_2, h_4 = l_6, h_2 = l_8\} \end{array}$$

Similarly for (8a) and (8c), $\{h_0 = l_1,\ h_1 = l_3,\ h_3 = l_2,\ h_4 = l_4\}$ is aborted:

$D(\top,\ \ l_1\colon \otimes\{\langle\{x\}, \{\}\rangle, h_1, h_2\}) =\ l_1\colon \otimes\{\langle\{x\}, \{\}\rangle, h_1, h_2\}$
$D(l_1\colon \otimes\{\langle\{x\}, \{\}\rangle, h_1, h_2\},\ \ l_3\colon \otimes\{h_3, h_4\}) =\ l_1\colon \otimes\{\langle\{x\}, \{\}\rangle, h_1, h_2\}$
$D(l_1\colon \otimes\{\langle\{x\}, \{\}\rangle, h_1, h_2\},\ \ l_2\colon \langle\{\}, \{\mathit{girl}(x)\}\rangle) =\ l_1\colon \otimes\{\langle\{x\}, \{\}\rangle, h_1, h_2\}$
$D(l_1\colon \otimes\{\langle\{x\}, \{\}\rangle, h_1, h_2\},\ \ l_4\colon \otimes\{\langle\{y\}, \{\}\rangle, h_5, h_6\}) =\ \bot.$

$$\{h_0{=}\ l_1, h_1{=}\ l_3, h_3{=}\ l_2, \mathbf{h}_4{=}\mathbf{l}_4,\ h_5{=}\ l_5, h_6{=}\ l_6, \begin{array}{l} \nearrow h_2{=}\ l_7, h_7{=}\ l_9, h_8{=}\ l_{10}, h_9{=}\ l_8\} \\ \searrow h_2{=}\ l_9, h_8{=}\ l_{10}, h_9{=}\ l_7, h_7{=}\ l_8\} \end{array}$$

In sum, the UR in (8) has a total of 18 disambiguations out of which 12 correspond to equivalent readings. The *Distinct Scope* restriction licenses only a subset of 6 pluggings, all corresponding to logically distinct scopings.

A Prolog implementation indicates that D speeds up the disambiguation of highly redundant URs by 10% to 25% (e.g. 1048 equivalent pluggings in 7.2 secs. *vs* a unique reading in 5.8 secs. total on a PIII 866MHz 256Mgs) and that no noticeable computational delay is induced by irredundant URs (CGI at `http://www.clul.ul.pt/clg/scope.html`). Although Chaves (2002b) uses a less efficient plugging method overall, maximum speed up is of 91%.

3.3 Double Negation

The wide/narrow scopings of the indefinite "a woman" in (9a) and (9b) respectively, are equivalent because of double negation (Corblin (1995, 1996)):

(9) It is not true that John did not see a woman.
(9a) $\langle\{x, y\}, \{John(x), woman(y), \neg\langle\{\}, \{\neg\langle\{\}, \{see(x, y)\}\rangle\}\rangle\}\rangle$
(9b) $\langle\{x\}, \{John(x), \neg\langle\{\}, \{\neg\langle\{y\}, \{woman(y), see(x, y)\}\rangle\}\rangle\}\rangle$

This is also true of two of the readings of (10)(Gabsdil and Striegnitz (1999)):

(10) No criminal does not love a woman.
(10a) $\langle\{y\}, \{woman(y), \neg\langle\{\}, \{\neg\langle\{x\}, \{criminal(x), love(x, y)\}\rangle\}\rangle\}\rangle$
(10b) $\langle\{\}, \{\neg\langle\{\}, \{\neg\langle\{x, y\}, \{criminal(x), woman(y), love(x, y)\}\rangle\}\rangle\}\rangle$

These interpretations differ in dynamic potential (e.g. in the *specific* readings (9a) and (10a) the referent for "woman" is anaphorically available for continuations), and the resolution of such equivalences belongs to a different processing stage. In spite of this, the weaker, *non-specific* readings could in principle be dispensed with by extending S with '$\langle\{\}, \{\neg h\}\rangle$' and adapting *Plug* to prohibit indefinites under double negation (since generalized quantifiers cannot have arbitrary wide scope and thus do not induce such equivalences). Informally: $D(\langle\{\}, \{\neg\langle\{\ldots\}, \{\neg \ldots h \ldots\}\rangle\}\rangle, \otimes\{k, h_1, h_2\}) = \bot$.

4 Conclusion

A general restriction is formalized within Hole Semantics with the aim of efficiently avoiding the generation of equivalent quantifier scopings. The proposed scopal ordering constraint interrupts redundant disambiguations, being therefore able to significantly reduce the set of generated formulas.

Acknowledgements I am grateful to the anonymous referees for helpfull comments and to Johan Bos for providing a version of the Althaus et. al. (2003) constraint solver. The errors that I failed to correct are mine.

Bibliography

Alshawi, H. and R. Crouch (1992). Monotonic semantic interpretation. In *Proceedings of the 30th ACL*, pp. 32–39. Newark, Delaware.

Althaus, E., D. Duchier, A. Koller, K. Mehlhorn, J. Niehren, and S. Thiel (2003). An efficient graph algorithm for dominance constraints. *Journal of Algorithms*, To appear.

Blackburn, P. and J. Bos (1999). Representation and Inference for Natural Language - A First Course in Computational Semantics. `http://www.coli.uni-sb.de/~bos/comsem`.

Bos, J. (1996). Predicate logic unplugged. In *Proceedings of the 10th Amsterdam Colloquium*, pp. 133–142. ILLC / Department of Philosophy, University of Amsterdam, The Netherlands.

Chaves, R. P. (2002a). Principle-based DRTU for HPSG: a case study. In R. Malaka, R. Porzel, and M. Strube (Eds.), *Proceedings of the First International Workshop on Scalable Natural Language Understanding*, European Media Laboratory, Villa Bosh, Heidelberg, Germany.

Chaves, R. P. (2002b). *Fundamentos para uma Gramática Computacional do Português – Uma abordagem lexicalista.* Master's dissertation, University of Lisbon, Lisbon.

Cooper, R. (1983). *Quantification and Syntactic Theory.* Reidel Publications, Dordrecht.

Corblin, F. (1995). Compositionality and complexity in multiple negation. *IGPL Bulletin 3, 3-2*, 449–473.

Corblin, F. (1996). Multiple negation processing in Natural Language. In *Theoria*, pp. 214–260.

Egg, M. and K. Lebeth (1995). Semantic underspecification and modifier attachment ambiguities. In J. Kilbury and R. Wiese (Eds.), *Integrative Ansätze in der computer linguistik*, pp. 19–24. Düsseldorf, Seminar für Allgemeine Sprachwissenschaft.

Egg, M., J. Niehren, P. Ruhrberg, and F. Xu (1998). Constraints over lambda-structures in semantic underspecification. In *Joint COLING/ACL*, pp. 353–359. Montreal, Canada.

Proceedings of the Eighth ESSLLI Student Session
Balder ten Cate (editor)

Gabsdil, M. and K. Striegnitz (1999). Classifying scope ambiguities. In C. Montz and M. de Rijke (Eds.), *ICOS-1, Institute for Logic, Language and Computation (ILLC)*, Amsterdam, pp. 125–131.

Hobbs, J. R. and S. M. Shieber (1987). An algorithm for generating quantifier scopings. *Computational Linguistics 13(1-2)*, 47–55.

Kamp, H. and U. Reyle (1993). *From Discourse to Logic: An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT*. Dordrecht, Holland: Kluwer.

Keller, W. R. (1988). Nested Cooper storage: The proper treatment of quantification in ordinary noun phrases. In U. Reyle and C. Roher (Eds.), *Natural Language Parsing and Linguistic Theories*, Dordrecht: Reidel, pp. 432–447.

Koller, A., J. Niehren, and S. Thater (2003). Bridging the gap between underspecification formalisms: Hole semantics as dominance constraints. In *Proceedings of the 11th EACL*, Budapest.

McCawley, J. D. (1981). *Everything that Linguists always wanted to know about logic... but were ashamed to ask*. Chicago: The University of Chicago Press.

Reyle, U. (1993). Dealing with Ambiguities by Underspecification: Construction, Representation and Deduction. *Journal of Semantics 102*, 123–179.

Vestre, E. J. (1991). An algorithm for generating non-redundant quantifier scopings. In *Fifth Conference of the European Chapter of the ACL*, pp. 251–256. Berlin, Germany.