24. Semantic underspecification

Markus Egg

Humboldt-Universität Berlin, Institut für Anglistik und Amerikanistik

Unter den Linden 6, 10099 Berlin

markus.egg@anglistik.hu-berlin.de

+49/(0)30/2093-2295 (fax: -2244)

September 2009

24. Semantic underspecification

This article reviews semantic underspecification, which has emerged over the last three decades as a technique to capture several readings of an ambiguous expression in one single representation by deliberately omitting the differences between the readings in the semantic descriptions. After classifying the kinds of ambiguity to which underspecification can be applied, important properties of underspecification formalisms will be discussed that can be used to distinguish subgroups of these formalisms. The remainder of the article then presents various motivations for the use of underspecification, and expounds the derivation and further processing of underspecified semantic representations.

1. Introduction

Underspecification is defined as the deliberate omission of information from linguistic descriptions to capture several alternative realisations of a linguistic phenomenon in one single representation.

Underspecification emerged in phonology (see Steriade 1995 or Harris 2007 for an overview), where it was used e.g. for values of features that need not be specified because they can be predicted independently, e.g., by redundancy rules or by phonological processes. The price for this simplification, however, were additional layers or stages in phonological processes/representations, which resurfaces in most approaches that use underspecification in semantics.

In the 1980's, underspecification was adopted by semanticists. For semantics, the relevant linguistic phenomenon is *meaning*, thus, underspecified representations are intended to capture whole sets of different meanings in one representation. Since this does not apply to just any set of meanings, only those that correspond to the readings of one linguistic expression, semantic underspecification emerges as a technique for the treatment of *ambiguity.* (Strictly speaking, underspecification could be applied to *semantic indefiniteness* in general, which also encompasses vagueness, see Pinkal 1995. But since underspecification focusses almost exclusively on ambiguity, vagueness will be neglected.)

While underspecification is not restricted to expressions with systematically related sets of readings (as opposed to homonyms), it is in practice applied to such expressions only. The bulk of the work in semantic underspecification focusses on scope ambiguity.

In natural language processing, underspecification is endorsed to keep semantic representations of ambiguous expressions tractable and to avoid unnecessary disambiguation steps; a completely new use of underspecification emerged in *hybrid processing*, where it serves as a common format for the results of deep and shallow processing.

Underspecification is used also in syntax and discourse analysis to obtain compact representations whenever several structures can be assigned to a specific sentence (Marcus, Hindle & Fleck 1983; Rambow, Weir & Shanker 2001; Muskens 2001) or discourse, respectively (Asher & Fernando 1999; Duchier & Gardent 2001; Schilder 2002; Egg & Redeker 2008; Regneri, Egg & Koller 2008).

This article gives an overview over underspecification techniques in semantics. First the range of phenomena in semantics to which underspecification (formalisms) can be applied is sketched in section 2.. Section 3. outlines important properties of underspecification formalisms which distinguish different subgroups of these formalisms. Various motivations for using underspecification in semantics are next outlined in section 4..

The remaining two sections focus on the derivation of underspecified semantic representations by a suitable syntax-semantics interface (section 5.) and on the further processing of these representations (section 6.).

2. The domains of semantic underspecification

Before introducing semantic underspecification in greater detail, ambiguous expressions that are in principle amenable to a treatment in terms of semantic underspecification will be classified according to two criteria. These criteria compare the readings of these expressions from a semantic and a syntactic point of view, respectively, and are called *semantic* and *syntactic homogeneity*:

- Do the readings all comprise the *same semantic material*?
- Is it possible to give a *single syntactic analysis* for all the readings?

These criteria will classify ambiguity in four classes, which only partially coincides with the taxonomy in Bunt (2007). In the descriptions of these classes, I will also outline how they compare to Bunt's classes.

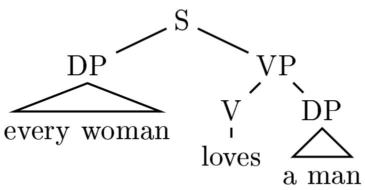
2.1 Semantically and syntactically homogeneous ambiguities

The main focus of attention in underspecification approaches to ambiguity is on ambiguous expressions that fulfil the two homogeneity conditions. Classic representatives of this group are quantifier scope ambiguities. (The word *quantifier* refers to DP meanings (sets of properties), except in expressions such as 'universal quantifier'.)

As an example, consider the well-worn (1) with the simplistic syntactic analysis (2) and its two readings (3a) 'for every woman, her own man' ($\forall > \exists$; '$>$' indicates scope of its left argument over the right one) and (3b) 'one man for all women' ($\exists > \forall$). Here and in (21) below, unary branching nodes are omitted. I ignore the discussion of whether indefinite quantifiers indeed introduce scope (see Kratzer 1998), my argumentation does not depend on this issue.

(1) Every woman loves a man.

(2)



The arrangement of the formulae in (3) highlights the fact that they consist of the same three parts (roughly coinciding with the semantic contributions of the verb and its two arguments), and that the relation of loving as introduced by the verb always gets lowest scope. The only difference between the formulae is the ordering of the semantic contributions of the arguments of the verb.

(3) a. $\forall x.\mathbf{woman'}(x) \rightarrow$ $\quad$ b. $\exists y.\mathbf{man'}(y) \wedge$

$\quad\quad \exists y.\mathbf{man'}(y) \wedge$ $\quad\quad \forall x.\mathbf{woman'}(x) \rightarrow$

$\quad\quad\quad \mathbf{love'}(x, y)$ $\quad\quad\quad \mathbf{love'}(x, y)$

Such cases of quantifier scope ambiguity are the prototypical domain for the application of underspecification, therefore, involved cases of quantifier scope ambiguity are handled in advanced underspecification formalisms. Some of these cases have developed into benchmark cases for underspecification formalisms. (4)-(6) belong to the group of these cases:

(4) Every researcher of a company saw most samples.

(5) [Every man]$_i$ read a book he$_i$ liked.

(6) Every linguist attended a conference, and every computer scientist did, too.

The subject in (4) illustrates *nested quantification*, where one quantifier-introducing DP comprises another one. The challenge of this example lies in the fact that the number of its readings is less than the number of the possible permutations of its quantifiers ($3! = 6$). The scope ordering that is ruled out in any case is $\forall > \mathbf{most'} > \exists$ (Hobbs & Shieber 1987). (While most approaches follow Hobbs & Shieber in assuming five readings for examples like (4), Park 1995 and Kallmeyer & Romero 2008 claim that in cases of nested quantification no quantifier may interfere between those introduced by the embedding and the embedded DP, regardless of their ordering. For (4), this would mean that the reading $\exists > \mathbf{most'} > \forall$ would have to be blocked, too, see section 3.1.)

In (5), the anaphoric dependency of *a book he liked* on *every man* restricts the quantifier scope ambiguity in that the DP with the anaphor must be in the scope of its antecedent (Reyle 1993).

In (6), quantifier scope is ambiguous, but must be the same in both sentences (i.e., if *every linguist* outscopes *a conference*, *every computer scientist* does, too). This yields two readings, and there is a third reading where *a conference* receives scope over everything else, i.e., both linguists and computer scientists attending the same conference (Hirschbühler 1982; Crouch 1995; Dalrymple, Shieber & Pereira 1991; Shieber, Pereira & Dalrymple 1996; Egg, Koller & Niehren 2001).

Other scope-bearing items can also enter into scope ambiguity, e.g., negation and modal expressions, as in the well-known examples (7) and (8):

(7) Everyone didn't come. ($\forall > \neg$ or $\neg > \forall$)

(8) A unicorn seems to be in the garden. ($\exists >$ *seem* or *seem* $> \exists$)

Such cases can also be described in terms of underspecification. This can be effected by underspecifying the scope of the quantifiers, with the other scope-bearing items being scopally fixed, e.g., in Minimal Recursion Semantics (Copestake et al. 2005).

But cases of scope ambiguity without quantifiers show that underspecifying quantifier scope only is not general enough. E.g., cases of '*neg raising*' (Sailer 2006) like in (9) have a reading denying that John believes that Peter will come, and one attributing to John the belief that Peter will not come:

(9) John doesn't think Peter will come.

Sailer analyses these cases as a scope ambiguity between the matrix verb and the negation (whose syntactic position is invariably in the matrix clause.)

Other such examples involve coordinated DPs, like in (10), (11), or (12) (Hurum 1988; Babko-Malaya 2004; Chaves 2005b):

(10) A man wants to marry Peggy or Sue.

(11) Every man and every woman solved a puzzle.

(12) Every lawyer and his secretary met.

(10) shows that in coordinated DPs scope ambiguity can arise between the conjunction and other scope-bearing material, i.e., it can emerge even in cases where DPs without scope (such as proper names) are coordinated. (10) is three-way ambiguous: The conjunction may have widest scope (there is either a man wishing to marry Peggy or another, possibly different man wishing to marry Sue), intermediate scope between *a man* and *want* (one man either wishing to

marry Peggy or wishing to marry Sue), or narrowest scope (one man wishing to marry either Peggy or Sue).

(11) has two readings, every man and every woman solving their own (possibly different) puzzle, or one puzzle being solved by every man and every woman. This shows that there are no intermediate readings where something can scopally intervene between conjoined scope-bearing DPs.

Finally, (12) has a reading in which every lawyer meets his own secretary, and one in which all the lawyers with their secretaries meet together. This example can be analysed in terms of a scope ambiguity between the operator G that forms groups out of individuals (assuming that only such groups can be the argument of a predicate like *meet*) and the conjoined DPs (Chaves 2005b). If G has narrow scope with respect to the DPs, every lawyer and his secretary form a specific group that meets (13a), if the DPs end up in G's restriction (indicated by brackets in (13)), there is one big meeting group consisting of all lawyers and their secretaries (13b).

(13) (a) $\forall x.\mathbf{lawyer'}(x) \rightarrow \exists y.\mathbf{secr\_of'}(y,x) \wedge \exists Z.[x \in Z \wedge y \in Z] \wedge \mathbf{meet'}(Z)$

(b) $\exists Z.[\forall x.\mathbf{lawyer'}(x) \rightarrow \exists y.\mathbf{secr\_of'}(y,x) \wedge x \in Z \wedge y \in Z] \wedge \mathbf{meet'}(Z)$

Another group of scope ambiguities is less visible, because it involves scope below the word level.

(14) beautiful dancer.

(15) John's former car.

(16) John almost died.

In (14), the adjective may pertain to the noun as a whole or to the stem only, which yields two readings that can roughly be glossed as 'beautiful person characterised by dancing' and 'person characterised by beautiful dancing', respectively (Larson 1998). This can be modelled as scope ambiguity between the adjective and the nominal affix *-er* (Egg 2004). (15) as discussed in Larson & Cho (2003) is a case of scope ambiguity between the possessive relation introduced by the Anglo-Saxon genitive *'s* and the adjective *former*, which yields the readings 'car formerly in the possession of John' or 'ex-car in the possession of John' (Egg 2007). Finally, the readings of (16), viz., 'John was close to undergoing a change from being alive to being dead' (i.e., in the end, nothing happened to him) and 'John underwent a change from being alive to being close to death' (i.e., something did happen) can be modelled as scope ambiguity between a change-of state operator like BECOME and the adverbial (Rapp & von Stechow 1999; Egg 2007).

Analyses of these cases in Generative Grammar reconstruct the ambiguity in terms of different syntactic constellations that involve constituents below the word level. These constituents can correspond to morphemes (as in the case of *dancer* or the Anglo-Saxon genitive), but need not (e.g., for the change-of-state operator in the semantics of *die*). (Note that the existence of such syntactically heterogeneous analyses is not incompatible with my claim that these cases are syntactically homogeneous: For syntactic homogeneity it is sufficient that a single syntactic analysis for all readings is *possible*.)

The cases of semantically and syntactically homogeneous ambiguity discussed so far have readings that not only comprise the same semantic building blocks, each reading has in addition exactly one instance of each of these building blocks. This was highlighted e.g. for (1) in the representation of its readings in (3), where each semantic building block appears on a different line.

However, the definition of semantically and syntactically homogeneous ambiguity includes also cases where the readings consist of the same building blocks, but differ in that some of the readings exhibit more than one instance of specific building blocks.

A prime example of this kind of semantically and syntactically homogeneous ambiguity is the ellipsis in (17). Its two readings 'John wanted to greet everyone that Bill greeted' and 'John wanted to greet everyone that Bill wanted to greet' differ in that there is only one instance of the semantic contribution of the matrix verb *want* in the first reading as opposed to two instances in the second reading (Sag 1976):

(17) John wanted to greet everyone that Bill did.

This is due to the fact that the pro-form *did* is interpreted in terms of a suitable preceding VP, and that there are two such suitable VPs in (17), viz., *wanted to greet everyone that Bill did* and *greet everyone that Bill did.* ((17) is a case of *antecedent-contained deletion*, see Shieber, Pereira & Dalrymple 1996 and Egg, Koller & Niehren 2001 for underspecified accounts of this phenomenon.)

Another example of this kind of semantically and syntactically homogeneous ambiguity is the case of the Afrikaans past tense in (18) (Sailer 2004). There are two tense markers, the inflected form of the matrix verb *wou* 'wanted' and the auxiliary *het* in the subordinate clause, both of which introduce a past tense operator. But these examples have three readings:

(18) Jan wou gebel het.
Jan want.PAST called have
'Jan wanted to call/Jan wants to have called/Jan wanted to have called.'

The readings can be analysed schematically (in the order given in (18)) as (19a-c): I.e., the readings of (18) comprise one or two instances of the past tense operator:

(19) a. $\mathrm{PAST}(\mathbf{want'}(\mathbf{j},^\wedge(\mathbf{call'}(\mathbf{j}))))$

b. $\mathbf{want'}(\mathbf{j},^\wedge \mathrm{PAST}(\mathbf{call'}(\mathbf{j})))$

c. $\mathrm{PAST}(\mathbf{want'}(\mathbf{j},^\wedge \mathrm{PAST}(\mathbf{call'}(\mathbf{j}))))$

Finally, the criterion 'syntactically and semantically homogeneous' as defined in this subsection will be compared to similar classes of ambiguity from the literature. Syntactic and semantic homogeneity is sometimes referred to as *structural ambiguity*. But this term is itself ambiguous in that it is sometimes used in the broader sense of 'semantically homogeneous' (i.e., syntactically homogeneous or not). But then it would also encompass the group of semantically but not syntactically homogeneous ambiguities discussed in the next subsection.

The group of semantically and syntactically homogeneous ambiguities coincides by and large with Bunt's (2007) 'structural semantic ambiguity' class. Exceptions are the ambiguity of compounds like *math problem* and the collective/distributive ambiguity of quantifiers, which I classify as syntactically but not semantically homogeneous: Different readings of a compound each instantiate an unspecific semantic relation between the components in a specific, non-identical way. Similarly, distributive and quantitative readings of a quantifier are distinguished in the semantics by the presence or absence of a distributive or collective operator, e.g., Link's (1983) distributive D-operator.

2.2 Semantically but not syntactically homogeneous ambiguities

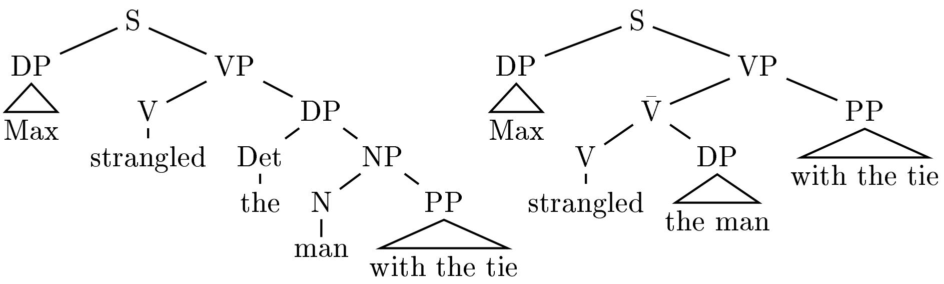
The second kind of ambiguity is semantically but not syntactically homogeneous. The ambiguity has a syntactic basis in that the same *syntactic* material is arranged in different ways. Consequently, the meanings of the resulting syntactic structures all consist of the same semantic material (though differently ordered, depending on the respective syntactic structure), but no common syntactic structure can be postulated for the different interpretations.

As a prime example of semantically but not syntactically homogeneous ambiguity, consider the notorious modifier attachment ambiguities as in (20):

(20) Max strangled the man with the tie.

There is no common phrase marker for the two readings of (20). In the reading that the man is wearing the tie, the constituent *the tie* is part of a DP (or NP) *the man with the tie*. In the other reading, in which the tie is the instrument of Max' deed, *the tie* enters a verbal projection (as the syntactic sister of *strangled the man*) as a constituent of its own:

(21) a. 'tie worn by victim' b. 'tie as instrument of crime'



There is an intuitive 1:1 relation between the two phrase markers in (21) and the two readings of (20). None of the phrase markers would be suitable as the syntactic analysis for *both* readings.

Semantically but not syntactically homogeneous ambiguity is usually not described in terms of semantic underspecification in the same fashion as semantically and syntactically homogeneous ambiguity; exceptions include Muskens (2001), Pinkal (1996), or Richter & Sailer (1996).

In Bunt's classification, the group of semantically but not syntactically homogeneous ambiguites are called 'syntactic ambiguity'.

2.3 Syntactically but not semantically homogeneous ambiguities

The third kind of ambiguity is instantiated by expressions whose readings share a single syntactic analysis but do not comprise the same semantic material.

These expressions can be classified in four subgroups. Members of the first subgroup comprise *lexically* ambiguous words, whose ambiguity is inherited by the whole expression. E.g., the ambiguity of the preposition *into* between a dynamic reading (a change of state whose result is location inside the object denoted by the NP argument of the preposition) and a stative reading (a state of being partially outside and partially inside this object) makes expressions like *into the garden* ambiguous, too.

For polysemy (as opposed to homonymy) it is feasible to give an underspecified account by modelling the semantics of the polysemous item in terms of the core meaning common to all readings. This was worked out in the so-called *two-level* semantics (Bierwisch 1983; Bierwisch & Lang 1987; Bierwisch 1988), which distinguished a level of semantics (where the core meanings reside) and relegated the specification of the individual readings to a conceptual level. In the case of *into*, the ambiguity can be captured in terms of a core meaning that comprises an abstract operator CHANGE. This operator can be instantiated on the conceptual level either temporally (yielding a change-of state operator), or spatially (which returns the stative reading) (Wunderlich 1991).

Underspecification formalisms that take into account polysemy comprise the semantic representation language in the PHLIQA question-answering system (Bronnenberg et al. 1979), Poesio's (1996) Lexically Underspecified Language LXUL, and Cimiano & Reyle's (2005) extension of Muskens's (2001) Logical Description Grammar.

Homonymy has not been a prime target of underspecification, because there is not enough common ground between the readings that would support a sufficiently distinctive underspecified representation (that would not be identical to the representation of other lexical items). Consider e.g., *jumper* in its textile and its electrical engineering sense: 'concrete object' as common denominator of the readings would fail to distinguish *jumper* from a similarly underspecified representation of the homonym *pen* ('writing instrument' or 'device for sheep').

Such lexical ambiguities were also sppotted in sentences with quantifiers that have *collective* and *distributive* readings (Alshawi 1992; Frank & Reyle 1995;

Chaves 2005a). E.g., in (22), the lawyers can act together or individually:

(22) The lawyers hired a secretary.

The distributive reading differs from the collective one in that there is a quantification over the set of lawyers whose scope is the property of hiring a secretary (instead of having this property apply to an entity consisting of all lawyers together). The collective reading lacks this quantification, which makes expressions like (22) semantically heterogeneous.

The proposed analyses of this ambiguity locate the ambiguity differently. The Core Language Engine account (Alshawi 1992) and the Underspecified DRT (UDRT) account of Frank & Reyle (1995) suggest an underspecification of the DP semantics (they refer to DPs as NPs) that can be specified to a collective or a distributive interpretation.

Chaves (2005a) notes that mixed readings like in (23) are wrongly ruled out if the ambiguity is attributed to the DP semantics.

(23) The hikers met in the train station and then left.

His UDRT analysis places the ambiguity in the verb semantics in the form of an underspecified operator, which can be instantiated as universal quantification in the spirit of Link's (1983) account of distributive readings.

Lexically based ambiguity includes also compounds like *math problem*. Their semantics comprises a not specified relation between their components, which is specified differently in the various readings (e.g., for *math problem*, 'mathematical problem' or 'problem with understanding mathematics').

Referential ambiguity is the second subgroup of syntactically but not semantically homogeneous expressions, because there are different interpretations of a deictic expression, which is eventually responsible for the ambiguity. For a discussion of referential ambiguity and its underspecificed representation, see e.g. Asher & Lascarides (2003) and Poesio et al. (2006).

Some cases of of referential ambiguity are due to ellipses where the VPs in terms of which the pro-forms are to be interpreted comprise anaphors, e.g., the pro-form *does* and the VP *walks his dog* in (24):

(24) John walks his dog and Max does, too.

The interpretation of *does* in terms of *walks his dog* comprises an anaphor, too. This anaphor can refer to the same antecedent as the one in *walks his dog* ('strict' reading, Max walks John's dog), or to its own subject DP in analogy to the way in which the anaphor in *John walks his dog* refers ('sloppy' reading, Max walks his own dog). For much more complex examples of this kind, see Gawron & Peters (1990).

A further kind of syntactically but not semantically homogeneous ambiguity where underspecification has been proposed is *missing information* (Pinkal 1999). In this case, parts of a message could not be decoded due to problems in production, transmission, or reception. These messages can be interpreted in different ways (depending on how the missing information is filled in), while the syntactic representation remains constant.

Finally, the fourth subgroup is *reinterpretation* (metonymy and aspectual coercion). It can pattern with homonymy, if it is modelled in terms of underspecified operators that are inserted during semantic construction (Hobbs et al. 1993, Dölling 1995; Pulman 1997; de Swart 1998; Egg 2005). Such operators will avoid impending clashes for semantic construction by being inserted between otherwise (mostly) incompatible semantic material during the construction process.

This strategy can introduce ambiguity, e.g., in (25). Here a coercion operator is inserted between *play the Moonlight Sonata* and its modifier *for some time*, which cannot be combined directly; this operator can be specified to a progressive or an iterative operator (i.e., she played part of the sonata, or she played the sonata repetitively):

(25) Amélie played the Moonlight Sonata for some time.

The readings of such expressions have a common syntactic analysis, but, due to the different specification of the underspecified reinterpretation operator, they no longer comprise the same semantic material.

Syntactically but not semantically homogeneous ambiguities (together with vagueness) encompass Bunt's (2007) classes 'lexical ambiguity', 'semantic imprecision', and 'missing information' with the exception of ellipsis: In ellipsis (as opposed to incomplete utterances), the missing parts in the target sentences are recoverable from the preceding discourse (possibly in more than one way), while no such possibility is available for incomplete utterances (e.g., for the utterance *Bill?* in the sense of *Where are you, Bill?*).

2.4 Neither syntactically nor semantically homogeneous ambiguities

To complete the typology of ambiguity, there are also ambiguous expressions that are neither syntactically nor semantically homogeneous, but these have the status of marginal (and often jocular) expressions like (26):

(26) We saw her duck.

The fringe status of this group might also be the reason why it does not show up in Bunt's (2007) taxonomy.

2.5 The focus of underspecified approaches to ambiguity

While underspecification can in principle be applied to all four groups of ambiguity, most of the work on underspecification focusses on semantically and syntactically homogeneous ambiguity. In my opinion, there are two reasons for this: First, it is more attractive to apply underspecification to semantically homogeneous (than to semantically heterogeneous) ambiguity: Suitable underspecified representations of a semantically homogeneous ambiguous expression can delimit the range of readings of the expression and specify them fully without disjunctively enumerating them (for a worked out example, see the discussion of example (41) on p. 28f.).

No such delimitation and specification are possible in the case of semantically heterogeneous ambiguity: Here semantic representations must restrict themselves to specifying the parts of the readings that are common to all of them and leave open those parts that distinguish the specific readings. Further knowledge sources are then needed to define the possible instantiations of these parts (which eventually delimits the set of readings and fully specifies them).

Second, syntactically heterogeneous ambiguity seems to be considered less of an issue for the syntax-semantics interface, because there each reading is motivated by a syntactic structure of its own, and underspecified presentations of these readings would then cancel out the differences between the readings in spite of their independent syntactic motivation. No such syntactic motivation of ambiguity is available for syntactically homogeneous ambiguity, which makes it a much greater challenge for the syntax-semantics interface (see section 4.1 for further discussion of this point).

I will go along with the trend in underspecification research and focus on syntactically and semantically homogeneous ambiguities in the remainder of this article.

3. Approaches to semantic underspecification

This section is devoted to the general description of underspecification formalisms. It will outline general properties that characterise these formalisms and distinguish subgroups of them.

I will first show that underspecification formalisms handle ambiguity by either *describing* it or by providing an algorithm for the *derivation* of the different readings of an ambiguous expression. Then I will point out that these formalisms may but need not distinguish different *levels of representation*, and implement *compositionality* in different ways. Finally, underspecification formalisms also differ with respect to their *compactness* (how efficiently can they delimit and specify the set of readings of an ambiguous expression) and their *expressivity* (can they also do this for arbitrary subsets of this set of readings).

3.1 Describing ambiguity

Underspecification is implemented in semantics in two different ways, in that the readings of an ambiguous expression can either be *described* or *derived*. This distinction shows up also in Robaldo (2007), who uses the terms 'constraint-based' and 'enumerative'. In a (no longer current) version of Glue Language Semantics (Shieber, Pereira & Dalrymple 1996) both approaches are mixed to handle antecedent-contained deletion as in (17).

The first way of implementing semantic underspecification is to describe the meaning of an ambiguous expression directly. The set of semantic representations for its readings is characterised in terms of *partial information* rather than in terms of *disjunction* or *enumeration*. This characterisation by itself delimits the range of readings of the ambiguous expression and specifies them. I.e., the way in which fully specified representations for the readings are derived from the underspecified representation does not contribute to the delimitation.

This strategy is based on the fact that there are two ways of describing a set: enumerating the elements or giving a property that characterises all the and only the elements of the set. In the second way, a set of semantic representations is defined by describing the common ground between the representations only. This description must be compatible with all the and only the elements of the set. Since it deliberately leaves out everything that distinguishes the elements of the set, the description can only be partial.

Most underspecification formalisms that follow this strategy distinguish an object level (semantic representations) and a meta-level (descriptions of these representations) at this point. The formalisms define the expressions of the meta-level and their relation to the described object-level representations.

3.1.1 A simple example

As an illustration, consider once more (27) [= (1)] and its ∀∃- and ∃∀-readings (28a-b) [= (3a-b)]:

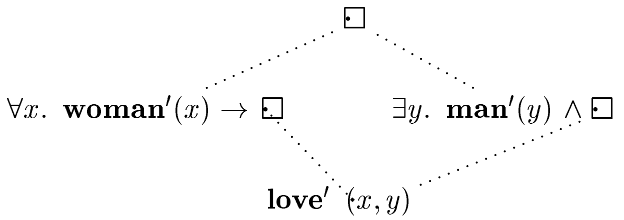
(27) Every woman loves a man

(28) (a) $\forall x.\mathbf{woman}'(x) \rightarrow \exists y.\mathbf{man}'(y) \wedge \mathbf{love}'(x,y)$

(b) $\exists y.\mathbf{man}'(y) \wedge \forall x.\mathbf{woman}'(x) \rightarrow \mathbf{love}'(x,y)$

A description of the common ground in (28) can look like this:

(29)



In (29), we distinguish four *fragments* of semantic representations (here, λ-terms) which may comprise *holes* (parts of fragments that are not yet determined, indicated by '□'). Then there is a relation R between holes and fragments (depicted as dotted lines), if R holds for a hole h and a fragment F, F must be part of the material that determines h.

R determines a partial scope ordering between fragments: A fragment F_1 has scope over another fragment F_2 iff F_1 comprises a hole h such that $R(h, F_2)$

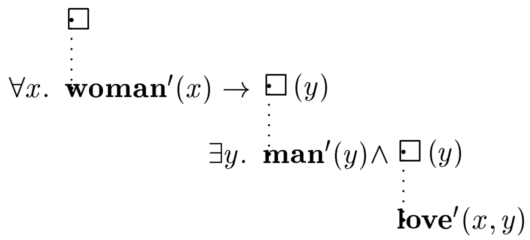
or $R(h, F_3)$, where F_3 is a third fragment that has scope over F_2 (cf. e.g. the definition of 'qeq relations' in Copestake et al. 2005). Furthermore, we assume that variable binding operators in a fragment F bind occurrences of the respective variables in all fragments outscoped by F (ignoring the so-called variable capturing problem, see Egg, Koller & Niehren 2001) and that the description explicates all the fragments of the described object-level representations.

The description (29) can then be read as follows: The fragment at the top consists of a hole only, i.e., we do not yet know what the described representations look like. However, since the relation R relates this hole and the right and the left fragment, they are both part of these representations - only the order is open. Finally, the holes in both the right and the left fragment are related to the bottom fragment in terms of R, i.e., the bottom fragment is in the scope of either quantifier. The only semantic representations compatible with this description are (28a-b), as desired.

To derive the described readings from such a constraint (its *solutions*), the relation R between holes and fragments is monotonically strengthened until all the holes are related to a fragment, and all the fragments except the one at the top are identified with a hole (this is called 'plugging' in Bos 2004).

In our example, one can strengthen R by adding the pair consisting of the hole in the left-hand fragment and the right-hand fragment. Here the relation between the hole in the universal fragment and the bottom fragment in (29) is omitted because it follows from a specific property of R: If $R(h_1, F_1)$, and F_1 comprises a hole h_2 such that $R(h_2, F_2)$, then $R(h_1, F_2)$. This property is eventually based on the fact that the order models a part-of relation between holes and fragments.

(30)



Identifying the hole-fragment pairs in R in (30) then yields (28a), one of the solutions of (29). The other solution (28b) can be derived by first adding to R the pair consisting of the hole in the right fragment and the left fragment.

Underspecification formalisms that implement scope in this way comprise Underspecified Discourse Representation Theory (UDRT; Reyle 1993; Reyle 1996; Frank & Reyle 1995), Minimal Recursion Semantics (MRS, Copestake et al. 2005), the Constraint Language for Lambda Structures (CLLS; Egg, Koller & Niehren 2001), the language of Dominance Constraints (DC, subsumed by CLLS; Althaus et al. 2001), Hole Semantics (HS; Bos 1996; Bos 2004; Kallmeyer & Romero 2008), and Logical Description Grammar (Muskens 2001).

Koller, Niehren & Thater (2003) show that expressions of HS can be translated into expressions of DC and vice versa; Fuchss et al. (2004) describe how to translate MRS expressions into DC expressions. Player (2004) claims that this is due to the fact that UDRT, MRS, CLLS, and HS are the same ‘modulo cosmetic differences’, however, his comparison does not pertain to CLLS but to the language of dominance constraints.

Scope relations like the one between a quantifying DP and the verb it is an argument of can also be expressed in terms of suitable variables. This is implemented e.g. in the Underspecified Semantic Description Language (USDL;

Pinkal 1996, Niehren, Pinkal & Ruhrberg 1997; Egg & Kohlhase 1997 present a dynamic version of this language). In USDL, the constraints for (27) are expressed by the equations in (31):

(31) (a) $X_0 = C_1(\mathbf{every\_woman}@L_{x_1}(C_2(\mathbf{love}@x_2@x_1)))$

(b) $X_0 = C_3(\mathbf{a\_man}@L_{x_2}(C_4(\mathbf{love}@x_2@x_1)))$

Here ‘**every_woman**’ and **a_man**’ stand for the the two quantifiers in the semantics of (27), ‘@’ denotes explicit functional application in the metalanguage, and ‘L_x’, λ-abstraction over x.

These equations can now be solved by an algorithm like the one in Huet (1975). E.g., for the ∀∃-reading of (27), the variables would be resolved as in (32a-c). This yields (32d), whose right hand side corresponds to (28a):

(32) (a) $C_1 = C_4 = \lambda P.P$

(b) $C_2 = \lambda P.\mathbf{a\_man}@L_{x_2}(P)$

(c) $C_3 = \lambda P.\mathbf{every\_woman}@L_{x_1}(P)$

(d) $X_0 = \mathbf{every\_woman}'@L_{x_1}(\mathbf{a\_man}@L_{x_2}(\mathbf{love}@x_2@x_1))$

Another way to express such scope relations is used in the version of the Quasi-Logical Form (QLF) in Alshawi & Crouch (1992), which uses list-valued meta-variables in semantic representations whose specification indicates quantifier scope. Consider e.g. the (simplified) representation for (27) in (33a), which comprises an underspecified scoping list (the variable `_s` before the colon). Here the meanings of *every woman* and *a man* are represented as complex terms; such terms comprise (among other things) term indices (`+m` and `+w`) and the restrictions of the quantifiers (`man` and `woman`, respectively). Specifying this underspecified reading to the reading with wide scope for the universal quantifier then consists in instantiating the variable `_s` to the list `[+w,+m]` in (33b), which corresponds to (28a):

(33) (a) `_s:love(term(+w,...,woman,...), term(+m,...,man,...))`

(b) `[+w,+m]:love(term(+w,...,woman, ...),`
`term(+m,...,man,...))`

Even though QLF representations seem to differ radically from the ones that use dominance constraints, Lev (2005) shows how to translate them into a expressions of an underspecification formalism based on dominance relations (viz., Hole Semantics).

Finally, I will show how Glue Language Semantics (GLS; Dalrymple et al. 1997; Crouch & van Genabith 1999; Dalrymple 2001) handles scope ambiguity. Each lexical item introduces so-called *meaning constructors* that relate syntactic constituents (I abstract away from details of the interface here) and semantic representations. E.g., for the proper name *John*, the constructor is '$DP \rightsquigarrow \mathbf{john}'$', which states that the DP *John* has the meaning $\mathbf{john}'$ ('$\rightsquigarrow$' relates syntactic constituents and their meanings).

In more involved cases, such statements are arguments of connectives of *linar logic* like the conjunction $\otimes$ and the implication $\multimap$, e.g., the meaning constructor for *love*:

(34) $\forall X, Y.DP_{subj} \rightsquigarrow X \otimes DP_{obj} \rightsquigarrow Y \multimap S \rightsquigarrow \mathbf{love}'(X, Y)$

In prose: Whenever the subject interpretation in a sentence S is X and the object interpretation is Y, then the S meaning is $\mathbf{love}'(X, Y)$. I.e., these constructors specify how the meanings of smaller constituents determine the meaning of a larger constituent.

The implication $\multimap$ is resource-sensitive: '$A \multimap B$' can be paraphrased as 'use a resource A to derive (or produce) B'. The resource is 'consumed' in this process, i.e., no longer available for further derivations. Thus, from A and $A \multimap B$ one can deduce B, but no longer A. For (34), this means that after deriving the S meaning the two DP interpretations are no longer available for further processes of semantic construction (consumed).

The syntax-semantics interface collects these meaning constructors during the construction of the syntactic structure of an expression, and, crucially, instantiates and/or identifies specific constituents that are mentioned in them.

For ambiguous expressions such as (27), the resulting collection of meaning constructors can be regarded as an underspecified representation of its different readings. Representations for the readings of the expression can then be derived from this collection of constructors by linear-logic deduction.

In the following, the presentation is simplified in that DP-internal semantic construction is omitted and only the DP constructors are given:

(35) (a) $\forall H, P.(\forall x.DP \rightsquigarrow x \multimap H \rightsquigarrow_t P(x)) \multimap H \rightsquigarrow \mathbf{every}'(\mathbf{woman}', P)$

(b) $\forall G, R.(\forall y.DP \rightsquigarrow y \multimap G \rightsquigarrow_t R(y)) \multimap G \rightsquigarrow \mathbf{a}'(\mathbf{man}', R)$

The semantics of *every woman* in (35a) can be paraphrased as follows: Look for a resource of the kind 'use a resource that a DP semantics is x, to build the truth-valued (subscript t of $\rightsquigarrow$) meaning $P(x)$ of another constituent H'. Then consume this resource and assume that the semantics of H is $\mathbf{every}'(\mathbf{woman}', P)$; here $\mathbf{every}'$ abbreviates the usual interpretation of *every*. The representation for *a man* works analogously.

With these constructors for the verb and its arguments, the semantic representation of (27) in GLS is (36d), the conjunction of the constructors of the verb and its arguments. Note that semantic construction has identified the DPs that are mentioned in the three constructors:

(36) (a) $\forall H, P.(\forall x.DP_{subj} \rightsquigarrow x \multimap H \rightsquigarrow_t P(x)) \multimap H \rightsquigarrow \mathbf{every}'(\mathbf{woman}', P)$

(b) $\forall G, R.(\forall y.DP_{obj} \rightsquigarrow y \multimap G \rightsquigarrow_t R(y)) \multimap G \rightsquigarrow \mathbf{a}'(\mathbf{man}', R)$

(c) $\forall X, Y.DP_{subj} \rightsquigarrow X \otimes DP_{obj} \rightsquigarrow Y \multimap S \rightsquigarrow \mathbf{love}'(X, Y)$

(d) (36a)$\otimes$(36b)$\otimes$(36c)

From such conjunctions of constructors, fully specified readings can be derived. For (27), the scope ambiguity is modelled in GLS in that two different semantic representations for the sentence can be derived from (36d).

Either derivation starts with choosing one of the two possible specifications of the verb meaning in (36c), which determine the order in which the argument interpretations are consumed:

(37) (a) $\forall X.DP_{subj} \rightsquigarrow X \multimap (\forall Y.DP_{obj} \rightsquigarrow Y \multimap S \rightsquigarrow \mathbf{love}'(X, Y))$

(b) $\forall Y.DP_{obj} \rightsquigarrow Y \multimap (\forall X.DP_{subj} \rightsquigarrow X \multimap S \rightsquigarrow \mathbf{love}'(X, Y))$

I will now illustrate the derivation of the reading of $\forall\exists$-reading of (27). The next step uses the general derivation rule (38) and the instantiations in (39):

(38) from $A \multimap B$ and $B \multimap C$ one can deduct $A \multimap C$

(39) $G \mapsto \mathrm{S}$, $Y \mapsto y$, and $R \mapsto \lambda y.\mathbf{love}'(X, y))$

From specification (37a) and the object semantics (36b) we then obtain (40a), this goes then together with the subject semantics (36a) to yield (40b), a notational variant of (28a):

(40) (a) $\forall X.DP_{subj} \rightsquigarrow X \multimap S \rightsquigarrow \mathbf{a}'(\mathbf{man}', \lambda y.\mathbf{love}'(X, y))$

(b) $\mathbf{every}'(\mathbf{woman}', \lambda x.\mathbf{a}'(\mathbf{man}', \lambda y.\mathbf{love}'(x, y))$

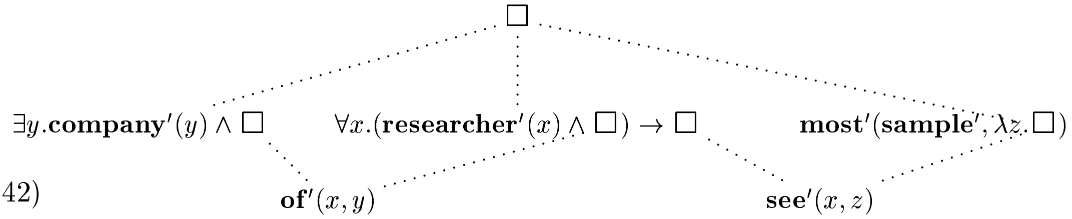
The derivation for the other reading of (27) chooses the other specification (37b) of the verb meaning and works analogously.

3.1.2 A more involved example

After this expository account of the way that the simple ambiguity of (27) is captured in various underspecification formalisms, reconsider the more involved nested quantification in (41) [= (4)], whose constraint is given in (42).

(41) Every researcher of a company saw most samples

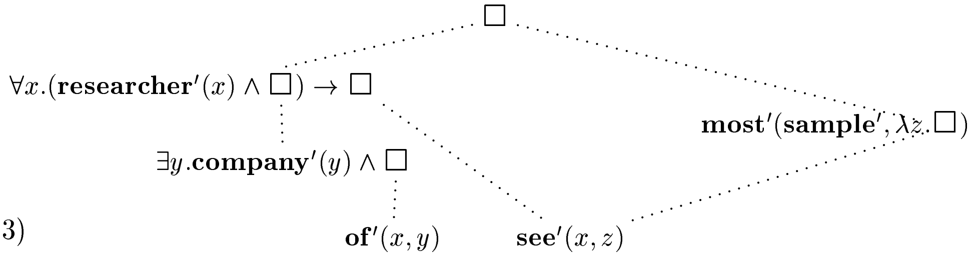
(42)



As expounded in section 2.1, not all scope relations of the quantifiers are possible in (41). I assume that (41) has exactly five readings, the one that is ruled out and hence must be excluded in a suitable underspecified representation of (41) is the one with the scope ordering $\forall > \mathbf{most}' > \exists$.

As a first step of disambiguation, we can order the existential and the universal fragment. Giving the former narrow scope yields (43):

(43)



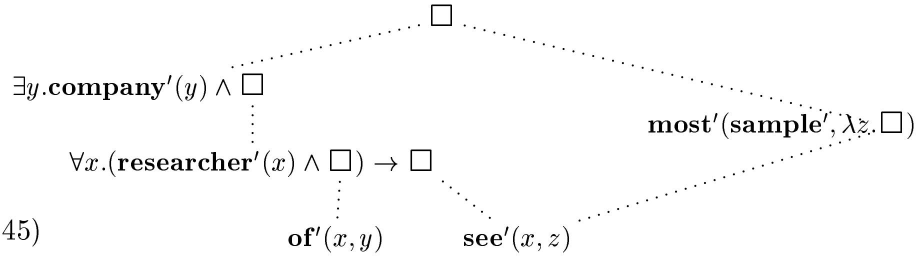
But once the existential fragment is outscoped by the universal fragment, it can no longer interact scopally with the *most*- and the *see*-fragment, because it is part of the *restriction* of the universal quantifier. I.e., there are two readings to be derived from (43), with the *most*-fragment scoping below or above the universal fragment. This rules out a reading in which *most* scopes below the universal, but above the existential quantifier:

(44) (a) $\forall x.(\mathbf{researcher}'(x) \wedge \exists y.\mathbf{company}'(y) \wedge \mathbf{of}'(x,y)) \to \mathbf{most}'(\mathbf{sample}', \lambda z.\mathbf{see}'(x,z))$

(b) $\mathbf{most}'(\mathbf{sample}', \lambda z \forall x.(\mathbf{researcher}'(x) \wedge \exists y.\mathbf{company}'(y) \wedge \mathbf{of}'(x,y)) \to \mathbf{see}'(x,z))$

The second way of fixing the scope of the existential w.r.t. the universal quantifier in (42) gives us (45):

(45)



This constraint describes three readings, whose difference is whether the *most*-fragment takes scope over, between, or below the other two quantifiers. In sum, constraint (42) encompasses the five desired interpretations:

(46) (a) $\mathbf{most}'(\mathbf{sample}', \lambda z \exists y.\mathbf{company}'(y) \wedge \forall x.(\mathbf{researcher}'(x) \wedge \mathbf{of}'(x,y)) \rightarrow \mathbf{see}'(x,z))$

(b) $\exists y.\mathbf{company}'(y) \wedge \mathbf{most}'(\mathbf{sample}', \lambda z \forall x.(\mathbf{researcher}'(x) \wedge \mathbf{of}'(x,y)) \rightarrow \mathbf{see}'(x,z))$

(c) $\exists y.\mathbf{company}'(y) \wedge \forall x.(\mathbf{researcher}'(x) \wedge \mathbf{of}'(x,y)) \rightarrow \mathbf{most}'(\mathbf{sample}', \lambda z.\mathbf{see}'(x,z))$

Kallmeyer & Romero (2008) block reading (46b) by the additional constraint that the quantifier Q_1 from the embedding DP outscopes the (immediate) scope of the quantifier Q_2 from the embedded DP. If this is resolved to identity, Q_2 has immediate scope over Q_1, otherwise, Q_1 has scope over Q_2.

For (42), this would affect the partial resolution in (45): Here the universal Q_1-fragment would have to be equated with the hole in the existential Q_2-fragment, i.e., there would be no more gap for the *most*-fragment to slip in between. The additional constraint would not affect the partial resolution in (44), where the universal fragment has scope over the existential fragment, and hence also over its scope hole, which would yield four readings altogether.

However, (41) is only a simple case of nested quantification. The challenge for underspecified representation lies in the fact that expressions with such nested quantifying DPs have less readings than the factorial of the number of the involved DPs, since some scoping options are ruled out. For instance, simple sentences consisting of a transitive verb with two arguments that together comprise n quantifying DPs have $C(n)$ readings, where $C(n)$ is the Catalan number of n ($C(n) = \frac{(2n)!}{(n+1)!n!}$). E.g., example (47) has 5 nested quantifiers and thus $C(5) = 42$ readings (Hobbs & Shieber 1987). Appropriate underspecification formalisms can handle nested quantification in general.

(47) Some representative of every department in most companies saw a few samples of each product

This example highlights the two main characteristics of this approach to semantic underspecification: Underspecified expressions (typically, of a meta-level formalism) *describe* a set of semantic representations and at the same time intend to *delimit* and *fully specify* the range of this set. The derivation of solutions from such expressions does thus not add information in that it restricts the number of solutions in any way.

3.2 Deriving ambiguity

The second approach to semantic underspecification differs in that it does not directly describe object-level semantic representations. For example, representations of structurally ambiguous expressions in the formalism of Schubert & Pelletier (1982) describe the semantics of DPs as *terms*, i.e., scope-bearing expressions whose scope has not been determined yet. Terms are triples of a quantifier, a bound variable, and a restriction. E.g., the initial semantic representation of (27) is (48), which closely resembles its syntactic structure:

(48) $\mathbf{love}'(\langle \text{forall} \quad x \quad \mathbf{woman}'(x)\rangle, \langle \text{exists} \quad y \quad \mathbf{man}'(y)\rangle)$

The set of fully specified representations encompassed by such a representation is then determined by a resolution algorithm that integrates terms by 'discharging' them at appropriate positions within the representation (i.e., applying them to suitable parts of the representation and thereby determining their scope). E.g., to obtain the representation (28a) for the '∀∃'-reading of (27) one would first integrate the existential term (formally: replace it by the bound variable and prefix the quantifier with the term's bound variable and restriction to the resulting expression), which yields (49):

(49) $\exists y.\mathbf{man'}(y) \wedge \mathbf{love'}(\langle \text{forall} \;\; x \;\; \mathbf{woman'}(x)\rangle, y)$

Integrating the remaining term then yields (28a); to derive (28b) from (48), one would have to integrate the universal term before the existential one. Such an approach is adopted e.g. in the Core Language Engine version described in Moran (1988) and Alshawi (1992).

While the resolution of representations such as (48) is intuitively clear, Hobbs & Shieber (1987) show that a rather involved algorithm is called for to prevent overgeneration in more complicated cases, in particular, for nested quantification. Initial semantic representations for nested quantification comprise nested terms, consider e.g. the representation (50) for (41):

(50) $\mathbf{see'}(\langle \text{forall} \;\; x \;\; \mathbf{researcher'}(x) \wedge \mathbf{of'}(x, \langle \text{exists} \;\; y \;\; \mathbf{company'}(y)\rangle)\rangle,$
$\qquad\qquad \langle \text{most} \;\; z \;\; \mathbf{sample'}(z)\rangle)$

Here the restriction on the resolution is that the inner quantifier may never be integrated before the outer one, which in the case of (41) rules out the unwanted 6th possible permutation of the quantifiers. Otherwise, this permutation could be generated by integrating the terms in the order '$\mathbf{most}', \exists, \forall$'. I.e., the algorithm must be designed in such a way that it does the work of (42).

Such resolution algorithms lend themselves to a straightforward integration of *preference rules* such as '*each* outscopes other determiners', see section 6.4.

Other ways of handling nested quantification in terms of externally restricting the resolution of underspecified representations have been discussed in the literature. First, one could block *vacuous binding* (even though vacuous binding would not make formulae ill-formed), i.e., requesting an appropriate bound variable in the scope of every quantifier. Translated into Hobbs & Shieber's (1987) terms, this would mean that in the resolution of the representation (52) for (51) the step from (52) to (53) is blocked, because the discharged quantifier fails to bind an occurence of a variable y in its scope (the only occurrence of y in its scope is inside a term, hence not accessible for binding). Thus, the unwanted solution (54) cannot be generated:

(51) Every researcher of a company came

(52) $\mathbf{come}'(\langle \text{forall} \quad x \quad \mathbf{researcher}'(x) \wedge \mathbf{of}'(x, \langle \text{exists} \quad y \quad \mathbf{company}'(y)\rangle)\rangle)$

(53) $\exists y.\mathbf{company}'(y) \wedge \mathbf{come}'(\langle \text{forall} \quad x \quad \mathbf{researcher}'(x) \wedge \mathbf{of}'(x, y)\rangle)$

(54) $\forall x.(\mathbf{researcher}'(x) \wedge \mathbf{of}'(x, y)) \rightarrow \exists y.\mathbf{company}'(y) \wedge \mathbf{come}'(x)$

But Keller (1988) shows that this strategy is not general enough: If there is a second instance of the variable that is not inside a term, as in the representation (56) for (55), the analogous step from (56) to (57) cannot be blocked, even though it would eventually lead to structure (58) where the variable y within the restriction of the universal quantifier is not bound:

(55) Every sister of [a boy]$_i$ hates him$_i$

(56) $\mathbf{hate}'(\langle \text{forall} \;\; x \;\; \textsf{\textbf{sister-of}}'(x, \langle \text{exists} \;\; y \;\; \mathbf{boy}'(y)\rangle)\rangle, y)$

(57) $\exists y.\mathbf{boy}(y) \land \mathbf{hate}'(\langle \text{forall} \;\; x \;\; \textsf{\textbf{sister-of}}'(x, y)\rangle, y)$

(58) $\forall x.\textsf{\textbf{sister-of}}'(x, y) \rightarrow \exists y.\mathbf{boy}(y) \land \mathbf{hate}'(x, y)$

A second way of handling nested quantification (Nerbonne 1993) is restricting the solutions of underspecified representations to *closed formulae* (without free variables), although free variables do not make formulae ill-formed.

While this approach does not run into problems with sentences such as (55), it is not too efficient, however, in that one has to perform resolution steps first before the result can be checked against the closedness requirement. Further disadvantages of this strategy are that it calls for an (otherwise redundant) bookkeeping of free variables (Nerbonne speaks of ‘overspecified’ representations) and that it bars the possibility of modelling the semantic contribution of non-anaphoric pronouns in terms of free variables.

Another formalism that belongs to this group is Ambiguous Predicate Logic (APL; Jaspars & van Eijck 1996). It describes scope underspecification in terms of so-called *formulae*, in which *contexts* (structured lists of scope-bearing operators) can be prefixed to expressions of predicate logic (or other formulae).

E.g., (59a) indicates that the existential quantifier has wide scope over the universal one, since they form one list element together, whereas negation, being another element of the same list, can take any scope w.r.t. the two quantifiers, viz., wide, intermediate, or narrow scope. In contrast, (59b) expresses that the scope of the existential quantifier and negation is open, and that the universal quantifier can have scope over or below (not between) the other operators, i.e., four scoping possibilities.

(59) (a) $(\exists x \Box \forall y \Box, \neg \Box) Rxy$

(b) $((\exists x \Box, \neg \Box) \Box, \forall y \Box) Rxy$

Explicit rewrite rules serve to derive the set of solutions from these formulae. In a formula $C(\alpha)$, one can either take any simple list element from the context C and apply it to α, or take the last part of a complex list element, e.g., $\forall y \Box$ from $\exists x \Box \forall y \Box$ in (59a). This would map (59a) onto (60a), which can then be rewritten as (60c) with the intermediate step (60b):

(60) (a) $(\exists x \Box, \neg \Box) \forall y.Rxy$

(b) $(\exists x \Box) \neg \forall y.Rxy$

(c) $\exists x. \neg \forall y.Rxy$

In sum, the underspecification formalisms expounded in this subsection give initial underspecified representations for ambiguous expressions that do not by themselves delimit the range of intended representations fully, this delimitation is the joint effect of the initial representations and the resolution algorithm.

The difference between underspecification formalisms that describe the readings of an ambiguous expression and those that derive these readings is thus not the existence of a suitable algorithm to enumerate the readings (see section

6. for such algorithms for descriptive underspecification formalisms), but the question of whether such an algorithm is essential in determining the set of solutions.

3.3 Levels of representation

In the previous sections, underspecification formalisms were introduced as distinguishing a meta and an object level of representation. This holds good for the majority of such formalisms, but in other ones both the underspecified and the fully specified representations are expressions of the same kind (what Cimiano & Reyle 2005 call 'representational' as opposed to 'descriptive' approaches).

UDRT (Reyle 1993, 1996) is a prime example of such a formalism. UDRT separates information on the ingredients of a semantic representation (DRS fragments) from information on the way that these fragments are combined. Consider e.g. (61) and its representation in (62):

(61) Everybody didn't pay attention

(62) $$\langle l_\top : \langle l_1 : \boxed{\begin{array}{|l|}\hline \text{x} \\ \hline \text{human(x)} \\ \hline \end{array} \Rightarrow \boxed{}}, l_2 : \boxed{\neg \boxed{}}, l_3 : \boxed{\text{x pay attention}} \rangle, \text{ORD} \rangle$$

In prose: The whole structure (represented by the label $l_\top$) consists of a set of labelled DRS fragments (for the semantic contributions of DP, negation, and VP, respectively) that are ordered in a way indicated by a relation ORD.

For an underspecified representation of the two readings of (61), the scope relations between l_1 and l_2 are left open in ORD:

(63) ORD = $\langle l_\top \geq l_1, l_\top \geq l_2, \mathit{scope}(l_1) \geq l_3, \mathit{scope}(l_2) \geq l_3 \rangle$.

Here '$\geq$' means 'has scope over', and *scope* maps a DRS fragment onto the empty DRS box it contains. Fully specified representations for the readings of (61) can then also be expressed in terms of (62). In these cases, ORD comprises in addition to the items in (63) a relation to determine the scope between universal quantifier and negation, e.g., $\mathit{scope}(l_1) \geq l_2$ for the reading with wide scope of the universal quantifier.

Another instance of such a 'monostratal' underspecification formalism is the (revised) Quasi-Logical Form (QLF) of Alshawi & Crouch (1992), which uses list-valued meta-variables in semantic representations whose specification indicates quantifier scope. The simplified representation for (27) in (33a) illustrated this point, the only difference between a scopally underspecified representation and one of its scopally specified solutions is the instantiation of a variable with an ordered list of (bound variables of) scope-bearing elements.

Kempson & Cormack (1981) also assume a single level of semantic representation (higher-order predicate logic) for quantifier scope ambiguities. Their example is (64), and its underspecified representation merely states the existence of a set of two examiners and one of six scripts, such that each of the researchers marks one of the scripts, and each script is marked by one of the researchers:

(64) Two examiners marked six scripts

This weak representation then is entailed by all the readings of (64), e.g., the one that each researcher marked each script, or the one that each researcher marked six different scripts. For cases in which a fully specified reading is entailed by the other(s), like in the case of (27), this weakest reading is taken as semantic representation.

3.4 Compositionality

Another distinction between underspecification formalisms centres upon the notion of *resource*: In most underspecification formalisms, the elements of a constraint show up at least once in all its solutions, in fact, exactly once, except in special cases like ellipses. This holds e.g. in UDRT, where constraints and their solutions share the same set of DRS fragments, in CLLS (Egg, Koller & Niehren 2001), where the relation between constraints and their solutions is defined as an assignment function from node variables (in constraints) to nodes (in the solutions), or in Glue Language Semantics, where this resource-sensitivity is explicitly encoded in the semantic representations (expressions of linear logic).

One of the consequences of this resource-sensitivity is that every solution of an underspecified semantic representation of a linguistic expression preserves the semantic contributions of the parts of the expression. If different parts happen to introduce instances of the same semantic material, then each instance must show up in each solution.

E.g., any solution to a constraint for (65a) must comprise two universal quantifiers. The contributions of the two DPs may not be conflated in the solution, which directly rules out that (65a) and (65b) could share a reading 'for every person: he likes himself':

(65) (a) Everyone likes everyone
     (b) Everyone likes himself

While this strategy seems natural in that the difference between (65a) and (65b) need not be stipulated by additional mechanisms, there are cases where different instances of the same semantic material seem to merge in the solutions.

Reconsider e.g. the case of Afrikaans past tense marking (66) [= (18)] in Sailer (2004). This example has two tense markers and three readings. Sailer points out that the two instances of the past tense marker seem to merge in the first and the second reading of (66):

(66) Jan wou gebel het
Jan want.PAST called have
'Jan wanted to call/Jan wants to have called/Jan wanted to have called'

A direct formalisation of this intuition is possible if one relates fragments in terms of *subexpressionhood*, as in the underspecified analyses in the LRS framework (Richter & Sailer 2006; see also the discussion in Kallmeyer & Richter 2006). If constraints introduce identical fragments as subexpressions of a larger fragment, these fragments can but need not coincide in the solutions of the constraints.

For the readings of (18), the constraint (simplified) is (67a):

(67) (a) $\langle [\mathrm{PAST}(\gamma)]_{\beta}, [\mathrm{PAST}(\zeta)]_{\epsilon}, [\mathbf{want'}(\mathbf{j}, {}^{\wedge}\eta)]_{\theta}, [\mathbf{call'}(\mathbf{j})]_{\iota}, \beta \triangleleft \alpha, \epsilon \triangleleft \delta, \theta \triangleleft \delta, \iota \triangleleft \gamma, \iota \triangleleft \zeta, \iota \triangleleft \eta \rangle$

(b) $\text{PAST}(\mathbf{want}'(\mathbf{j}, {}^{\wedge}(\mathbf{call}'(\mathbf{j}))))$

In prose: The two PAST- and the *want*-fragments are subexpressions of (relation '$\triangleleft$') the whole expression (as represented by the variables α or δ), while the *call*-fragment is a subexpression of the arguments of the PAST operators and the intensionalised second argument of *want*. This constraint describes all three semantic representations in (19); e.g., to derive (67b) [= (19b)], the following equations are needed: $\alpha = \delta = \beta = \epsilon$, $\gamma = \zeta = \theta$, and $\eta = \iota$. The crucial equation here is $\beta = \epsilon$, which equates two fragments (not a fragment and a variable or two variables). (Additional machinery is needed to block unwanted readings where both PAST operators show up outside or inside the scope of *want*. See Sailer 2004 for details.)

This approach is more powerful than resource-sensitive formalisms. The price one has to pay for this additional power is the need to control explicitly whether identical material may or may not coincide (see e.g. the analyses in Richter & Sailer 2006 on negative concord).

3.5 Expressivity and compactness

The standard approach to evaluate an underspecification formalisms is to apply it to challenging ambiguous examples and to check whether there is an expression of the formalism that can express *all* and *only* the attested readings of the example. As expounded in section 3.1, examples like (68) [= (41)] serve as benchmark tests, and any reasonable underspecification formalism must provide an expression that excompasses exactly the five representations of the example.

(68) Every researcher of a company saw most samples

However, what if these readings are contextually restricted, or, if the sentence has only four readings, as claimed by Kallmeyer & Romero 2008) and others, lacking the reading (46b) with the scope ordering $\exists > \mathbf{most}' > \forall$?

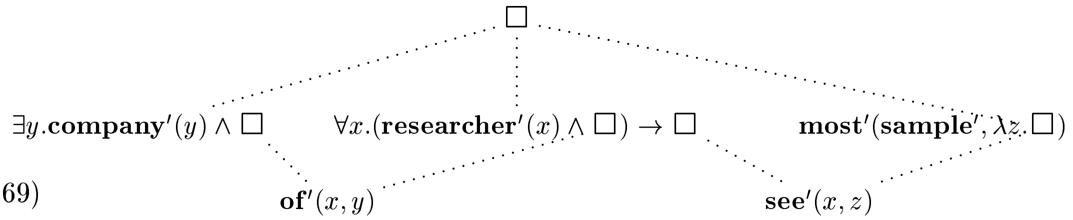
Underspecification approaches that model scope in terms of partial order between fragments of semantic representations run into problems already with the second of these possibilities: Any constraint set that encompasses the four readings in which $\mathbf{most}'$ has highest or lowest scope also covers the fifth reading (46b) (Ebert 2005). This means that such underspecification formalisms are not expressive in the sense of König & Reyle (1999) or (Ebert 2005), since they cannot represent *any* subset of readings of an ambiguous expression.

The formalisms are of different expressivity, e.g., approaches that model quantifier scope by lists (such as Alshawi 1992) are less expressive than those that use dominance relations, or scope lists together with an explicit ordering of list elements as in Fox & Lappin's (2005) *Property Theory with Curry Typing.*

Fully expressive is the approach of Koller, Regneri & Thater (2008), which uses *Regular Tree Grammars* for scope underspecification. Rules of these grammars expand nonterminals into tree fragments. E.g., the rule $S \rightarrow f(A, B)$ expands S into a tree whose mother is labelled by f, and whose children are the subtrees to be derived by expanding the nonterminals A and B.

Koller, Regneri & Thater (2008) show that dominance constraints can be translated into RTGs, e.g., the constraint (69) [= (42)] for the semantics of (41) is translated into (70).

(69)



(70)

$$\begin{array}{lcl|lcl}
\{1\text{-}5\} & \to & \exists comp(\{2\text{-}5\}) & \{1\text{-}4\} & \to & \exists comp(\{1\},\{2\text{-}4\}) \\
\{1\text{-}5\} & \to & \forall res(\{1\text{-}2\},\{4\text{-}5\}) & \{1\text{-}2\} & \to & \exists comp(\{2\}) \\
\{1\text{-}5\} & \to & most(\{1\text{-}4\}) & \{2\text{-}4\} & \to & \forall res(\{2\},\{3\}) \\
\{2\text{-}5\} & \to & \forall res(\{2\},\{4\text{-}5\}) & \{4\text{-}5\} & \to & most(\{4\}) \\
\{2\text{-}5\} & \to & most(\{2\text{-}4\}) & \{2\} & \to & of \\
\{1\text{-}4\} & \to & \forall(\{1\text{-}2\},\{4\}) & \{4\} & \to & see
\end{array}$$

In (70), the fragments of (69) are addressed by numbers, 1, 3, and 5 are the fragments for indefinite, definite, and *most*-DP, respectively, and 2 and 4 are the fragments for *of* and *see*. All nonterminals correspond to parts of constraints; they are abbreviated as sequences of fragments. E.g., {2-5} corresponds to the whole constraint except the existential fragment.

Rules of the RTG specify on the right hand side the root of the partial constraint introduced on the left hand side, for instance, the first rule expresses wide scope of *a company* over the whole sentence. The RTG (70) yields the same five solutions as (69).

In (70), the reading $\exists > \textbf{most}' > \forall$ can be excluded easily, by removing the production rule $\{2\text{-}5\} \to most(\{2\text{-}4\})$: This leaves only one expansion rule for {2-5}. Since {2-5} emerges only as child of $\exists comp$ with widest scope, only $\forall res$

can be the root of the tree below widest-scope $\exists comp$. This shows that RTGs are more expressive than dominance constraints or a variant thereof.

In more involved cases, restricting the set of solutions is less simple: One must sometimes distinguish different versions of the same partial constraint with respect to their derivation history, which must then be expanded separately by different rules. (E.g., even in the simple example (70), {1-2} can be derived in two different ways, as it appears on the right of two production rules.) But this means that RTGs usually get larger if one wants to exclude specific solutions.

This last observation points to another property of underspecification formalisms that is interdependent with expressivity, viz., *compactness*: A (sometimes tacit) assumption is that underspecification formalisms should be able to characterise a set of readings of an ambiguous expression in terms of a representation that is shorter or more efficient than an enumeration (or disjunction) of all the readings (König & Reyle 1999). Ebert (2005) defines this intuitive notion of compactness in the following way: An underspecification formalism is compact iff the maximal length of the representations is at most polynomial (with respect to the number of scope-bearing elements).

Ebert shows that there is a trade-off between expressivity and compactness, and that no underspecification formalism can be both expressive and compact in his sense at the same time.

4. Motivation

This section outlines a number of motivations for the introduction and use of semantic underspecification formalisms.

4.1 Functionality of the syntax-semantics interface

The first motivation for semantic underspecification formalisms lies in the syntax-semantics interface: Semantic underspecification is one way of keeping the mapping from syntax to semantics *functional* in spite of semantically and syntactically homogeneous ambiguities like (27). These expressions can be analysed in terms of a single syntactic structure even though they have several readings. This seems in conflict with the functional nature of semantic interpretation, which associates one specific syntactic structure with only one single semantic structure (see Westerståhl 1998 and Hodges 2001).

Competing approaches to the syntax-semantics interface either multiply syntactic structures for semantically and syntactically homogeneous ambiguities (one for each reading) of relinquish the functionality of the syntax-semantics interface altogether to accommodate these ambiguities.

4.1.1 Multiplying syntactic structures

Syntactic structures can be multiplied in two ways. First, one can postulate the functional relation between syntactic *derivation trees* (a syntactic structure and its derivation history) and semantic structures rather than between syntactic and semantic structures. This strategy shows up in Montague's (1974) account of quantifier scope ambiguity and in approaches like Hoeksema (1985). This strategy is motivated by the definition of semantic interpretation as a homomorphism from the syntactic to the semantic algebra (every syntactic operation is translated into a semantic one), but demotes the semantic structure that results from this derivation by giving the pride of place to the derivation itself.

Second, one can model the ambiguous expressions as syntactically *heterogeneous*. This means that each reading corresponds to a unique syntactic structure (on a semantically relevant syntactic level). Syntactic heterogeneity can then emerge either through different ways of combining the parts of the expression (which themselves need not be ambiguous), through systematic lexical ambiguity of specific parts of the expression which enforces different ways of combining them syntactically, or through systematic lexical ambiguity of parts of the expression which are nevertheless combined uniformly.

The first way of making the relevant expressions syntactically heterogeneous is implemented in *Generative Grammar*. Here syntactic structures unique to specific readings show up on the level of *Logical Form* (LF). For instance, quantifier scope is be determined by (covert) DP movement and adjunction (mostly, to a suitable S node); relative scope between quantifiers can then be put down to relations of c-command between the respective DPs on LF (Heim & Kratzer 1998). (The standard definition of c-command is that a constituent A c-commands another constituent B if A does not dominate B and vice versa and the lowest branching node that dominates A also dominates B.)

The second way of inducing syntactic heterogeneity is to assume that specific lexical items are ambiguous because they occur in different syntactic categories.

This means that depending on their reading they combine with other constituents in different ways syntactically. E.g., *Combinatory Categorial Grammar* (CCG) incorporates rules of *type raising*, which change the syntactic category and hence also the combinatory potential of lexical items. For instance, an expression of category X can become one of type $T/(T\backslash X)$, i.e., a T which lacks to its right a T lacking an X to its left. If $X = DP$ and $T = S$, a DP becomes a sentence without a following VP, since the VP is a sentence without a preceding DP ($S\backslash DP$).

Hendriks (1993) and Steedman (2000) point out that these rules could be used for modelling quantifier scope ambiguities in terms of syntactically heterogeneous ambiguity: Syntactic type raising changes the syntactic combinatory potential of the involved expressions, which may change the order in which the expressions are combined in the syntactic construction. This in turn affects the order of combining elements in semantic construction. In particular, if a DP is integrated later than another one (DP$'$), then DP gets wide scope over DP$'$: The semantics of DP is applied to a semantic representation that already comprises the semantic contribution of DP$'$.

In an example such as (27), the two readings could thus emerge by either first forming a VP and then combining it with the subject (wide scope for the subject), or by forming a constituent out of subject and verb, which is then combined with the object (which consequently gets widest scope).

Finally, syntactic heterogeneity can be due to lexical ambiguity that does not affect the syntactic combinatory potential of the involved expressions. This approach is instantiated by Hendriks's (1993) *Flexible Montague Grammar* and

Sailer's (2000) *Lexicalized Flexible Ty2*. These approaches want to retain the semantic flexibility of interpretation without making it dependent on syntactic flexibility. The basic idea is that specific constituents (in particular, verbs and their arguments) have an (in principle unlimited yet systematically related) set of interpretations. This ambiguity can be inherited by expressions that these constituents are part of, but this does not influence the constituent structure of the expression, because all readings of these constituents are of the same syntactic category.

Every lexical entry is given a maximally simple interpretation, which can then be changed by general rules such as *Argument Raising* (AR). E.g., *love* would (in an extensional framework) be introduced as a relation between two arguments, and twofold application of AR can return the λ-terms in (71), whose difference is due to the different order of applying AR to the arguments:

(71) (a) $\lambda Y \lambda X.X(\lambda x.Y(\lambda y.\mathbf{love}'(x,y)))$

(b) $\lambda Y \lambda X.Y(\lambda y.X(\lambda x.\mathbf{love}'(x,y)))$

Applying these λ-terms to the semantic representations of *a man* and *every woman* (in this order, which follows the syntactic structure in (2)) then returns the two semantic representations in (28).

4.1.2 Giving up functionality of the syntax-semantics interface

Other researchers reject the semantically motivated multiplication of syntactic structures for the relevant ambiguous syntactic expressions and give up the functionality of the syntax-semantics interface instead. One syntactic structure may thus correspond to several readings, which is due to a less strict coupling of syntactic and semantic construction rules.

This strategy is implemented in *Cooper store* approaches (Cooper 1983), where specific syntactic operations are coupled to more than one corresponding semantic operation in the syntax-semantics interface. In particular, the syntactic combination of a DP with a syntactic structure S may lead to the immediate combination of the semantic contributions of both DP and S or to appending the DP semantics to a list of DP interpretations (the 'store'). Subsequently, material can be retrieved from the store for any sentence constituent, which is then combined with the semantic representation of the sentence constituent. This gives the desired flexibility to derive scopally different semantic representations like in (28) from uniform syntactic structures like (2). The approach of Woods (1967, 1978) works in a similar fashion: Semantic contributions of DPs are collected separately from the main semantic representation; they can be combined with this main semantic representation immediately or later.

Another approach of this kind is Steedman (2007). Here non-universal quantifiers and their scope with respect to universal quantifiers are modelled in terms of Skolem functions. (See Kallmeyer & Romero 2008 for further discussion of this strategy.) These functions can have arguments for variables bound by universal quantifiers to express the fact that they are outscoped by these quantifiers. Consider e.g. the two readings of (27) in Skolem notation:

(72) (a) $\forall x.\mathbf{woman}'(x) \rightarrow \mathbf{man}'(sk_1) \wedge \mathbf{love}'(x, sk_1)$ ('one man for all women')

(b) $\forall x.\mathbf{woman'}(x) \rightarrow \mathbf{man'}(sk_2(x)) \wedge \mathbf{love'}(x, sk_2(x))$ ('a possibly different man per woman')

For the derivation of the different readings of a scopally underspecified expression, Steedman uses underspecified Skolem functions, which can be specified at any point in the derivation w.r.t. its *environment*, viz., the tuple of variables bound by universal quantifiers so far. For (27), the semantics of *a man* would be represented by $\lambda Q.Q(\mathbf{skolem'}(\mathbf{man'}))$, where $\mathbf{skolem'}$ is a function from properties P and environments E to generalised skolem terms like $f(E)$, where P holds of $f(E)$.

The term $\lambda Q.Q(\mathbf{skolem'}(\mathbf{man'}))$ can be specified at different steps in the derivation, with different results: Immediately after the DP has been formed specification returns a Skolem constant like sk_1 in (72a), because the environment is still empty. After combining the semantics of the DPs and the verb, the environment is the 1-tuple comprising the variable x bound by the universal quantifier from the subject DP, hence, specification at that point yields a skolem term like $sk_2(x)$.

This sketch of competing approaches to the syntax-semantics interface shows that the functionality of this interface (or, an attempt to uphold it in spite of semantically and syntactically homogeneous ambiguous expressions) can be a motivation for underspecification: Functionality is preserved for such an expression directly in that there is a function from its syntactic structure to its underspecified semantic representation that encompasses all its readings.

4.2 Ambiguity and negation

Semantic underspecification also helps avoiding problems with disjunctive representations of the meaning of ambiguous expressions that show up under *negation*: Negating an ambiguous expression is intuitively interpreted as the disjunction of the negated expressions, i.e., one of the readings of the expressions is denied. However, if the meaning of the expression itself is modelled as the disjunction of its readings, the negated expression emerges as the negation of the disjunctions, which is equivalent to the *conjunction* of the negated readings, i.e., every reading of the expression is denied, which runs counter to intuitions.

E.g., for (27) such a semantic representation can be abbreviated as (73), which turns into (74) after negation:

(73) $\forall\exists \vee \exists\forall$

(74) $\neg(\forall\exists \vee \exists\forall) = \neg\forall\exists \wedge \neg\exists\forall$

However, if we model the meaning of the ambiguous expression as the set of its fully specified readings, and assume that understanding such an expression proceeds by forming the disjunction of this set, these interpretations follow directly. For (27), the meaning is thus $\{\forall\exists, \exists\forall\}$. The assertion of (1) is understood as the disjunction of its readings $\{\forall\exists, \exists\forall\}$; its denial, as the disjunction of its readings $\{\neg\forall\exists, \neg\exists\forall\}$, which yields the desired interpretation (van Eijck & Pinkal 1996).

For examples more involved than (27), the most efficient strategy of describing these set of readings would then be to describe their elements rather than to enumerate them, which then calls for underspecification.

4.3 Underspecification in Natural Langugage Processing

One of the strongest motivations for semantic underspecification was its attractiveness for Natural Language Processing (NLP).

The first issue for which underspecification is very useful is the fact that simple expository examples like (27) hide the fact that scope ambiguity resolution can be really hard - even for human analysts, and thus even more for NLP systems. E.g., in a small corpus study on quantifier scope in the CHORUS project at the University of the Saarland (using the NEGRA corpus; Brants, Skut & Uszkoreit 2003), roughly 10% of the sentences with more than one scope-bearing element were problematic, e.g., the slightly simplified (75):

(75) Alle Teilnehmer erhalten ein Handbuch
all participants receive a handbook
'All participants receive a handbook'

The interpretation of (75) is that the same kind of handbook is given to every participant, but that everyone gets his own copy. I.e., the scope between the DPs interacts with a type-token ambiguity: an existential quantification over handbook types outscopes the universal quantification over participants, which in turn gets scope over an existential quantification over handbook tokens.

For those examples, underspecification is useful to allow a semantic representation for NLP systems at all, because it does not force the system to make arbitrary choices and nevertheless returns a semantic analysis of the examples.

But the utility of underspecification for NLP is usually discussed with reference to *efficiency*, because this technique allows one to evade the problem of *combinatorial explosion* (Poesio 1996; Ebert 2005). The problem is that in many cases, the number of readings of an ambiguous expression gets too large to be generated and enumerated, let alone to be handled efficiently in further modules of an NLP system (e.g., for Machine Translation). This argumentation needs a slight modification, however: Player (2004) points out that ambiguity would not be a problem if there were systems that could derive the respective *preferred* reading with sufficient accuracy.

Deriving an underspecified representation of an ambiguous expression that captures only the common ground between its readings and fully deriving a reading only by need is less costly than generating all possible interpretations and then selecting the relevant one.

What is more, there are cases in which a complete disambiguation is not even *necessary*. In these cases, postponing ambiguity resolution, and resolving ambiguity only on demand makes NLP systems more efficient. E.g., scope ambiguities are in many cases irrelevant for translation, therefore it would be a waste of time to try and find the intended reading of a scopally ambiguous expression: After all, its translation into the target language would be ambiguous in the same way again. This was the reason why for instance the Verbmobil project (machine translation of spontaneous spoken dialogue; Wahlster 2000) used a scopally underspecified semantic representation (Schiehlen 2000).

That combinatorial explosion is indeed a problem for NLP that suggests the use of underspecification (pace Player 2004) becomes evident if one looks at the analyses of concrete NLP systems. The large number of readings that are attributed to linguistic expressions have to do with the fact that, first, the number of scope-bearing constituents per expression is underestimated (there are many more such constituents in addition to DPs, e.g., negation, modal verbs, quantifying adverbials like *three times* or *again*), and, second and much worse, there is the problem of *spurious ambiguities* that come in during syntactic and semantic analysis of the expressions.

Koller, Regneri & Thater (2008) investigated the Rondane Treebank (underspecified representations of sentences from the domain of Norwegian tourist information in MRS, distributed as part of the English Resource Grammar, Copestake & Flickinger 2000) and found that 5% of the representations in this treebank have more than 650 000 solutions, record holder is the (rather innocuous looking) sentence (76) with about 4.5×10^{12} scope readings:

(76) Myrdal is the mountain terminus of the Flåm rail line (or Flåmsbana)
which makes its way down the lovely Flåm Valley (Flåmsdalen) to its
sea-level terminus at Flåm.

The median number of scope readings per sentence is 56 (Koller, Regneri & Thater 2008), so, short of applying specific measures to eliminate spurious ambiguities (see section 6.2), combinatorial explosion definitely is a problem for semantic analysis in NLP.

In recent years, underspecification has turned out to very useful for NLP in another way, viz., in that underspecified semantics emerges as an *interface* bridging the gap between deep and shallow processing. To combine the advantages of both kinds of processing (accuracy vs. robustness and speed), both can be combined in NLP applications (*hybrid processing*). The results of deep and shallow syntactic processing can straightforwardly be integrated on the semantic level (instead of combining the results of deep and shallow syntactic analyses). An example for an architecture for hybrid processing is the 'Heart of Gold' developed in the project 'DeepThought' (Callmeier et al. 2004).

Since shallow syntactic analyses provide only a part of the information to be gained from deep analysis, the semantic information derivable from the results of a shallow parse (e.g., by a part-of-speech tagger or an NP chunker) can only be a part of the one derived from the results of a deep parse. Underspecification formalism can be used to model this kind of partial information as well.

For instance, deep and shallow processing may yield different results with respect to argument linking: NP chunkers (as opposed to systems of deep processing) do not relate verbs and their syntactic arguments, e.g., experiencer and patient in (77). Any semantic analysis based on such a chunker will thus fail to identify individuals in NP and verb semantics as in (78):

(77) Max saw Mary

(78) named(x_1, Max), see(x_2, x_3), named(x_4, Mary)

Semantic representations of different depths must be compatible in order to combine results from parallel deep and shallow processing or to transform shallow into deep semantic analyses by adding further pieces of information. Thus, the semantic representation formalism must be capable of separating the semantic information from different sources appropriately. E.g., information on argument linking should be listed separately, thus, a full semantic analysis of (77) should look like (79) rather than (80). Robust MRS (Copestake 2003) is an underspecification formalism that was designed to fulfill this demand:

(79) named(x_1, Max), see(x_2, x_3), named(x_4, Mary), $x_1 = x_2$, $x_3 = x_4$

(80) named(x_1, Max), see(x_1, x_4), named(x_4, Mary)

4.4 Semantic construction

Finally, underspecification formalisms turn out to be interesting from the perspective of semantic construction in general, independently of the issue of ambiguity. This interest is based on two properties of these formalisms, viz., their *portability* and their *flexibility*.

First, underspecification formalisms do not presuppose a specific syntactic analysis (which would do a certain amount of preprocessing for the mapping from syntax to semantics, like the mapping from surface structure to Logical Form in Generative Grammar). Therefore the syntax-semantics interface can be defined in a very transparent fashion, which makes the formalisms very *portable* in that they can be coupled with different syntactic formalisms. Fig. 1 lists some of the realised combinations of syntactic and semantic formalisms:

Second, the flexibility of the interfaces that are needed to derive underspecified representations of ambiguous expressions is also available for unambiguous cases that pose a challenge for any syntax-semantics interface. E.g., semantic construction for the modification of modifiers and indefinite pronouns like *everyone* is a problem, because the types of functor (semantics of the modifier)

	HPSG	LFG	(L)TAG
MRS	Copestake et al. (2005)	Oepen et al. (2004)	Kallmeyer and Joshi (1999)
GLS	Asudeh and Crouch (2001)	Dalrymple (2001)	Frank and van Genabith (2001)
UDRT	Frank and Reyle (1995)	van Genabith and Crouch (1999)	Cimiano and Reyle (2005)
HS	Chaves (2002)		Kallmeyer and Joshi (2003)

Figure 1: Realised couplings of underspecification formalisms and syntax formalisms

and argument (semantics of the modified expression) do not fit: The PP semantics is a function from properties to properties, the semantics of the pronoun as well as the one of the whole modification structure are sets of properties.

(81) everyone in this room

Interfaces for the derivation of underspecified semantic representations for examples like (27) can be reused to perform this semantic construction, see Egg 2004 and Egg 2006) for the semantic construction of (81) and of many more examples of that kind. Similarly, Richter & Sailer (2006) use their underspecification formalism to handle semantic construction for unambiguous cases of negative concord.

The analyses of Richter and Sailer and of Egg highlight the fact that for these unambiguous expressions, the use of underspecification formalisms requires a careful control of the solutions of the resulting constraints: These constraints must have a single solution only (since the expressions are unambiguous), but underspecification constraints were designed primarily for the representation of ambiguous expressions, whose constraints have several solutions. Therefore, potential ambiguity must be blocked to avoid unwanted overgeneration.

5. Semantic underspecification and the syntax-semantics interface

In this section, I will sketch the basic interface strategy to derive underspecified semantic structures from (surface-oriented) syntactic structures. The strategy consists in deliberately not specifying scope relations between potentially scopally ambiguous constituents of an expression, e.g., in the syntax-semantics interfaces described for UDRT (Frank & Reyle 1995), MRS (Copestake et al. 2005), CLLS (Egg, Koller & Niehren 2001) or Hole Semantics (Bos 2004).

To derive underspecified semantic structures, explicit bookkeeping of specific parts of these structures is necessary. These parts have 'addresses' (e.g., the labels of UDRT or the handles of MRS) that are visible to the interface rules. This allows interface rules to address these parts in the subconstituents when they specify how the constraints of the subconstituents are to be combined in the constraints of the emerging new constituent. (The rules also specify these parts for the constraint of the new constituent.) Therefore, these interfaces are more powerful than interfaces that only combine the semantic contributions of the subconstituents as a whole.

As an example, consider the (greatly simplified) derivation of the underspecified representation (29) of example (27) by means of the syntax-semantics interface rules (82)-(84). In the interface, each atomic or complex constituent C is associated with a constraint and has two special fragments, a top fragment $[\![C_{top}]\!]$ (which handles scope issues) and a main fragment $[\![C]\!]$. These two fragments are addressed in the interface rules as 'glue points' where the constraints of the involved constituents are put together; each interface rule determines these fragments anew for the emerging constituent. Furthermore, all fragments of the subconstituents are inherited by the emerging constituent.

The first rule builds the DP semantics out of the semantic contributions of determiner and NP:

(82)
$$[_{\mathtt{DP}}\ \mathtt{Det}\ \mathtt{NP}] \overset{(\mathrm{SSI})}{\Rightarrow} \quad \begin{array}{rl} [\![\mathtt{DP}_{\mathtt{S}}]\!] : [\![\mathtt{Det}]\!]([\![\mathtt{NP}]\!])(\lambda z.\Box); & \qquad [\![\mathtt{DP}_{top}]\!] = [\![\mathtt{Det}_{top}]\!] = [\![\mathtt{NP}_{top}]\!] \\ \vdots & \\ [\![\mathtt{DP}]\!] : z & \end{array}$$

In prose: To obtain the secondary DP fragment, apply the main determiner fragment to the main NP fragment and a hole with a λ-abstraction over a variable that is dominated by the hole and constitutes by itself the main DP fragment. The top fragments (holes that determine the scope of the DP, because the top fragment of a constituent always dominates all its other fragments) of DP, determiner, and NP are identical. ('SSI' indicates that it is a rule of the syntax-semantics interface.)

The main fragment of a VP (of a sentence) emerges by applying the main verb (VP) fragment to the main fragment of the object (subject) DP. The top fragments of the verb (VP) and its DP argument are identical to the one of the emerging VP (S):

(83) $[_{\mathtt{VP}}\ \mathtt{V}\ \mathtt{DP}] \quad \overset{\text{(SSI)}}{\Rightarrow} \quad \llbracket\mathtt{VP}\rrbracket\text{: } \llbracket\mathtt{V}\rrbracket(\llbracket\mathtt{DP}\rrbracket); \qquad \llbracket\mathtt{VP}_{top}\rrbracket = \llbracket\mathtt{V}_{top}\rrbracket = \llbracket\mathtt{DP}_{top}\rrbracket$

(84) $[_{\mathtt{S}}\ \mathtt{DP}\ \mathtt{VP}] \quad \overset{\text{(SSI)}}{\Rightarrow} \quad \llbracket\mathtt{S}\rrbracket\text{: } \llbracket\mathtt{VP}\rrbracket(\llbracket\mathtt{DP}\rrbracket); \qquad \llbracket\mathtt{S}_{top}\rrbracket = \llbracket\mathtt{DP}_{top}\rrbracket = \llbracket\mathtt{VP}_{top}\rrbracket$

We assume that for all lexical entries, main and secondary fragments are identical to the standard semantic representation (e.g., for *every*, we get $\llbracket\mathtt{DP}\rrbracket$, $\llbracket\mathtt{DP_S}\rrbracket$: $\lambda Q \lambda P \forall x.Q(x \to P(x)))$, and that in unary projections like the one of *man* from N to $\bar{\mathrm{N}}$ and NP main and secondary fragments are merely inherited. Then the semantics of *a man* emerges as (85):

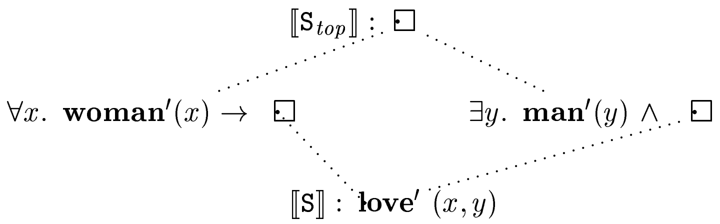
(85)
$$\begin{array}{lll}
\llbracket\mathtt{DP}_{top}\rrbracket : \square & & \\
\qquad\quad \vdots & & \\
\llbracket\mathtt{DP_S}\rrbracket : \exists y.\ \mathbf{man}'(y) \wedge \square & & \\
\qquad\qquad\qquad\qquad\quad \vdots & & \\
\qquad\qquad\quad \llbracket\mathtt{DP}\rrbracket : y & &
\end{array}$$

The crucial point is the decision to let the bound variable be the main fragment in the DP semantics. The intermediate DP fragment between top and main fragment is ignored in further processes of semantic construction. Combining (85) with the semantics of the verb yields (86):

(86)
$$\begin{array}{l}
\llbracket\mathtt{VP}_{top}\rrbracket : \square \\
\qquad\quad \vdots \\
\quad \exists y.\ \mathbf{man}'(y) \wedge \square \\
\qquad\qquad\qquad\quad \vdots \\
\qquad\quad \llbracket\mathtt{VP}\rrbracket : \mathbf{love}'(y)
\end{array}$$

Finally, the semantics of *every woman*, which is derived in analogy to (85), is combined with (86) through rule (84). According to this rule, the two top fragments are identified and the two main fragments are combined by functional application into the main S fragment, but the two intermediate fragments, which comprise the two quantifiers, are not addressed at all, and hence remain dangling in between. The result is the desired dominance diamond:

(87)



The technique of splitting the semantic contribution of a quantifying DP resurfaces in some way or other in many underspecification approaches, among them CLLS, Muskens, and LTAG (Cimiano & Reyle 2005).

6. Further processing of underspecified representations

So far, this article has focussed on the underspecified representations; the topic of this section is the derivation of fully specified semantic representations from underspecified representations. There are three main methods of doing this, one can either *enumerate* the set of solutions of a constraint or derive one solution (or a small set of solutions) in terms of *preferences*. The first enterprise has been the topic of much work in computational approaches to underspecification, the second one has been pursued both in computational linguistics and in psycholinguistics. Related to the enumeration of solutions is work on *redundancy elimination*, in which one tries to avoid enumerating more than one element of every set of equivalent readings. The third line of approach is the attempt to derive (fully specified) information from underspecified one by *reasoning* with underspecified representations.

6.1 Resolution of underspecified representations

The first way of deriving fully specified semantic representations from underspecified representations is to enumerate the readings by *resolving* the constraints. For a worked out example of such a resolution, reconsider the derivation of fully specified interpretations from the set of meaning constructors in Glue Language Semantics as expounded in section 3.1 or the detailed account of resolving USDL representations in Pinkal (1996).

For a number of formalisms, specific systems, so-called *solvers*, are available for this derivation. For MRS representations, there is a solver in the LKB (Linguistic Knowledge Builder) system (Copestake & Flickinger 2000). Blackburn & Bos (2005) present a solver for Hole Semantics. For the language of dominance constraints, a number of solvers have been developed (see Koller & Thater 2005 for an overview); the last and most efficient of these solvers (Koller, Regneri & Thater 2008) translates dominance constraints into Regular Tree Grammars (see section 3.5).

6.2 Redundancy elimination

In NLP applications that use underspecification, *spurious ambiguities* (which do not correspond to attested readings) are an additional complication, because they drastically enlarge the number of readings assigned to an ambiguous expression. E.g., Koller & Thater (2006) found high numbers of spurious ambiguities in the Rondane Treebank.

Hurum's (1988) algorithm, the CLE resolution algorithm (Moran 1988; Alshawi 1992), and Chaves's (2003) extension of Hole Semantics detect specific cases of equivalent solutions (e.g., when one existential quantifier immediately dominates another one) and block all but one of them. The blocking is only effective once the solutions are enumerated.

In contrast, Koller & Thater (2006) present an algorithm to reduce spurious ambiguities that maps underspecified representations on (more restricted) underspecified representations. For the Rondane Treebank, Koller & Thater (2006) found that their algorithm reduces the number of readings from an average of 56 to an average of 4 ambiguities. In the meantime, this algorithm is outperformed by far by the new redundancy elimination algorithm in the WTG approach to underspecification of Koller, Regneri & Thater (2008).

6.3 Reasoning with underspecified representations

Sometimes it is possible to deduct fully specified information from an underspecified semantic representation. E.g., if Amélie is a woman, then (27) allows us to conclude that she loves a man, because this conclusion is valid no matter which reading of (27) is chosen. For UDRT (König & Reyle 1999; Reyle 1992; Reyle 1993; Reyle 1996) and Ambiguous Predicate Logic (APL; Jaspars & van Eijck 1996), there are calculi for such reasoning with underspecified representations. van Deemter (1996) discusses different kinds of consequence relations for this reasoning.

As an example for reasoning with underspecified representations, consider Jaspars & van Eijck's (1996) proof of the above conclusion (here $\mathbf{woman}(x)$, $\mathbf{man}'(y)$, and $\mathbf{love}'(x,y)$ are abbreviated as Wx, My and Lxy, respectively;

see section 3.2 for further information on APL):

(88)

$$\dfrac{\dfrac{\exists y.My \wedge \forall x.Wx \rightarrow Lxy \vdash \forall x.Wx \rightarrow \exists y.My \wedge Lxy}{(\exists y.My \wedge \Box)\forall x.Wx \rightarrow Lxy \vdash \forall x.Wx \rightarrow \exists y.My \wedge Lxy} \qquad \dfrac{\forall x.Wx \rightarrow \exists y.My \wedge Lxy \vdash \forall x.Wx \rightarrow \exists y.My \wedge Lxy}{(\forall x.Wx \rightarrow \Box)\exists y.My \wedge Lxy \vdash \forall x.Wx \rightarrow \exists y.My \wedge Lxy}}{(\exists y.My \wedge \Box, \forall x.Wx \rightarrow \Box)Lxy \vdash \forall x.Wx \rightarrow \exists y.My \wedge Lxy}$$

The result on the bottom line of (88) can be paraphrased as: ‘if every woman loves a man, then every woman is involved in a love-relationship to some man or other’ (i.e., the underspecified representation entails the weaker $\forall\exists$-reading). This then allows the desired conclusion that Amélie loves a man.

The proof starts on the left upper line with the statement that the strong reading entails the weak one. From this one can deduce the claim that an underspecified representation with a single solution (the strong reading) entails this solution. The right upper line is a tautology (the weak reading entails itself), then it follows again that we can derive the statement that an underspecified representation with a single solution (the weak reading) entails this solution. The crucial step is the last one, it uses the intuition that if every possible disambiguation of an underspecified expression entails ϕ, then the underspecified expression itself entails ϕ. Here the underspecified expression is (89a), its two possible disambiguations are (89b) and (89c), and ϕ is (89d):

(89) (a) $(\exists y.My \wedge \Box, \forall x.Wx \rightarrow \Box)Lxy$

(b) $(\exists y.My \wedge \Box)\forall x.Wx \rightarrow Lxy$

(c) $(\forall x.Wx \rightarrow \Box)\exists y.My \wedge Lxy$

(d) $\forall x.Wx \rightarrow \exists y.My \wedge Lxy$

6.4 Integration of preferences

In many cases of scope ambiguity, the readings are not on a par in that some are more preferred than others. Consider e.g. a slight variation of (27), here the ∃∀-reading is preferred over the ∀∃-reading:

(90) A woman loves every man

One could integrate these preferences into underspecified representations of scopally ambiguous expressions to narrow down the number of its readings or to order the generation of solutions (Alshawi 1992).

6.4.1 Kinds of preferences

The preferences discussed in the literature can roughly be divided into three groups. The first group have to do with *syntactic structure*, starting with Johnson-Laird's (1969) and Lakoff's (1971) claim that surface linear order or precedence introduces a preference for wide scope of the preceding scope-bearing item. Others argue against this claim, e.g., Villalta (2003) presents experimental counterevidence (she concentrated on *wh*-elements and DPs that introduce universal quantification).

This linear preference can be interpreted in terms of a syntactic configuration such as c-command (e.g., VanLehn 1978), since in a right-branching binary phrase-marker preceding constituents c-command the following ones.

However, these preferences are not universally valid: Kurtzman & MacDonald (1993) report a clear preference for wide scope of the embedded DP in the case of nested quantification as in (91). Here the indefinite article precedes (and c-commands) the embedded DP, but the ∀∃-reading is nevertheless preferred:

(91) I met a researcher from every university

Hurum (1988) and VanLehn (1978) make the preference of scope-bearing items to take scope outside the constituent they are directly embedded in also dependent on the category of that constituent (e.g., much stronger for items inside PPs than items inside infinite clauses).

The scope preference algorithm of Gambäck & Bos (1998) give scope-bearing non-heads (complements and adjuncts) in binary-branching syntactic structures immediate scope over the respective head.

The second group of preferences focusses on grammatical functions and thematic roles. Functional hierarchies have been proposed that indicate preference to take wide scope in Ioup (1975) (92a) and VanLehn (1978) (92b):

(92) (a) topic > deep and surface subject > deep subject or surface subject > indirect object > prepositional object > direct object

(b) preposed PP, topic NP > subject > (complement in) sentential or adverbial PP > (complement in) verb phrase PP > object

While Ioup combines thematic and functional properties in her hierarchy (by including the notion of ‘deep subject’), Pafel (2005) distinguishes grammatical functions (only subject and sentenctial adverb) and thematic roles (strong and weak patienthood) explicitly.

There is a certain amount of overlap between structural preferences and the functional hierarchies, at least in a language like English: Here DPs higher on the functional hierarchy also tend to c-command DPs lower on the hierarchy, because they are more likely to surface as subjects.

The third group of preferences addresses the quantifiers (or, the determiners expressing them) themselves. Ioup (1975) and VanLehn (1978) introduce a hierarchy of determiners:

(93) each > every > a > all > most > many > several > some (plural) > a few

(Ioup claims that the size of the set specified by the quantifier determines the position of the corresponding determiner on this hierarchy. The indefinite determiner and *some* (singular) do not fit this claim and are therefore not included in the hierarchy, w.r.t. scope preference, they could be placed between *every* and *all*, however.)

CLE incorporates such preference rules, too (Moran 1988; Alshawi 1992), e.g., the rule that *each* outscopes other determiners, and that negation is outscoped by *some* and outscopes *every*.

Some of these preferences can be put down to a more general preference for logically weaker interpretations, in particular, the tendency of universal quantifiers to outscope existential ones (recall that the $\forall\exists$-reading of sentences like (27) is weaker than the $\exists\forall$-reading; VanLehn 1978; Moran 1988; Alshawi 1992). Similarly, scope of the negation above *every* and below *some* returns existential statements, which are weaker than the (unpreferred) alternative scopings (universal statements) in that they do not make a claim about the whole domain.

Pafel (2005) lists further properties, among them focus and discourse binding (whether a DP refers to an already established set of entities, as e.g. in *few of the books* as opposed to *few books*).

6.4.2 Interaction of preferences

It has been argued that the whole range of quantifier scope effects can only be accounted for in terms of an interaction of different principles.

Fodor (1982) and Hurum (1988) assume an interaction between linear precedence and the determiner hierarchy, which is corroborated by experimental results of Filik, Paterson & Liversedge (2004). They show that a conflict of these principles leads to longer reading times.

The results of Filik, Paterson & Liversedge (2004) are also compatible with the predictions of Ioup (1975), who puts down scoping preferences to an interaction of the functional and quantifier hierarchy. To get wider scope than another quantifier in the same sentence, it is important to score high on both hierarchies. Kurtzman & MacDonald (1993) present empirical evidence for this interaction. They point to a clear contrast between sentences like (94a) [=(27)] and their passive version (94b), where the clear preference of (94a) for the ∀∃-reading is no longer there:

(94) (a) Every woman loves a man

(b) A man is loved by every woman

If preferences were determined by a single principle, one would expect a preference for the passive version, too, either one for its (new) subject, or for the *by*-PP (the former demoted subject).

Kurtzman & MacDonald (1993) argue that the interaction of a syntax-oriented principle with the thematic role principle can account for these findings. The principles agree on the scope preference for the subject in the active sentence, but conflict in the case of the passive sentence, which consequently exhibits no clear-cut scope preference.

The interaction between linear ordering/thematic hierarchy and the position of the indefinite article w.r.t. the universally quantifying *every* and *each* on the quantifier hierarchy is explained by Fodor (1982) and Kurtzman & MacDonald (1993) in that it is easier to interpret indefinite DPs in terms of a single referent than in terms of several ones. The second, more complex interpretation must be motivated, e.g., in the context of an already processed universal quantifier, which suggests several entities, one for each of the entities over which the universal quantifier quantifies.

The most involved model of interacting preferences for quantifier scope is the one of Pafel (2005). He introduces no less than eight properties of quantifiers that are relevant for scope preferences, among them syntactic position, grammatical function, thematic role, discourse binding and focus. The scores for the different properties are added up for each quantifier, the properties carry weights that were determined empirically.

7. References

Alshawi, Hiyan (ed.) 1992. *The Core Language Engine*. Cambridge: MIT Press.

Alshawi, Hiyan & Richard Crouch 1992. Monotonic semantic interpretation. In:

Proceedings of the 30th ACL. 32–39.

Althaus, Ernst, Denys Duchier, Alexander Koller, Kurt Mehlhorn, Joachim Niehren & Sven Thiel 2001. An efficient algorithm for the configuration problem of dominance graphs. In: *Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms*. Washington, 815–824.

Asher, Nicholas & Tim Fernando 1999. Labeled representations, underspecification and disambiguation. In: Harry Bunt & Reinhard Muskens (eds.) *Computing Meaning*, Dordrecht: Kluwer, volume 1. 73–95.

Asher, Nicholas & Alex Lascarides 2003. *Logics of conversation*. Cambridge: Cambridge University Press.

Asudeh, Ash & Richard Crouch 2002. Glue Semantics for HPSG. In: Frank van Eynde, Lars Hellan & Dorothee Beermann (eds.) *Proceedings of the 8th International Conference on Head-Driven Phrase Structure Grammar*, Stanford: CSLI Publications. 1–19.

Babko-Malaya, Olga 2004. LTAG semantics of NP-coordination. In: *Proceedings of the Seventh International Workshop on Tree Adjoining Grammars and Related Formalisms*. Vancouver, 111–117.

Bierwisch, Manfred 1983. Semantische und konzeptionelle Repräsentation lexikalischer Einheiten. In: Rudolf Růžička & Wolfgang Motsch (eds.) *Untersuchungen zur Semantik*, Berlin: Akademie-Verlag. 61–99.

Bierwisch, Manfred 1988. On the grammar of local prepositions. In: Manfred Bierwisch, Wolfgang Motsch & Ilse Zimmermann (eds.) *Syntax, Semantik und das Lexikon*, Berlin: Akademie-Verlag. 1–65.

Bierwisch, Manfred & Ewald Lang (eds.) 1987. *Grammatische und konzeptuelle Aspekte von Dimensionsadjektiven*, volume 26/27 of *studia grammatica.* Berlin: Akademie-Verlag.

Blackburn, Patrick & Johan Bos 2005. *Representation and inference for natural language. A first course in Computational Semantics.* Stanford: CSLI Publications.

Bos, Johan 1996. Predicate logic unplugged. In: Paul Dekker & Martin Stokhof (eds.) *Proceedings of the 10th Amsterdam Colloquium.* ILLC, 133–142.

Bos, Johan 2004. Computational Semantics in Discourse: Underspecification, Resolution, and Inference. *Journal of Logic, Language and Information* 13, 139–157.

Brants, Thorsten, Wojciech Skut & Hans Uszkoreit 2003. Syntactic annotation of a German newspaper corpus. In: Anne Abeillé (ed.) *Treebanks: Building and Using Parsed Corpora*, Dordrecht: Kluwer, volume 20 of *Text, speech and language technology.* 73–88.

Bronnenberg, Wim, Harry Bunt, Jan Landsbergen, Remko Scha, Wijnand Schoenmakers & Eric van Utteren 1979. The question answering system PHLIQA1. In: Leonard Bolc (ed.) *Natural language question answering systems*, London: McMillan. 217–305.

Bunt, Harry 2007. Underspecification in semantic representations: Which technique for what purpose? In: Harry Bunt & Reinhard Muskens (eds.) *Computing Meaning*, Amsterdam: Springer, volume 3. 55–85.

Callmeier, Ulrich, Andreas Eisele, Ulrich Schäfer & Melanie Siegel 2004.

The DeepThought core architecture framework. In: Maria Teresa Lino, Maria Francisca Xavier, Fatima Ferreira, Rute Costa & Raquel Silva (eds.) *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC) 2004*. ELRA, Lisbon: European Language Resources Association, 1205–1208.

Chaves, Rui 2002. Principle-based DRTU for HPSG: a case study. In: *First International Workshop on Scalable Natural Language Understanding*. EML, Heidelberg.

Chaves, Rui 2003. Non-Redundant Scope Disambiguation in Underspecified Semantics. In: Balder ten Cate (ed.) *Proceedings of the 8th ESSLLI Student Session*. Vienna, 47–58.

Chaves, Rui 2005a. DRT and underspecification of plural ambiguities. In: *Proceedings of the 6th International Workshop in Computational Semantics*. Tilburg, 78–89.

Chaves, Rui 2005b. Underspecification and NP coordination in constraint-based grammar. In: Frank Richter & Manfred Sailer (eds.) *Proceedings of the Workshop on Empirical Challenges and Analytical Alternatives to Strict Compositionality*. Heriot-Watt University, Edinburgh, 38–58.

Cimiano, Philipp & Uwe Reyle 2005. Talking about trees, scope and concepts. In: *Proceedings of the 6th International Workshop in Computational Semantics*. Tilburg, 90–102.

Cooper, Robin 1983. *Quantification and syntactic theory*. Dordrecht: Reidel.

Copestake, Ann 2003. Report on the design of RMRS. Deliverable D1.1a;

Project DeepThought.

Copestake, Ann & Dan Flickinger 2000. An open-source grammar development environment and broad-coverage English grammar using HPSG. In: *Proceedings of the 2nd International Conference on Language Resources and Evaluation*. Athens, 591–600.

Copestake, Ann, Daniel Flickinger, Carl Pollard & Ivan Sag 2005. Minimal Recursion Semantics. An introduction. *Research on Language and Computation* 3, 281–332.

Crouch, Richard 1995. Ellipsis and quantification: a substitutional approach. In: *Proceedings of Proceedings of the 7th conference on European chapter of the Association for Computational Linguistics*. 229–236.

Crouch, Richard & Josef van Genabith 1999. Context change, underspecification and the structure of Glue Language derivations. In: Mary Dalrymple (ed.) *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach*, Cambridge: MIT Press. 117–189.

Dalrymple, Mary 2001. *Lexical Functional Grammar*. Number 34 in Syntax and Semantics, New York: Academic Press.

Dalrymple, Mary, John Lamping, Fernando Pereira & Vijay Saraswat 1997. Quantifiers, anaphora, and intensionality. *Journal of Logic, Language, and Information* 6, 219–273.

Dalrymple, Mary, Stuart Shieber & Fernando Pereira 1991. Ellipsis and higher-order unification. *Linguistics & Philosophy* 14, 399–452.

van Deemter, Kees 1996. Towards a logic of ambiguous expressions. In: Kees van Deemter & Stanley Peters (eds.) *Semantic ambiguity and underspecification*, Stanford: CSLI Publications. 203–237.

Dölling, Johannes 1995. Ontological domains, semantic sorts and systematic ambiguity. *International Journal of Human-Computer Studies* 43, 785–807.

Duchier, Denys & Claire Gardent 2001. Tree descriptions, constraints and incrementality. In: Harry Bunt, Reinhard Muskens & Elias Thijsse (eds.) *Computing Meaning, Volume 2*, Dordrecht: Kluwer. 205–227.

Ebert, Christian 2005. *Formal investigations of underspecified representations.* Ph.D. thesis, King's College, London.

Egg, Markus 2004. Mismatches at the syntax-semantics interface. In: Stefan Müller (ed.) *Proceedings of the 11th International Conference on Head-Driven Phrase Structure Grammar*, Stanford: CSLI Publications. 119–139.

Egg, Markus 2005. *Flexible semantics for reinterpretation phenomena.* Stanford: CSLI Publications.

Egg, Markus 2006. Anti-Ikonizität an der Syntax-Semantik-Schnittstelle. *Zeitschrift für Sprachwissenschaft* 25, 1–38.

Egg, Markus 2007. Against opacity. *Journal on Research on Language and Computation* 5, 435–455.

Egg, Markus & Michael Kohlhase 1997. Dynamic control of quantifier scope. In: Paul Dekker, Martin Stokhof & Yde Venema (eds.) *Proceedings of the 11th Amsterdam Colloquium.* ILLC, 109–114.

Egg, Markus, Alexander Koller & Joachim Niehren 2001. The Constraint Language for Lambda-Structures. *Journal of Logic, Language, and Information* 10, 457–485.

Egg, Markus & Gisela Redeker 2008. Underspecified discourse representation. In: Anton Benz & Peter Kühnlein (eds.) *Constraints in Discourse*, Amsterdam: Benjamins. 117–138.

van Eijck, Jan & Manfred Pinkal 1996. What do underspecified expressions mean? In: Robin Cooper et al. (eds.) *Building the framework*, FraCaS Deliverable D 15, section 10.1. 264–266.

Filik, Ruth, Kevin Paterson & Simon Liversedge 2004. Processing doubly quantified sentences: Evidence from eye movements. *Psychonomic Bulletin and Review* 11, 953–959.

Fodor, Janet 1982. The mental representation of quantifiers. In: Stanley Peters & Esa Saarinen (eds.) *Processes, beliefs and questions*, Dordrecht: Reidel. 129–164.

Fox, Chris & Shalom Lappin 2005. Underspecified Interpretations in a Curry-typed representation language. *Journal of Logic and Computation* 15, 131–143.

Frank, Anette & Josef van Genabith 2001. GlueTag. Linear Logic based semantics for LTAG – and what it teaches us about LFG and LTAG –. In: Miriam Butt & Tracy Holloway King (eds.) *The Proceedings of the LFG '01 Conference*. Stanford: CSLI Publications, 104–126.

Frank, Annette & Uwe Reyle 1995. Principle-based semantics for HPSG. In:

Proceedings of the 7th conference on European chapter of the Association for Computational Linguistics. Dublin, 9–16.

Fuchss, Ruth, Alexander Koller, Joachim Niehren & Stefan Thater 2004. Minimal Recursion Semantics as dominance constraints: translation, evaluation, and analysis. In: *Proceedings of the 42nd annual meeting of the Association for Computational Linguistics*. 247–254.

Gambäck, Björn & Johan Bos 1998. Semantic-head based resolution of scopal ambiguities. In: *In Proceedings of the 17th International Conference on Computational Linguistics and the 36th Annual Meeting of the Association for Computational Linguistics*. Montreal, 433–437.

Gawron, Jean & Stanley Peters 1990. *Anaphora and quantification in Situation Semantics*. Stanford: CSLI Publications.

van Genabith, Josef & Richard Crouch 1999. Dynamic and underspecified semantics for LFG. In: Mary Dalrymple (ed.) *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach*, Cambridge: MIT Press. 209–260.

Harris, John 2007. Representation. In: Paul de Lacey (ed.) *The Cambridge handbook of phonology*, Cambridge: Cambridge University Press. 119–137.

Heim, Irene & Angelika Kratzer 1998. *Semantics in Generative Grammar*. Malden: Blackwell.

Hendriks, Herman 1993. *Studied flexibility: categories and types in syntax and semantics*. ILLC dissertation series, Institute for Logic, Language and Computation, Universiteit van Amsterdam, Amsterdam.

Hirschbühler, Paul 1982. VP deletion and across the board quantifier scope. In: James Pustejovsky & Peter Sells (eds.) *Proceedings of the North Eastern Society of Linguistics 12*. 132–139.

Hobbs, Jerry & Stuart Shieber 1987. An algorithm for generating quantifier scoping. *Computational Linguistics* 13, 47–63.

Hobbs, Jerry, Mark Stickel, Douglas Appelt & Paul Martin 1993. Interpretation as abduction. *Artificial Intelligence* 63, 69–142.

Hodges, Wilfrid 2001. Formal features of compositionality. *Journal of Logic, Language, and Information* 10, 7–28.

Hoeksema, Jack 1985. *Categorial morphology.* New York: Garland Publishing.

Huet, Gérard P. 1975. A unification algorithm for the typed λ-calculus. *Theoretical Computer Science* 1, 27–57.

Hurum, Sven 1988. Handling scope ambiguities in English. In: *Proceedings of the second conference on Applied natural language processing.* Morristown: Association for Computational Linguistics, 58–65.

Ioup, Georgette 1975. Some universals for quantifier scope. In: John Kimball (ed.) *Syntax and Semantics*, New York: Academic Press, volume 4. 37–58.

Jaspars, Jan & Jan van Eijck 1996. Underspecification and reasoning. In: Robin Cooper et al. (eds.) *Building the framework*, FraCaS Deliverable D 15, section 10.2. 266–287.

Johnson-Laird, P. 1969. On understanding logically complex sentences. *Quarterly Journal of Experimental Psychology* 21, 1–13.

Kallmeyer, Laura & Aravind Joshi 2003. Factoring predicate argument and scope semantics: Underspecified semantics with LTAG. *Research on Language and Computation* 1, 3–58.

Kallmeyer, Laura & Frank Richter 2006. Constraint-based computational semantics: a comparison between LTAG and LRS. In: *Proceedings of the 8th International Workshop on Tree-Adjoining Grammar and Related Formalisms*. Sydney: Association for Computational Linguistics, 109–114.

Kallmeyer, Laura & Maribel Romero 2008. Scope and situation binding in LTAG using semantic unification. *Research on language and computation* 6, 3–52.

Keller, William 1988. Nested Cooper storage: the proper treatment of quantification in ordinary noun phrases. In: Uwe Reyle & Christian Rohrer (eds.) *Natural language parsing and linguistic theories*, Reidel. 432–447.

Kempson, Ruth & Annabel Cormack 1981. Ambiguity and quantification. *Linguistics & Philosophy* 4, 259–309.

Koller, Alexander, Joachim Niehren & Stefan Thater 2003. Bridging the gap between underspecification formalisms: hole semantics as dominance constraints. In: *Proceedings of the 11th EACL*. Budapest.

Koller, Alexander, Michaela Regneri & Stefan Thater 2008. Regular tree grammars as a formalism for scope underspecification. In: *Proceedings of ACL-08: HLT*. Columbus: Association for Computational Linguistics, 218–226.

Koller, Alexander & Stefan Thater 2005. The evolution of dominance constraint solvers. In: *Proceedings of the ACL-05 Workshop on Software*. Ann Arbor.

Koller, Alexander & Stefan Thater 2006. An improved redundancy elimination algorithm for underspecified descriptions. In: *Proceedings of COLING-ACL 2006*. Sydney.

König, Esther & Uwe Reyle 1999. A general reasoning scheme for underspecified representations. In: Hans Jürgen Ohlbach & Uwe Reyle (eds.) *Logic, Language and Reasoning. Essays in Honour of Dov Gabbay*, Dordrecht: Kluwer. 1–28.

Kratzer, Angelika 1998. Scope or pseudoscope? Are there wide-scope indefinites? In: Susan Rothstein (ed.) *Events and Grammar*, Dordrecht: Kluwer. 163–196.

Kurtzman, Howard & Maryellen MacDonald 1993. Resolution of quantifier scope ambiguities. *Cognition* 48, 243–279.

Lakoff, George 1971. On generative semantics. In: Danny Steinberg & Leon Jakobovits (eds.) *Semantics: An Interdisciplinary Reader in Philosophy, Linguistics and Psychology*, Cambridge: Cambridge University Press. 232–296.

Larson, Richard 1998. Events and modification in nominals. In: Devon Strolovitch & Aaron Lawson (eds.) *Proceedings from SALT VIII.* Ithaca: CLC Publications, 145–168.

Larson, Richard & Sungeun Cho 2003. Temporal adjectives and the structure of possessive DPs. *Natural Language Semantics* 11, 217–247.

Lev, Iddo 2005. Decoupling Scope Resolution from Semantic Composition. In: *Proceedings of IWCS 6*. Tilburg, 139–150.

Link, Godehard 1983. The logical analysis of plurals and mass terms: a lattice-theoretic approach. In: R. Bäuerle, Ch. Schwarze & v. Stechow. A. (eds.) *Meaning, use, and the interpretation of language*, Berlin: de Gruyter. 303–323.

Marcus, Mitchell, Donald Hindle & Margaret Fleck 1983. D-theory: talking about talking about trees. In: *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics*. Cambridge, MA: Association for Computational Linguistics, 129–136.

Montague, Richard 1974. *Formal philosophy: selected papers of Richard Montague*. New Haven: Yale University Press. Ed. by R. Thomason.

Moran, Douglas 1988. Quantifier scoping in the SRI core language engine. In: *Proceedings of the 26th ACL*. Buffalo, 33–40.

Muskens, Reinhard 2001. Talking about trees and truth-conditions. *Journal of Logic, Language, and Information* 10, 417–455.

Nerbonne, John 1993. A feature-based syntax/semantics interface. *Annals of Mathematics and Artificial Intelligence* 8, 107–132.

Niehren, Joachim, Manfred Pinkal & Peter Ruhrberg 1997. A uniform approach to underspecification and parallelism. In: *Proceedings of ACL'97*. Madrid, Spain, 410–417.

Oepen, Stephan, Helge Dyvik, Jan Tore Lønning, Erik Velldal, Dorothee Beermann, John Carroll, Dan Flickinger, Lars Hellan, Janne Bondi Johannessen, Paul Meurer, Torbjørn Nordgård & Victoria Rosén 2004. Som å kapp-ete med trollet? Towards MRS-based Norwegian-English Machine Translation.

In: *Proceedings of the 10th TMI*. Baltimore.

Pafel, Jürgen 2005. *Quantifier scope in German*. Amsterdam: Benjamins.

Park, Jong 1995. Quantifier scope and constituency. In: *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 05)*. 205–212.

Pinkal, Manfred 1995. *Logic and Lexicon. The semantics of the indefinite*. Dordrecht: Kluwer.

Pinkal, Manfred 1996. Radical underspecification. In: Paul Dekker & Martin Stokhof (eds.) *Proceedings of the 10th Amsterdam Colloquium*. Amsterdam: ILLC, 587–606.

Pinkal, Manfred 1999. On semantic underspecification. In: Harry Bunt & Reinhard Muskens (eds.) *Computing Meaning, Vol. 1*, Dordrecht: Kluwer. 33–55.

Player, Nickie 2004. *Logics of ambiguity*. Ph.D. thesis, University of Manchester.

Poesio, Massimo 1996. Semantic ambiguity and perceived ambiguity. In: Kees van Deemter & Stanley Peters (eds.) *Ambiguity and underspecification*, CSLI Publications. 159–201.

Poesio, Massimo, Patrick Sturt, Ron Artstein & Ruth Filik 2006. Underspecification and anaphora: Theoretical issues and preliminary evidence. *Discourse Processes* 42, 157–175.

Pulman, Steve 1997. Aspectual shift as type coercion. *Transactions of the Philo-*

logical Society 95, 279–317.

Rambow, Owen, David Weir & Vijay Shanker 2001. D-tree substitution grammars. *Computational Linguistics* 27, 89–121.

Rapp, Irene & Arnim von Stechow 1999. Fast "almost" and the visibility parameter for functional adverbs. *Journal of Semantics* 16, 149–204.

Regneri, Michaela, Markus Egg & Alexander Koller 2008. Efficient processing of underspecified discourse representations. In: *Proceedings of ACL-08: HLT, Short Papers.* Columbus: Association for Computational Linguistics, 245–248.

Reyle, Uwe 1992. Dealing with ambiguities by underspecification: a first order calculus for unscoped representations. In: Martin Stokhof & Paul Dekker (eds.) *Proceedings of the Eighth Amsterdam Colloquium.* 493–512.

Reyle, Uwe 1993. Dealing with ambiguities by underspecification: construction, representation, and deduction. *Journal of Semantics* 10, 123–179.

Reyle, Uwe 1996. Co-indexing labeled DRSs to represent and reason with ambiguities. In: Kees Deemter & Stanley Peters (eds.) *Semantic ambiguity and underspecification*, Stanford: CSLI. 239–268.

Richter, Frank & Manfred Sailer 1996. Syntax für eine unterspezifizierte Semantik: PP-Anbindung in einem deutschen HPSG-Fragment. In: Stephan Mehl, Andreas Mertens & Marion Schulz (eds.) *Präpositionalsemantik und PP-Anbindung*, Gerhard-Mercator-Universität Duisburg. 39–47.

Richter, Frank & Manfred Sailer 2006. Modeling typological markedness in semantics: the case of negative concord. In: Stefan Müller (ed.) *Proceedings of the HPSG 06 Conference*. Stanford: CSLI Publications, 305–325.

Robaldo, Livio 2007. *Dependency tree semantics.* Ph.D. thesis, University of Turin.

Sag, Ivan 1976. *Deletion and logical form.* Ph.D. thesis, MIT, Cambridge.

Sailer, Manfred 2000. *Combinatorial semantics and idiomatic expressions in Head-Driven Phrase Structure Grammar.* Ph.D. thesis, Universität Tübingen.

Sailer, Manfred 2004. Past tense marking in Afrikaans. In: Cecile Meier & Matthias Weisgerber (eds.) *Proceedings of Sinn und Bedeutung 8.* 233–248.

Sailer, Manfred 2006. *Don't believe* in underspecified semantics. In: Olivier Bonami & Patricia Cabredo Hofherr (eds.) *Empirical issues in syntax and semantics.* 375–403.

Schiehlen, Michael 2000. Semantic construction. In: Wolfgang Wahlster (ed.) *Verbmobil: Foundations of speech-to-speech translation*, Berlin: Springer. 200–216.

Schilder, Frank 2002. Robust discourse parsing via discourse markers, topicality and position. *Natural Language Engineering* 8, 235–255.

Schubert, Lenhart & Francis Pelletier 1982. From English to logic: Context-free computation of 'conventional' logical translation. *American Journal of Computational Linguistics* 8, 26–44.

Shieber, Stuart, Fernando Pereira & Mary Dalrymple 1996. Interaction of scope and ellipsis. *Linguistics & Philosophy* 19, 527–552.

Steedman, Mark 2000. *The syntactic process.* Cambridge: MIT Press.

Steedman, Mark 2007. Surface-compositional scope-alternation without existential quantifiers. Draft 6.0, August 2007. Available from http://www.iccs.informatics.ed.ac.uk/~steedman/papers.html, last checked 25 August 2009.

Steriade, Donka 1995. Underspecification and markedness. In: John Goldsmith (ed.) *The handbook of phonological theory*, Cambridge: Blackwell. 114–174.

de Swart, Henriette 1998. Aspect shift and coercion. *Natural Language and Linguistic Theory* 16, 347–385.

VanLehn, Kurt 1978. *Determining the Scope of English Quantifiers.* Master's thesis, MIT, Cambridge, MA. MIT Technical Report AI-TR-483.

Villalta, Elisabeth 2003. The role of context in the resolution of quantifier scope ambiguities. *Journal of Semantics* 20, 115–162.

Wahlster, Wolfgang (ed.) 2000. *Verbmobil: Foundations of speech-to-speech translation.* Heidelberg: Springer.

Westerståhl, Dag 1998. On mathematical proofs of the vacuity of compositionality. *Linguistics & Philosophy* 21, 635–643.

Woods, William 1967. *Semantics for a question-answering system.* Ph.D. thesis, Harvard University.

Woods, William 1978. Semantics and quantification in natural language question answering. In: Marshall Yovits (ed.) *Advances in computers*, New York:

Academic Press, volume 17. 2–87.

Wunderlich, Dieter 1991. How do prepositional phrases fit into compositional syntax and semantics? *Linguistics* 29, 591–621.

Keywords:

ambiguity, underspecification, syntax-semantics interface.

Markus Egg, Berlin (Germany)