HARRY BUNT

SEMANTIC UNDERSPECIFICATION: WHICH TECHNIQUE FOR WHAT PURPOSE?

1. Introduction

In recent years a variety of representation formalisms have been proposed that support the construction of underspecified semantic representations, such as Quasi-Logical Form, Underspecified Logical Form, Underspecified Discourse Representation Theory, Minimal Recursion Semantics, Ontological Promiscuity, Hole Semantics, the Constraint Language for Lambda Structures, and Normal Dominance Constraints. These formalisms support methods of underspecification which sometimes seem very different but in fact have similar underlying concepts, and in other cases appear deceptively similar, using the same terminology but with different interpretations. UDRT and Normal Dominance Constraints, for example, at first blush seem quite different but upon closer inspection have much in common; on the other hand, the term 'metavariable' is used by different authors to refer to different concepts in different underspecification formalisms.

Recent studies have produced interesting results about the relative expressive capabilities of some of these formalisms. Koller (2004) has for instance shown that under certain conditions the underspecified representations of Hole Semantics can be translated into normal dominance constraints, and vice versa. Ebert (2005) has shown that the most prominent underspecification formalisms for the representation of scopal phenomena all suffer from lack of expressive power.

This paper aims at contributing to the understanding of the merits of the various underspecification formalisms by considering what different underspecification techniques have to offer for dealing with a range of phenomena that motivate the use of underspecified semantic representations.

H. Bunt and R. Muskens, (eds.), Computing Meaning, Volume 3, 55–85.

2. Why underspecified semantic representation

The use of underspecified semantic representations is motivated primarily by the massive ambiguity that is found in natural language, in particular as revealed through attempts to build effective language understanding systems. Computer implementation of a Montague-style semantics, using a set of construction rules for building formal meaning representations compositionally from lexical meanings, has turned out not to be feasible due to the astronomical number of alternative representations that would have to be built for an ordinary sentence. Hobbs and Shieber (1987) have shown that the ambiguity of relative quantifier scopes means that a sentence with n noun phrases can have up to $n!$ readings, although syntactic constraints tend to reduce this number (giving a sentence with 5 NPs typically between 30 and 40 readings).

The pervasive ambiguity of words is another major cause of the ambiguity explosion in natural language analysis. Bunt and Muskens (1999) estimate that, due to lexical ambiguity and quantifier scope ambiguity alone, an average-length Dutch sentence[1] has some 2.000.000 possible readings. Add to this (or rather, multiply this with) the ambiguities due to collective/individual distincions, specific/nonspecific readings, count/mass ambiguities, PP-attachment possibilities, extensional versus intensional readings,... and it is obvious that the construction of representations for all the possible readings of a given sentence is a computationally extremely expsensive task. Moreover, in a given context nearly all of the possible readings have to be discarded. Constructing representations for all possible readings and subsequently discarding nearly all of them, is arguably the most inefficient way of organising the interpretation process... Clearly, the construction and use of compact underspecified semantic representations, that correspond to sets of fully disambiguated readings, may allow a much more efficient way of processing.

While ambiguity makes the construction of representations of all the possible interpretations of a sentence exceedingly expensive, the phenomenon of *vagueness* presents an even greater, more fundamental problem. In the case of an ambiguity, the number of alternative readings can be large, but is finite. Vagueness is worse, in particular that form of vagueness or 'imprecision' that is caused by the infinite range of

[1] The average length of a written Dutch sentence is approximately 12 words.

possibilities that a speaker has when choosing a certain *granularity* for his referring expressions. Take for instance the adjective *green.* In itself, *green* is not ambiguous between, for example, *light green* and *dark green*, but when your house is being painted using a light and a dark shade of green, then an instruction like *That window frame should be painted green* is ambiguous, and when you enter a paint shop and ask for *green paint*, the shopkeeper will want you to be even more precise; such a context requires a finer granularity. Since the required precision of a referring expression depends entirely on the context, there is no *a priori* way to know how many interpretations of a given sentence should be distinguished; there is in general not even a *finite* number of interpretations. This means that we do not have the choice of constructing the semantic representations of 'all possible interpretations'; the only alternative seems to be to construct representations down to a certain level of granularity, and allowing to infer more specific interpretations in a given context.

Another form of vagueness is the relational vagueness that is found in nominal complexes, where the semantic relation between the constituents is left implicit, as in *Apple computers, Apple employees, Apple logo, university computers, university offices, office teachers*, and so on. This form of vagueness is not due to a certain coarseness in granularity (although the implicit internominal relations may be inferred with varying precision), but it is a case of apparently infinite ambiguity, since there appears to be no limit to the relations that can be intended to connect the constituents. For handling this phenomenon as well as for that of granularity-related vagueness, there seems to be no alternative to the employment of underspecified semantic representations.

Apart from the efficiency and feasibility considerations of dealing with ambiguity and vagueness, there are other processing considerations that favour the construction of underspecified semantic representations. One consideration is for a language understanding system the occurrence of an unknown word, Often, when an unknown word occurs the context makes it possible to guess what the word means approximately, as in *Yesterday I ran into a great green lizard when I crossed the ??tarket?? street; it kept running ahead of me on the pavement until it disappeared under a car*, and it will depend on the context whether it is necessary to know precisely what the word means. The same goes for unknown proper names. A similar situation arises when a word cannot be understood (or read) well for some reason. Clearly, in such situations it is not the case that we have an utterance that cannot be interpreted at all, we rather construct an interpretation with a ‘hole’ for the unknown word; in other words, we construct an underspecified interpretation for the utterance as a whole. For computer systems with spoken input, this situation is quite common.

An independent motivation for assigning underspecified semantic representations to sentences comes from machine translation, since ambiguities in a sentence in the source language are often retained in the target language.

Finally, psycholinguistic considerations also provide arguments in favour of underspecified semantic representations. When listening to a sentence, people clearly do not wait constructing an interpretation until the sentence is complete; instead, they interpret incrementally. This means that underspecified semantic representations are constructed from incomplete input. With a lot of luck, at the end of the sentence the interpreter may be able to construct a fully-specified representation from this, but more probable is that human listeners use context information to construct a representation which is not fully specified, but underspecified to a degree that is acceptable in the given context.

The use of underspecified semantic representations is thus motivated by a range of rather different phenomena, including the following, and summarized in Table 1:

Lexical ambiguity: the referential ambiguity of content words; the count-mass ambiguity of nouns; the possible resolutions of anaphoric and deictic expressions; adjectives by concatenation, also the internal relational ambiguity of compound words.

Syntactic ambiguity: the ambiguities resulting from alternative possible parsings, such as the possible attachments of PPs and relative clauses;

Structural semantic ambiguity: ambiguities that do not have a lexical or syntactic basis, such as the scoping of quantifiers and modifiers; also the collective/distributive ambiguity of quantifiers; for English also the ambiguity of noun-noun complexes;

Semantic imprecision: vagueness, due to relatively coarse granularity in reference; also, the apparently infinite ambiguity of implicit semantic relations;

Missing information: the absence of information due to speech recognition problems, unknown words, or interrupted speech; the requirements of incremental processing; also the use of constructions such as ellipsis and short answers.

Table 1. A taxonomy of motivations of semantic underspecification

General phenomenon	*Instance*
Lexical ambiguity	homonymy; polysemy
	anaphora and deixis
	count/mass use of nominals
	metonymy
	compound nouns
Syntactic ambiguity	PP-attachment
	relative clause attachment
	scope of adjectives and adverbs
	thematic/semantic role assignment
Structural semantic ambiguity	quantifier scope
	quantifier distributivity
	modifier distributivity
	noun-noun complexes
Semantic imprecision	varying granularity of reference
	relational vagueness
Missing information	unknown words
	incomplete input
	ellipsis, short answers
	incremental processing

In the following sections we will discuss the applicability of a variety of underspecification techniques to various important forms of ambiguity, vagueness, and missing information. These findings are summarized at the end of the chapter in Table 2.

3. Semantic Underspecification

3.1. *Underspecified Semantic Representations*

As expressions in a formal language, such as the language of first-order logic or that of constructive type theory. semantic representations can be described syntactically as formed by the recursive combination of subexpressions by means of logical *constructions* such as function application, conjunction, negation, and universal quantification. The semantic definitions of these constructions determine the logically correct patterns of reasoning in which these representations may be used, and usually take the form of specifying how the denotation of an expression, given the way it is constructed, can be computed from the denotations of its subexpressions. Since the atomic subexpressions, such as predicate terms and individual constants, have precisely specified denotations, it follows that every semantic representation also has a precise denotation.

Being the result of applying constructions to subexpressions, a semantic representation can be underspecified in two ways:[2]

1. atomic subexpressions (constants and variables) may be ambiguous, i.e. do not have a single value specified as their denotation, but a range of possible values;
2. the way in which subexpressions are combined by means of constructions may not be fully specified.

A representation which is underspecified in one or both of these ways may be viewed as a representation of *constraints* on fully specified meaning representations, i.e. as a meta-representation describing the set of representations that satisfy the constraints. Such a representation can therefore be a compact representation of a set of readings of a natural language expression. (More on compactness below.)

Since the combination of subexpressions by means of constructions is in general not fully specified, an underspecified semantic representation (USR) is not a single expression, but a set of (sub-)expressions representing the meanings of parts of the sentence, possibly containing

[2] In theory, a third form of underspecification could be to allow ambiguous constructions. For example, one might allow a two-place predicate to combine with a *set* rather than an ordered *pair* of arguments, leaving the semantic role of each of the arguments underspecified. We are not aware of any proposal in this direction.

ambiguous constants and variables, plus a possibly incomplete specification of how these subexpressions may be combined to form a complete semantic representation. So a USR for a given utterance U is a pair:

(18) $R_U = < E_U, C_U >$
where E_U is a set of expressions and C_U is a set of constraints on the admissible ways of combining the subexpressions in E_U.[3]

It follows from definition (18) that a framework for expressing USRs requires not only a language $\mathcal{L}_E$ for (subexpressions of a) semantic representation, but also a *constraint language* $\mathcal{L}_C$ for specifying constraints on combining $\mathcal{L}_E$-expressions. Since the constraints expressed in $\mathcal{L}_C$ refer to $\mathcal{L}_E$-expressions, or more precisely to *occurrences* of such expressions, the expressions in E_U must carry identifiers that can be used in $\mathcal{L}_C$. Therefore $\mathcal{L}_E$ is not just a language as we know it for fully specified semantic representation, but $\mathcal{L}_E$ should additionally have such identifiers and ('meta')-variables ranging over these identifiers.

Note that the expressions in the E_U-component of a USR may themselves be either single $\mathcal{L}_E$-expressions or underspecified representations. As formulated, definition (18) suggests that E_U would consist of single $\mathcal{L}_E$-expressions; this is not really the case, but is in fact immaterial from the point of view of the representational structures that the definition allows. Consider for example a natural language expression S, consisting of two subexpressions S_1 and S_2, and suppose one would want to represent S as consisting of two USRs for these two parts, plus a set of constraints on how to combine them. This would mean that the underspecified representation of S is structured as:

(19) $$\begin{aligned} R_S &= < E_S, C_S > = < \{usr_1, usr_2\}, C_S > \\ &= < \{< E_1, C_1 >, < E_2, C_2 >\}, C_S >. \end{aligned}$$

This last representational structure is equivalent to

(20) $R_S' = < E_1 \cup E_2, C_1 \cup C_2 \cup C_S > = < E_S', C_S' >$

since (19) and (20) contain the same sets of subexpressions and the same sets of combination constraints; the only difference is that in (19)

[3] One could also imagine constraints on the possible interpretations of ambiguous constants and variables, but such constraints are in practice determined by the interpretation framework in which the USR) is used (see below, section 3.3), rather than expressed as part of the USR.

the set of subexpressions has been structured into subsets, and the set of constraints has likewise been structured into subsets that apply to the subsets of subexpressions and to their combination.

3.2. *Underspecification techniques*

The techniques that have been proposed for underspecificied semantic representation can be classified in five groups: (1) *in situ* representations; (2) use of ambiguous terms; (3) labels, holes, and dominance constraints; (4) flat, conjunctive expressions: (5) use of stores and lists. We briefly characterize each of these groups.

In situ representations

One of the oldest approaches to the representation of quantifier structures in an underspecified way is the use of operators in structurally similar positions as the corresponding determiners or quantifiers in the natural language expressions. So a sentence such as *Every student read a book* is represented as something like `read(every student, a book)`.

Various proposals to this effect have been put forward, such as Schubert and Pelletier's 'conventional translations' (Schubert & Pelletier, 1982), but also 'situation schemata' (Fenstad et al., 1987), Quasi-Logical Forms (Alshawi & van Eijck, 1987; Alshawi, 1992), and Underspecified Logical Forms (Geurts & Rentier, 1991; Kievit, 1998).[4]

In the most influential of these proposals, that of Quasi-Logical Form (QLF), predicates have arguments in the form of terms ('quasi-determiners') which include a list of features that capture quantificational information relating to the determiners that are represented. For example, *Every student read a book* would be represented as:

(21)
```
read,
     qterm(<t=quant,n=sing,l=every>, X, [student,X]),
     qterm(<t=quant,n=sing,l=a>, Y, [book,Y]))
```

QLFs and other *in situ* representations were intended to be used in intermediate stages of semantic interpretation, and to be disambiguated (or 'resolved') in a later stage, which involves extracting the *in situ* terms from the USR in a certain order which determiners the relative scoping of quantifiers in a fully resolved logical form.

[4] The oldest proposal in this direction dates back to Woods (1978), who used underspecified representations in the LUNAR question-answering system.

Ambiguous terms in a formal language

The pervasive ambiguity and vagueness of words has inspired the idea of using equally ambiguous and vague predicate constants and other terms in formal representations. This approach was pioneered by the designers of the PHLIQA question answering system (see Bronnenberg et al., 1979), where a formal representation language with ambiguous constants was defined, using a model-theoretic semantics with supervaluations (Van Fraassen, 1966). In fact, *all* the nonlogical constants in this language are ambiguous, at least in principle, just as the content words of natural language.

Once ambiguous terms have been allowed in a formal language, they can be useful also for other purposes, such as for the underspecified representation of the collective/distributive interpretations of quantifiers (Bunt, 1985) and for the compact representation of attachment ambiguities (Bunt, 1995). Such constants do not correspond to any natural language words, and have an ambiguity that is determined by semantic theory; for instance, the possible values of formal metaconstants for quantifier scoping and distributivity are provided by the theory of quantification that is used.

Yet another use of ambiguous terms has been introduced in some approaches to underspecification, such as CLLS (see below), where so-called *metavariables* are used that are intended to be replaced by semantic expressions that are generated from the context. We therefore distinguish the following three cases in the use of ambiguous terms in USRS:

1. Predicate constants, function constants or individual constants that represent ambiguous or vague content words, such as homonymous, polysemous, or 'vague' nouns, verbs and adjectives. They may be 'disambiguated' by being replaced by more specific terms or expressions of the representation language. We will refer to these as *referential metaconstant.*
2. Constants that represent a formal semantic property or relation which is not expressed explicitly in natural language, such as the scoping relation between noun phrases, or the distributivity of a quantification. Their 'disambiguation' typically consists of structuring a semantic represen tation in a certain way. We will refer to such terms as *formal metaconstant.*
3. Terms which are intended to be replaced by a constant, a bound variable, or another subexpression that either occurs elsewhere in

the representation, or that is generated from the context. We will refer to such terms as *metavariables*.

Labels, holes, and dominance constraints.
For underspecifying the way in which the E_U-elements in a USR $U =< E_U, C_U >$ are to be combined, a label may be associated with each element in E_U, and the fact that a certain subexpression labelled L_1, consists of two subexpressions joined by means of the construction κ, of which the first one a subexpression labelled L_2 and the second is unknown, may be represented as $L_1 : \kappa(L_2, h_1)$, where h_1 is a 'hole', i.e. a variable that ranges over the labels of the subexpressions in E_U.[5]. The precise ways in which holes may be used has been described in terms of possible 'pluggings', operations for replacing hole variables by subexpressions.

The approach of labelling subexpressions and using variables to refer to subexpressions in the specification of constraints in a metalanguage, is clearly applicable to any given object language. Bos (1995) formalizes the use of labels and holes in propositional and (dynamic) predicate logic in a way that is easily extended to other object languages. In (Bos, 2002) he applies Hole Semantics to DRT.

The use of labels to mark subexpressions and constraints on their possible combinations was originally invented in DRT, leading to UDRT (Reyle, 1993; 1996), specifically for the underspecified representation of quantifier scopes.

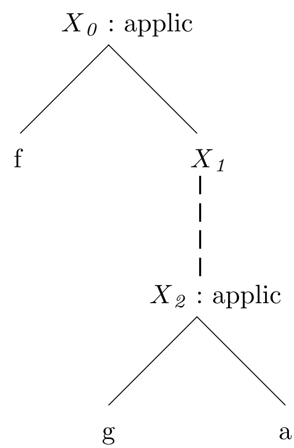
Minimal Recursion Semantics (MRS) applies the idea of labelling subexpressions and expressing constraints on their possible combinations to a language of typed feature structures (attribute-value matrices), fitting in with the object language of HPSG (Copestake et al., 1995). Labels are called 'handles' (or 'handels') in MRS, and a peculiarity of MRS is that structure sharing between features is interpreted as conjunction, which gives MRS representation a relatively flat structure. An MRS representation is essentially a pair consisting of a list of labelled feature matrices, where the labels may occur as feature values, and a set of constraints on the labels. Although MRS representations look rather differently from USRs in first-order logic HS and from UDRSs, the underlying ideas of all three approaches are clearly very similar. The studies by Koller et al. (see Koller, 2004) and Ebert (2005) make the similarities and differences explicit between the various approaches based on labels and constraints.

[5] Bos (2002) also calls hole variables *metavariables*; see below.

'Dominance Constraints' refers to a general framework for the partial description of trees, which has been used in various parts of computational linguistics (see e.g. Rogers & Vijay-Shanker, 1994; Gardent & Webber, 1998; Muskens, 2001). For underspecified semantic representation, the use of dominance constraints relies on the fact that logical formulas can be represented as trees, labeled with construction names. A USR can therefore take the form of a partial description of a tree. For instance, the USR (22a) can be repressented as the partially specified tree (22b), where a dotted line connecting two nodes indicates a dominance relation but leaves open that some material may come in between.

(22) a. $< \{X_0\text{: applic}(f, X_1),\ X_2\text{: applic}(g, a)\},\ \{X_0 > X_2, X_1 > X_2\} >$

b.



A dominance constraint representation is formally a USR as defined in (1) above, where the set of constraints is restricted to constraints of two kinds; those of the form $X_1 > X_2$, interpreted as indicating that the node X_1 dominates the node X_2, and those of the form $X_1 \neq X_2$, expressing inequality of nodes. (See Koller et al., 2003 for more detailed formal definitions.) General dominance constraints have bad computational poperties, therefore Koller et al. (2003) devised the restricted form of dominance constraints called Normal Dominance Constraints (NDC), which are computationally more tractable. Koller (2004) has shown that NDC has the same expressive power as Hole Semantics with certain plausible normality constraints, and claims that these consraints as well as the normality restrictions of NDC are linguistically adequate in the sense that all underspecified semantic representations of natural language expressions satisfy these restrictions. Ebert (2005) provides evidence that this is not the case, however, and argues that both NDC and Hole Semantics are therefore expressively incomplete.

The NDC framework has been constructed as a restriction of the more powerful Constraint Language for Lambda Structures (CLLS, Egg et al., 1998). CLLS is an expressive language of tree descriptions which combines dominance constraints with parallelism constraints for dealing with VP ellipsis and anaphoric binding constraints to represent intrasentential anaphora.

Glue Semantics was originally developed not as a formalism for, semantic underspecification, but for defining the syntax-semantics interface in LFG (Dalrymple, 2001). Glue Semantics uses Linear Logic in order to deductively piece together the meanings of individual words and constituents in a sentence. Premises for such deductions are 'meaning constructors' obtained from the lexical entries of the words, showing how meanings assigned to various constituents can be combined to build meaning assignments for other constituents. This naturally leads to intermediate underspecified representations in the glue language.

The most important insight of Glue Semantics is perhaps not so much its particular representation forms, but its strategy of using logical inference to construct semantic representations. Crouch et al. (2001) show that the glue language can be used to construct UDRSs in a computationally attractive way, and Pulman (2000) shows how the Glue Semantics strategy can be applied to (a cleaned up version of) the QLF formalism in order to infer fully resolved logical forms from QLFs using context information.

Flat conjunctive representations

The davidsonian approach of reifying the states and events associated with verbs (Davidson, 1967) naturally leads to conjunctive semantic representations like (23) for a sentence like *John saw Mary yesterday*:

(23) $\exists e{:}\mathsf{see(e)} \wedge \mathsf{agent(e, john)} \wedge \mathsf{theme(e, mary)} \wedge \mathsf{time(e, yesterday)}$

Hobbs (1983) has proposed to apply reification not only to verbs but also to nouns, adjectives, adverbs, and prepositions, constructing 'flat' representations in the form of conjunctions in first-order logic. An interesting property of such representations is that underspecification takes the form of leaving out certain conjuncts. This means that an 'underspecified' representation is in no way different in form from a fully specified representation. For instance, a representation of *Somebody read every article* which leaves the relative scoping of the two NPs underspecified could be as follows:

(24) $\exists e$:read(e) $\wedge$ somebody(x) $\wedge$ agent(e,x) $\wedge$ every(y) $\wedge$ article(y) $\wedge$ theme(e,y)

This approach entails a rather bewildering ontological picture, which has made Hobbs refer to it as 'ontological promiscuity'. We will call this approach *radical reification* (RR).

Another form of representations that is in a sense flat, and shares with RR-representations the property that USRs have the same form as fully specified semantic representations, is in terms of typed feature structures. Bunt (2005) has shown that representations of quantification which leave scope or distributivity (or both) underspecified, can be cast in the form of feature structures. In this case, underspecification takes the form of leaving out those attributes in an attribute-value matrix that have no value specified. These representations are also conjunctive in nature, since the interpretation of an attribute-value structure is in terms of the logical and of its attribute-value pairs.

Stores and lists

A relatively old idea is to accompany the construction of semantic representations by a symbolic memory in which those components of a representation are temporarily stored whose position in the representation is not yet fully determined. Cooper storage was developed for doing this for quantifier scopes (Cooper, 1983). Keller (1998) has developed an improved version of the Cooper store techniques with nested stores, known as Keller stores. Some experimental language understanding systems use a similar technique, placing NP representations on a list from which they can be tretrieved in an order that corresponds to their relative scopes.

In ULF, a representation language that was used in the DenK system (Kievit, 1998) lists of variables are used as an alternative to metavariables constrained to be instantiated as one of the elements in the list. Also, lists of predicates are used to indicate that the predicates are to be combined somehow in order to form a complex predicate.

3.3. *Requirements on Representations, and Interpretation Frameworks*

The various techniques for underspecified semantic representations have often been developed for being used within a certain theoretical or computational framework. As already noted, MRS was developed for use within the framework of HPSG, and therefore employs feature structures that integrate well with the 'signs' that are used in HPSG for representing linguistic information of all kinds; Glue Semantics was designed for developing the syntax-semantics interface in LFG. UDRT was a further development of DRT (Kamp & Reyle, 1993).

Other underspecification techniques were developed for use within a certain processing architecture. This is for instance true of the QLFs developed in the CLE system, the ambiguous metaconstants of the PHLIQA system, and the ULFs of the PLUS and DENK systems. Radical reification assumes an abduction-driven interpretation process; CLLS and Pulman's renewed version of QLF (Pulman, 2000) assume an interpretation process with higher-order unification.

We will use the generic term *Interpretation Framework* for indicating the theoretical or computational framework which a certain underspecification approach assumes. The distinguishing features of many underspecification techniques are due to their interpretation framework, which brings certain theoretical or computational requirements.

There are also certain requirements that *any* underspecification technique should meet. Two particularly important requirements are those of *compactness* and *expressive completeness.* The requirement of compactness is, informally, that the use of a USR should have a real advantage over the use of the set of all fully specified representations. A USR which simply lists all the fully specified representations, for example, does not satisfy this requirement. Ebert (2005) has formalized the notion of compactness of USRs.

The requirement of expressive completeness means that an underspecification approach should allow the representation not only of interpretations which are entirely unspecified w.r.t. a particular aspect, such as quantifier scoping, but also those interpretations which are *partly* specified, or, in other words, which are partly disambiguated. The need to represent these stems from the desire to represent in a USR what the syntactic and lexical information in a given sentence tell us semantically, no more and no less. This is obviously motivated from the wish to have a satisfactory syntax-semantics interface, but also from the observation that people sometimes express themselves deliberately vaguely or ambiguously, and an adequate semantic representation framework should be able to deal with that. Koller (2004) has shown that several underspecification formalisms are equally expressive, if certain normality conditions are imposed on the representation structures. NDC already has such conditions in its definition, and Koller argues that HS and MRS could very well also have some such conditions imposed on their definitions, since from a linguistic point of view only 'normalized' MRS and HS representations are needed. On the other hand, Ebert (2005) provides evidence to the effect that HS as well as NDC and MRS are unable to represent certain linguistically relevant partial disambiguations, and are therefore expressively incomplete, Hole Semantics and Normal Dominance Constraints were designed in a framework-independent way. The same goes for Radical Reification and the use of flat typed feature structures.

A rather obvious but far from trivial requirement on USRs is that they should be semantically well-defined. Van Deemter (1996) analyses USR proposals from this point of view, in particular in relation to their role in inference patterns; König & Reyle (1996) provide a logical basis for a broad range of underspecification formalisms.

We will now consider the various underspecification techniques that have been introduced here for their suitability to handle the various phenomena that motivate the use of underspecified semantic representations.

4. Applicability of underspecification techniques

4.1. *Lexical ambiguity*

The efficient processing of sentences with homonymous words has motivated the oldest known use of underspecified semantic representation. which makes use of the technique of ambiguous constants in a formal representation language. In the PHLIQA question answering system (see Medema et al., 1975; Bronnenberg et al., 1979) a representation language was used (a typed higher-order lambda calculus) with a *referential metaconstant* for each English content word, the idea being that there is hardly any content word that does not display some degree of homonymy or polysemy. So for instance, the adjective *American* corresponds to the metaconstant american, representing the various senses of the word *American* as illustrated by *American car*, *American city, American flag*, and *American airplane.* A domain-specific lexicon lists the possible instantiations of each metaconstant, such as american having as three of its interpretations:

(25) a. american $\rightsquigarrow \lambda x.$ Nationality-of(Manufacturer-of(x)) = USA.
b. american $\rightsquigarrow \lambda x.$ Country-location-of(x)) = USA.
c. american $\rightsquigarrow \lambda x.$ Nationality-of-(Carrier-of(x)) = USA.

Referential metaconstants seem the perfect instrument for underspecifying these forms of semantic indeterminacy, especially within an interpretation framework like that of the PHLIQA system, where a domain-specific lexicon determines the contextually relevant senses of ambiguous lexical items.

A rather different kind of lexical ambiguity is the one between anaphoric and deictic use of pronouns and definite NPs (*Did you see that?*), and between their possible referents in the linguistic or situational context. The simplest way to underspecify the intended interpretation of an anaphoric or deictic expression would seem to be the use of a metavariable, with the syntactic/semantic constraints that the expression provides. For instance, for the English pronoun *her* as occurring in *Did you see her* the constraints include that the referent plays the role of the theme in a *see* event and is a female person or animal. This underspecification technique has been used in the DENK system (see Kievit et al., 2001).

Sentence pairs such as

(26) a. *There's no chicken in the yard*
b. *There's no chicken in the salad*

illustrate the count/mass ambiguity that is found in many languages. In English, virtually every noun can be used both as a count noun and as a mass noun.[6] Treating every noun in the lexicon as ambiguous between a count and a mass reading is computationally unattractive and conceptually unsatisfactory, since there are systematic semantic relations between the count and mass uses of a word. Bunt (1985) has introduced the use of *formal metaconstants* for converting a count noun to its mass use and the other way round. For instance, the noun

[6] In a pamphlet in a hospital ward the following text was found: *Nurses spend a great deal of their time washing patients, and since the population has become more than 10% taller in the last ten years, they have correspondingly more patient to wash.*. See Bunt (2006) for more about the count/mass distinction

bread is represented in the lexicon as bread, and the grammar rules for constructing noun phrases may turn this into μ(bread), adding a function constant μ that can be instantiated in two ways, representing the count and the mass reading. This approach has been implemented in the TENDUM dialogue system (Bunt et al., 1984).

Other underspecification techniques besides the use of referential or formal metaconstants are hardly available for dealing with lexical ambiguity. For instance, the use of labels, holes and dominance constraints for lexical sense underspecification runs up against the problem that there is no definite, finite list of possible pluggings or substitutions for holes or metavariables. One could use metavariables with an interpretation framework where they are instantiated by means of a context-specific word sense lexicon, but that is in fact precisely the use of referential metaconstants.

4.2. *Structural semantic ambiguity: quantifier scoping*

Underspecified semantic representation of quantifier scopes have been proposed using the following of the above mentioned techniques.

In situ representation. Alshawi's original QLF proposal was implemented in the Core Language Engine (CLE). This interpretation framework includes a disambiguation process that pulls the quantifiers out of the predicate arguments and assigns them a scope (Alshawi, 1990). Alshawi and Crouch (1992) have provided a semantics for QLF representations in terms of their disambiguations (resolved quasi-logical forms), an approach that is generally available for USRs – at least, if they have a well-defined set of possible disambiguations.

Willis & Manandhar (2001) have argued that QLFs cannot represent *partial* scope information adequately (for which the indices of the qterms can be used), and for instance does not get the scope constraints for the sentence *Every representative of a company saw some samples* right. The QLF formalism thus suffers from expressive incompleteness (Ebert, 2005). Pulman (2000) also notes several shortcomings of the CLE QLF and proposes an improved version. The in situ representations used in Allen's textbook (Allen, 1995) and those used in ULF (Geurts & Rentier, 1991; Kievit, 1994), suffer from the lack of a constraint specification language $\mathcal{L}_C$, and therefore do not meet the requirement of expressive completeness.

Stores and lists. The mechanisms of Cooper storage and Keller storage do not quite form an underspecified representation technique, but a procedure for postponing decisions on the relative scopes of noun phrases during much of the interpretation of a sentence. Some language understanding systems of the seventies and eighties, such as PHLIQA, SPICOS and TENDUM have incorporated list-based representations with a similar effect (van Deemter et al., 1984; Scha, 1981).

Bunt and Muskens (1999) have described a formal calculus assigning logical forms to syntactic trees using an NP store, which (like Keller store) has the nice property that when an NP can raise out of an NP, as in *Every representative of a company saw some samples*, this is allowed only if the embedded NP is retrieved from the store later than the embedding one.

Ambiguous terms. A treatment with formal metaconstants has been implemented in the TENDUM dialogue system. All the noun phrases in a clause are collected in a 'noun phrase sequence' constituent, collecting the NP representations as the argument of a metafunction. Alternative instantiations of this metafunction correspond to alternative relative scopings. Clause formation occurs through combination of the NP sequence with the verbal constituent. A shortcoming of this technique is that sentence-specific constraints on the possible instantiations of the metaconstant cannot be expressed as constraints on metaconstant instantiation, hence there is no adequate representation of partial scoping, and so the requirement of expressive completeness is not met.

Labels, holes, and dominance constraints. The use of subexpression labelling with scope constraints was originally invented in UDRT (Reyle, 1993) for the underspecified representation of quantifier scopes. NDC and HS were likewise designed specifically for the underspecification of quantifier scopes, Ebert (2005) has shown that UDRT, MRS and NDC are all expressively *incomplete*, as they are not always able to represent all the linguistically relevant sets of readings of a given sentence. Still, these formalisms offer the best known possibilities for representing scopal ambiguities in a compact manner.

The ideas of UDRT and MRS have been implemented successfully in the Verbmobil on-line translation system (see Schiehlen, Bos & Dorna, 2000); these of NDC were implemented in the CHORUS project (see Koller, 2004).

Flat, conjunctive representations. Hobbs (1983) has proposed flat, conjunctive semantic representations which contain less information then is usually considered adequate, but where the abduction-driven pragmatic component in the interpretation framework supplies additional information. The flatness of the representations is attractive, but the price is a rather cumbersome treatment of quantification, involving the notion of a ‘typical element’ of every set, and other ontologically strange creatures. Moreover, as Hobbs (1996) shows, in an attempt to fix certain shortcomings of his original proposal, the representations of simple sentences in fact become very complicated. For instance, the sentence *Most men work* would be represented in RR as:

(27) $(\exists s_2, s_1, x, e, y)[\mathsf{most}(s_2, s_1) \wedge \mathsf{dset}(s_1, x, e) \wedge \mathsf{man}(e, x) \wedge \mathsf{typelt}(y, s_2) \wedge \mathsf{work}(y)]$

which is to be read as: There is a set s_1 defined by the property e of its typical element x being a man, and there is a set s_2 which is most of s_1 and has y as its typical element, and y works.

RR does not come with a constraint language, and is therefore unable to express partial scope disambiguations. This makes RR expressively incomplete.

Another kind of flat representation that has been suggested exploits the expressive power of typed feature structures. These representations are for the most part (though not entirely) flat in the sense that the feature structures that have been proposed contain very little nesting, and are by and large of a flat, conjunctive character (see Bunt, 2005), thus allowing efficient processing. This use of feature structures has grown out of work on the development of expressive but efficient systems for semantic annotation.

4.3. *Other structural semantic ambiguities*

The ambiguities involved in the *distributivity* of quantifiers has received much less attention than those in quantifier *scoping*, although every NP introduces distributivity ambiguities. A single NP can be argued to be multiply ambiguous between individual, collective, cumulative, and group readings, and the combination of two ore more NPs gives rise to additional distributivity ambiguities. Some examples are:

(28) a. These machines will lift the platform. [*together*]
b. These machines lift 5 crates. [*in one go*]
c. These machines have lifted 2000 crates. [*in total*]

The only approach to underspecifying quantifier distributivity that has been proposed, to our knowledge, is the use of formal metaconstants (Bunt, 1985). In this approach, predicates like Lift are applied to arguments from a domain represented as $\delta(num, \mathsf{machines})$, where δ is a formal metaconstant representing distributivity, and *num* stands for the (absolute or relative) numerical information in the (generalized) quantifier. The metaconstant can be instantiated in alternative ways, (where *num* is a group size), as (29a), b and c:

(29) a. $\delta \rightsquigarrow (\lambda M.\ \lambda X.\ X)$
b. $\delta \rightsquigarrow (\lambda M.\ \lambda X.\ \{X\})$
c. $\delta \rightsquigarrow (\lambda M.\ \lambda X.\ \{Y | Y \subseteq X\ \wedge\ num(Y)\})$,

Distributivity ambiguities arise not only in quantification but also in *modification*, as in *The crates that this machine lifted*, which can be taken both individually and collectively. The kind of representation of quantifier distributivity that we have just seen, by means of a formal metaconstant, can also be used to represent modifier distributivity in an underspecified way (see Bunt, 2005).

Another form of structural ambiguity occurs in English for nominal compounds. Hobbs et al. (1993) have proposed a treatment with formal metavariables, to be instantiated through abductive reasoning with context information. For example, the compound *Boston office* in sentence (30a) is represented schematically as in (30b):

(30) a. The Boston office called
b. $\mathsf{boston}(x) \wedge \mathsf{office}(x) \wedge \mathsf{NN}(x)$

where NN is a metavariable (in the sense defined above), representing the unknown semantic relation between the two nouns. In languages where nouns (and adjectives) are concatenated to form compound words, rather than multi-word expressions, this form of ambiguity arises at the lexical level and can be treated in essentially the same way, decomposing the compound word into its constituent parts.

The interpretation problem for nominal compounds is very similar to that of *metonymy* (in fact, *The Boston office called* is also metonymous). Metonymy is one of the types of ambiguity for which Pinkal (1999) suggests an underspecified treatment with dominance constraints using the CLLS formalism. For the example sentence *John began the book* he provides the schematic representation (31a).

(31) a. $< \{X_0$: begin(john, X_1), X_2: the-book$\}$,
$\{X_1 > X_2, X_0 > X_1\} >$

b. begin(john, writing-of(the-book))

The subexpressions labelled X_0 and X_2 in this representation are not connected, but *the book* is constrained to be outscoped by the unspecified subexpression labelled X_1. The idea is of course that X_1 identifies a subexpression that should be plugged in and connect the two subexpressions, such as X_1: writing-of(X_2), giving the result (31b), after replacing labels by the subexpressions that they label and suppressing the top label X_0.

The use of labels and pluggings that we see here differs importantly from that of Hole Semantics, where labels serve only to formulate constraints on the possible instantiation of holes, which range over the labels in the E_U part of a USR $< E_U, C_U >$. By contrast, a hole like X_1 in the above example does not relate to any element in the set $E_U = \{X_0$: begin(john, X_1), X_2: the-book$\}$; instead, X_1 stands for any object-level expression that can be generated through reasoning with contextual information. The dominance constraints approach is therefore in general more powerful than that of Hole Semantics , and its power is in fact determined by the interpretation framework associated with it, wthat determines which expressions can be generated from the context to instantiate hole variables. The use of a term that stands for an expression which is not given in the underspecified representation, but that has to be generated from the context, is is in fact the use of a *metavariable*, in the sense defined above, rather than the use of a hole variable.

4.4. *Syntactically-based ambiguity*

Of the many forms of syntactic ambiguity, we consider syntactic scope ambiguities and attachment ambiguities.

Syntactic scope ambiguities can be handled elegantly by means of labels and holes. For instance, Bos (1995) shows how the sentence (32a) can be represented schematically without resolving the relative scopes of *do not* and *and* by the USR (32b).

(32) a. Do not sleep and pay attention, please.

b. $< \{L_1 : \neg h_1,\ L_3 : \textsf{sleep},\ L_4 : \textsf{pay-attention},\ L_2 : h_2 \wedge h_3\},$
$\{h_1 \geq L_3, h_2 \geq L_3, h_3 \geq L_4\} >$

The constraints in the second part of the USR express that the argument of the negation outscopes *sleep* and that the two arguments of *and* outscope *sleep* and *pay attention*, respectively.

Similar treatments are obviously possible in other label-based approaches (MRS, UDRT, CLLS, NDC). Formal metaconstants have also been proposed for treating this kind of ambiguity (Bunt, 1995), but in the absence of a constraint specification language (in which the possible instantiations of the metaconstants would be constrained), this proposal is expressively incomplete.

Molla (2001) has proposed a variant of radical reification in which all logical relations are reified as well, resulting in flat list representations. For the most plausible reading of (32a) the RR representation would be:

(33) $\textsf{not}(e_1, e_3),\ \textsf{sleep}(e_3, x),\ \textsf{and}(e_4, e_1, e_6),\ \textsf{pay-attention}(e_6, x)$

To underspecify the scoping, the arguments of the scope-bearing elements are simply not tied together, and we get the USR (34):

(34) $<\{\textsf{not}(e_1, e_2),\ \textsf{sleep}(e_3, x),\ \textsf{and}(e_4, e_5, e_6),\ \textsf{pay-attention}(e_6, x)\},$
$\{(e_2 = e_3 \wedge e_5 = e_1) \vee (e_2 = e_4 \wedge e_5 = e_3)\} >$

While interesting as a way of dealing with such scopal ambiguities, this variant of RR is deficient in its treatment of quantification. Note that the e_i variables in (33) act as a kind of subexpression labels, but do not have the same expressive power since one cannot have something like dominance constraints for them.

An *attachment ambiguity* occurs when a sentence contains several candidates for being modified by a certain modifier. Two important cases of this are PP attachment and relative clause attachment, as illustrated by (35):

(35) a. John saw the man with binoculars.

b. The crates on the platform that Hercules lifted.

Syntactically, the different ways of attaching the PP and the relative clause come down to different ways of connecting the subtree, describing the modifier, to the rest of the tree, and it might seem that this corresponds semantically to different ways of inserting the representation of the modifier in the rest of the semantic representation. Underspecifying the attachment would then take the form of keeping the modifier representation separate and indicating its possible insertion points. Labels and holes would seem to be the obvious instruments for achieving this. Schematically, we can represent the two readings of (35a) as (36a) and (36b), and in labelled form as (37a) and (37b).

(36) a. $\mathsf{saw}(e_1,\mathsf{j},\mathsf{x}) \wedge \mathsf{theman}(\mathsf{x}) \wedge \mathsf{withbinocs}(e_1)$

b. $\mathsf{saw}(e_1,\mathsf{j},\mathsf{x}) \wedge \mathsf{theman}(\mathsf{x}) \wedge \mathsf{withbinocs}(\mathsf{x})$

(37) a. $\{L1 : \mathsf{saw}(e_1,\mathsf{j},\mathsf{x}),\ L2{:}\mathsf{theman}(\mathsf{x}),\ L3{:}\mathsf{withbinocs}(e_1)\}$

b. $\{L1 : \mathsf{saw}(e_1,\mathsf{j},\mathsf{x}),\ L2{:}\mathsf{theman}(\mathsf{x}),\ L3{:}\mathsf{withbinocs}(x)\}$

The only difference between the two readings is the argument of the modifier. Indeed, from a semantic point of view an attachment ambiguity is a choice of modifier argument. Therefore, holes can be used for underspecifying the attachment only if the alternative arguments of the modifier are labelled subexpressions, as in (38):

(38) $< \{L1 : \mathsf{john},\ L2 : e_1,\ L3 : \mathsf{saw}(L2, L1, L3),\ L4 : \mathsf{theman}(L5),$
$L5 : x,\ L6 : \mathsf{withbinocs}(h_1)\},\ \{h_1 = L2 \vee h_1 = L5\} >$

Technically this seems possible, but note that it it does not make sense to label a variable or a constant, as in L_1 : john, $L2 : e_1$ and $L5 : x$, since the constant and variable themselves can be inserted directly in the semantic representation, resulting in the simpler USR (39).

(39) $< \{\mathsf{saw}(e_1,\mathsf{j},\mathsf{x}) \wedge \mathsf{theman}(\mathsf{x}) \wedge \mathsf{withbinocs}(h_1)\},$
$\{h_1 = e_1 \vee h_1 = x\} >$

In this representation, h_1 is clearly a variable that can be instantiated as a constant or an object-language variable, hence h_1 is in fact not so much a hole but a *metavariable*.

A general point to note about ambiguities that have their origin in a syntactic ambiguity, is that, even if it is possible to represent the various possible readings by a single underspecified semantic representation, this is only useful if the corresponding syntactic analyses are equally representable by a single ambiguous ‘packed’ syntactic representation; otherwise the interpretation process would generate a number of syntactic analyses, each associated with the same USR. That would not only miss the efficiency gain that motivates the use of USRs, but would even be wrong, since it suggests that each of the syntactic analyses has all the possible semantic readings.

Attachment ambiguities are especiaaly difficult in this respect. Take for instance the following sentence:

(40) John saw the man on the hill with the telescope.

The PP *on the hill* has two possible attachments, and the PP *with the telescope* three. However, of the six possible combinations, one does not correspond to a possible reading of the sentence, namely the one where the man has the telescope and the see-event occurred on the hill. The impossibility of this reading must be due to syntactic reasons, for semantically that reading makes perfect sense. It seems very difficult to model this phenomenon with syntactically and semantically underspecified representations. Muskens (2001) has argued strongly in favour of a unified approach where both syntactic and semantic representations take the form of (partial) descriptions of trees, using Tree-Adjoing Grammar for syntactic analysis, in order to have a better handle on the desired parallelism of syntactic and semantic underspecification.

4.5. *Granularity and vagueness*

The treatment of lexical ambiguity with the help of ambiguous referential metaconstants in the frepresentation language, outlined above in section 4.1, can be applied equally well to effectively deal with the vagueness that is inherent to virtually all nouns, verbs and adjectives because they refer with a certain granularity that may be too coarse in a given context.

The virtually infinite ambiguity of implicit semantic relations, of which we saw examples in section 2, can be treated effectively by introducing a metavariable, similar to the NN predicate of Hobbs' treatment of metonymy. This is especially useful in an interpretation framework where the representation language is typed, so that the types of the arguments of this predicate can be used to infer a contextually suitable interpretation of it.

4.6. *Missing information*

In the case of intrasentential ellipsis, some linguistic material is missing locally, which can be supplied from elsewhere in the sentence. This seems an ideal application of labels and holes. As Pinkal (1999) points out, however, parallelism has to be taken into account, for instance to make sure that the relative scope assignments in the first and the second part of a sentence like *Two European languages are spoken by every linguist, and two Asian languages are, too* are the same. So the constraints in the USR should take such parallelism into account. The CLLS formalism was developed specifically with this aim. The other labels-and-constraints based formalisms are unable to represent parallelism constraints in their constraint language.

The occurrence of unknown words in the input to a language understanding system may be considered as causing ambiguity in the extreme: unknown words can mean anything that could make sense in the context of utterance. Therefore the treatment of ambiguous and vague words by means of metavariables can also be applied in this case.

Another plausible approach to the occurrence of unknown words or of parts of an utterance that cannot be recognized (as in the case of imperfect speech recognition), consists of constructing labelled semantic representations for those parts of the input that can be processed, and add labels for any material that cannot be interpreted, possibly with certain constraints on the interpretation and on how the various pieces of the input may connect. For instance, Pinkal (1999) has suggested a treatment using the CLLS representation language, where the example sentence *We meet XX next week*, where *XX* marks an unrecognized part of the input, would have the following (schematic) underspecified representation (41):

(41) $< \{X_1\text{: meet(we)}, X_2, X_3\text{: next-week}\}, \{X_0 \geq X_1, X_0 \geq X_2\} >$

where X_1 and X_2 label the two semantic chunks roccresponding to the recognized parts of the input, and X_3 the unrecognized parts (and X_0 is the top label of the representation). These pieces might get connected, as Pinkal suggest, by adding the constraints $X_3 = T\_rel(X_2)$, $X_0 = X_3(X_1)$, and $T\_rel \in \{\ldots\}$, where $\{\ldots\}$ is a set of temporal relations, so T_rel is a referential metaconstant (which Pinkal calls a 'metavariable') ranging over temporal relations. These constraints express the assumption that the unrecognized part expresses a temporal relation which takes the part X_2 as its second argument and the part X_1 as its first. The result of taking these constraints into account could be that we obtain a fully specifed representation like (42).

(42) (In(next-week))(meet(we)

Notice that in the process we have used a constraint of the form $X_3 = T\_rel(X_2)$, which contains a semantic construction of the object language, as well as a referential metaconstant *T_rel* ranging over object-language predicates; and the additional constraint $X_0 = X_3(X_1)$ which does not specify a dominance constraint, but a semantic relation connecting the twho first of the two recognized parts to the rest of the input. This consequently is not just an application of the technique of labels, holes and constraints; it also makes essential use of metavariables (X_3) and of the powerful interpretation framework of higher-order lambda calculus that comes with the CLLS representation language.

Incomplete input and incremental processing both have the effect that the utterance interpretation process has the task of assigning a semantic representation to an incomplete fragment of the input. Incremental processing is equivalent to processing an input with an unrecognized part at the end, like *We will meet some time during XX*. This suggest that the treatment of unrecognized input with metavariables , dominance constraints and referential metaconstants can be applied also in this case.

5. Summary and conclusion

We summarize the applicability of the various underspecification techniques in Table 2. The techniques based on the use of labels and dominance constraints (UDRT, HS, MRS, NDC) have been grouped together in one column in view of their comparable expressiveness, as shown by Koller (2004). A '+' sign in this table means that the technique under consideration is suitable for dealing with a particular phenomenon; it does not mean that the technique is perfect for that purpose – probably *no* technique is perfect, given the results on expressive adequacy from Ebert (2005). A ± sign is used to indicate that a technique is suitable only if supplemented with an adequate constraint language. Three rows near the bottom of the table contain the pattern '(+) + (+)'; this indicates that the phenomena under consideration can be handled by a combination of these three techniques, with the one in the column that has a '+' without parentheses playing center stage.

Three columns occupying the center of the table represent the applicability of various forms of ambiguous terms in a formal representation language. Recall that *metavariables* are understood here as terms which are intended to be replaced by a subexpression that either occurs

Table 2. Applicability of underspecification techniques

Phenomenon	Labels, domin. consts.	Meta-variables	Ref. meta-cons.	Formal meta-cons.	Radical reifi-cation	Stores, lists, in situ
Lexical ambiguity						
homonymy	–	–	+	–	–	+
anaphora; deixis	–	+	–	–	–	±
count/mass use	–	–	–	+	–	–
metonomy	–	–	–	+	–	–
compound nouns	–	+	–	–	–	–
Syntactic ambiguity						
modifier attachment	–	+	–	±	±	–
syntactic scope	+	–	–	–	±	–
thematic/semantic role	–	+	–	–	±	–
Struct. sem. ambiguity						
quantifier scope	+	–	–	–	–	±
quantifier distributivity	–	–	–	+	–	–
modifier distributivity	–	–	–	+	–	–
nominal complexes	–	+	–	–	–	+
Semantic imprecision						
polysemy	–	–	+	–	–	–
granular vagueness	–	–	+	–	–	–
relational vagueness	–	–	+	–	–	–
Missing information						
unknown words	(+)	+	(+)	–	–	–
incomplete input	(+)	+	(+)	–	–	–
ellipsis; short answers	±	–	–	–	–	–
incremental processing	(+)	+	(+)	–	–	–

elsewhere in the representation or that is generated from the context. *Referential metaconstants*, by contrast, are ambiguous predicates and other nonlogical constants that represent ambiguous or vague content words, and *formal metaconstants* are terms in the representation language that do not correspond to anything that is expressed explicitly in natural language, but to a formal semantic property or relation such as the distributivity of a quantifier, or the relation between the count and mass senses of a noun.

Two observations from Table 2 are that:

1. each underspecification technique is applicable only to a limited subset of the phenomena that call for the use of underspecified semantic representations;
2. the various kinds of underspecification techniques have some overlapping applicability, but by and large they each apply to different phenomena.

In fact, we see quite clearly that the techniques based on subexpression labelling and dominance constraints are useful for dealing with scope ambiguities, both syntactic and purely semantic ones, and in combination with metavariables also for dealing with cases of missing information. Ambiguous terms of the various kinds are particularly useful for dealing with lexical ambiguity and vagueness, and metavariables have interesting applications in combination with labels *cum* dominance constraints and referential metaconstants.

One general conclusion seems unescapable: a single, unified framework for dealing with all kinds of ambiguity, vagueness, and incomplete information will not be based on just one of the underspecification techniques that we currently know. Instead, the wide range of phenomena for which underspecified semantic representations is useful or even a necessity, calls for the use of a combination of underspecification tools and techniques.

Acknowledgements

I would like to thank Johan Bos, Manfred Pinkal and Steve Pulman for their comments on an earlier version of this chapter.

References

Allen, J.: 1995, *'Natural Language Understanding'.* Redwood City, California: Benjamin/Cummings.

Alshawi, H.:1990, 'Resolving Quasi Logical Form.' *Computational Linguistics* 16:133–144.

Alshawi, H.: 1992, 'The Core Language Engine'. MIT Press, Cambridge, MA.

Alshawi, H. and J. van Eijck: 1992, 'Logical Forms in the Core Language Engine'. In *Proceedings ACL'87.*

Alshawi, H. and D. Crouch: 1992, 'Monotonic semantic interpretations'. In *Proceedings ACL'92*, pp. 33–39.

Bos, J.: 1995, 'Predicate Logic Unplugged'. In *Proceedings of the 10th Amsterdam Colloquium*, Amsterdam: ILLC, pp. 133–142.

Bos, J.: 2002, 'Underspecification and resolution in discourse semantics'. Ph.D. Thesis, Saarland University, Saarbrücken.

Bronnenberg, W., H. Bunt, J. Landsbergen, R. Scha, W. Schoenmakers & E. van Utteren: 1979, The question answering system PHLIQA1. In L. Bolc (ed.) *Natural language question answering systems.* London: McMillan, pp. 217–305.

Bunt, H.: 1984, 'The resolution of quantificational ambiguity in the TENDUM system.' *Proceedings COLING 1984*, Stanford University, pp. 130–133.

Bunt, H.: 1985, *'Mass terms and model-theoretic semantics.'* Cambridge University Press.

Bunt, H.: 1995, 'Semantics and Pragmatics in the ΔELTA Project.' In: L. Dybkjaer, editor, *Proceedings of the Second Spoken Dialogue and Discoure Workshop*, Topics in Cognitive Science and HCI, Vol.8. Roskilde: Centre for Cognitive Science, pp. 1–27.

Bunt, H.: 2005, 'Quantification and modification as Feature Structures'. In *Proceedings of the Sixth Internationanl Workshop on Computational Semantics IWCS-6*, Tilburg, pp. 54–65.

Bunt, H.: 2006: 'Mass expressions'. In K.Brown, editor, *Encyclopedia of Language and Liguistics, Second Edition.* Amsterdam: Elsevier, pp. 5757–5760.

Bunt, H., R.J. Beun, F. Dols, J.v.d. Linden & G. Schwartzenberg: 1984, 'The TENDUM dialogue system and its theoretical foundations.' *IPO Annual Progress Report* 19, 105–113, Eindhoven: IPO.

Bunt, H. & R. Muskens: 1999, 'Computational Semantics'. In H. Bunt & R. Muskens (eds.) *Computing Meaning, Vol. 1.* Dordrecht: Kluwer, 1–32.

Cooper, R.: 1983, *'Quantification and Syntactic Theory.'* Dordrecht: Reidel.

Copestake, A., D. Flickinger, R. Malouf, I. Sag & S. Riehemann: 1995, 'Minimal Recursion Semantics'. Unpublished ms., CSLI, Stanford University.

Copestake, A. and D. 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, Greece.

Dalrymple, M.: 2001. *'Lexical-Functional Grammar'.* San Diego, Calif.; London : Academic Press.

Davidson, D.: 1967, 'The Logical Form of Action Sentences'. In: N. Rescher (ed.): *The Logic of Decision and Action.* Pittsburgh: University of Pittsburgh Press, pp. 81–95.

Deemter, K. van: 1996, 'Towards a logic of Ambiguous Expressions'. In S. Peters and K. van Deemter, editors, *Semantic Ambiguity and Underspecification.* Stanford: CSLI, pp. 203–237.

Deemter, K. van, G. Brockhoff, H. Bunt, M. Meya and J. de Vet: 1985 'From TENDUM to SPICOS, or: How flexible is the TENDUM approach to question answering?' *IPO Annual Progress Report 20*, 83–90.

Ebert, C.: 2005, *Formal Investigations of Underspecified Representations.* Ph.D. Thesis, King's College, University of London.

Egg, M., A. Koller and J. Niehren: 2001, 'The constraint language for lambda structures'. *Journal for Logic, Language, and Information* 10, 457–485.

Fenstad, J.E., P. Halvorsen, T. Langholm, and van Benthem: 1987, 'Situations, Language and Logic'. Studies in Linguistics and Philosophy 34, Reidel, Dordrecht.

Fraassen, B. van: 1966, 'Singular terms, truth-values gaps, and free logic'. *Journal of Philosophy* 63.17: 481–495.

Gardent, C. and B. Webber: 1998, 'Describing discourse semantics.' In *Proceedings of the 4^{th} TAG+ Workshop, Philadelphia.*

Geurts, B. & G. Rentier: 1991, 'Quasi logical form in PLUS'. Esprit project P5254 (PLUS) internal report, Tilburg: Institute for Language Technology and Artificial Intelligence ITK.

Hobbs, J.: 1985, 'Ontological Promiscuity.' In *Proc. 23rd Annual meeting of the ACL*, Chicago, 61–69.

Hobbs, J. and S. Shieber: 1987, 'An algorithm for generating quantifier scopings'. *Computational Linguistics* 13 (1–2):47–63.

Hobbs, J., M. Stickel, D. Appelt & P. Martin: 1993, 'Interpretation as Abduction'. *Artificial Intelligence* 63: 69–142.

Keller, W.: 1998, 'Nested Cooper Storage: The Proper Treatment of Quantification in Ordinary Noun Phrases'. In: U. Reyle & C. Rohrer (eds.) *Natural Language Parsing and Linguistic theories.* Dordrecht: Reidel, pp. 1–32.

Kievit, L.: 1998, *'Context-driven natural language interpretation'.* Ph.D. Thesis, Tilburg University.

Kievit, L., P. Piwek, R.J. Beun & H. Bunt: 2001, 'Multimodal Cooperative Resolution of Referential Expressions in the DENK system.' In H. Bunt & R.J. Beun (eds) *Cooperative Multimodal Communication.* Heidelberg: Springer, pp. 197–214.

Koller, A., J. Niehren, and S. Thater: 2003, 'Bridging the gap between underspecification formalisms: Hole semantics as dominance constraints.' *Proceedings of the 11^{th} EACL*, Budapest, pp. 195–202.

Koller, A.: 2004, 'Constraint-Based and Graph-Based Resolution of Ambiguities in Natural Language'. Ph.D. Thesis, Saarland University, Saarbrücken.

König, E. & Reyle, U.: 1996, 'A General Reasoning Scheme for Underspecified Representations.' In: H.J. Ohlbach & U. Reyle (eds.) *Logic and its Applications. Festschrift for Dov Gabbay.* Dordrecht, Kluwer, pp. 1–28.

Medema, P., W. Bronnenberg, H. Bunt, J. Landsbergen, W, Schoenmakers and E. van Utteren: 1975, 'PHLIQA1: 'Multilevel semantics in question answering.' *American Journal of Computational Linguistics* microfiche 32.

Molla, D.: 1999, 'Ontologically Promiscuous Flat Logical Forms for NLP'. In *Proceedings 4th International Workshop on Computational Semantics IWCS-4*, Tilburg University, pp. 249–265.

Muskens, R.: 2001, 'Talking about Trees and Truth-Conditions.' *Journal of Logic, Language and Information* 10: 417–455.

Pinkal, M.: 1999, 'On semantic underspecification.' In H. Bunt & R. Muskens (eds.) *Computing Meaning, Vol. 1.* Dordrecht: Kluwer, pp. 33–55.

Pulman, S.: 2000, 'Bidirectional Contextual resolution.' *Computational Linguistics* 26:4. 497–538.

Reyle, U.: 1993, 'Dealing with Ambiguities by Underspecification'. *Journal of Semantics* 10 (2), 123–179.

Reyle, U.: 1996, 'Co-indexing labelled DRSs to represent and reason with ambiguities'. In S. Peters and K. van Deemter, editors, *Semantic Ambiguity and Underspecification.* Stanford: CSLI, pp. 239–268.

Scha, R.: 1981, 'Distributive, collective and cumulative quantification'. In J. Groenendijk and M. Stokhof, editors, *Formal methods in the study of language.* Amsterdam: Mathematical Centre.

Schiehlen, M., Bos, J. & Dorna, M.:2000, 'Verbmobil Interface Terms (VITs).' In W. Wahlster (ed.) *Verbmobil: Foundations of Speech-to-Speech Translation.* Berlin: Springer, pp. 183–199.

Schubert, L.K. and F.J. Pelletier: 1982, 'From English to Logic: Context-free computation of conventional logical translation'. *American Journal of Computational Linguistics* 8 (1): 165 –176.

Willis, A. & S. Manandhar: 2001, 'The Availability of Partial Scopings in an Underspecified Semantic Representation'. In H. Bunt, R. Muskens & E. Thijsse (eds.) *Computing Meaning, Vol. 2*, Dordrecht: Kluwer, pp. 129–145.

Woods, W.: 1978, 'Semantics and quantification in question answering/. In M. Yovits (ed.) *Advances in Computers.* New York: Academic Press, pp. 2–64.