Semantic construction for reinterpretation phenomena*

Markus Egg, Universität des Saarlandes

e-mail: egg@coli.uni-sb.de

Abstract

Reinterpretation is a 'creative' or 'dynamic' aspect of natural language, where the literal meaning of an expression is augmented with the help of extralinguistic knowledge, in particular, in cases where there is no well-formed literal interpretation of the expression. Instances of reinterpretation call for a careful design of semantic construction mechanisms that leave enough leeway for the contributions of these additional knowledge sources.

I propose the following account of reinterpretation. Semantic construction yields underdetermined, hence, ambiguous structures for reinterpretation cases, which are then monotonically enriched with information from extralinguistic sources (especially world knowledge). Semantic ambiguities are described by a suitable underspecification formalism that integrates the treatment of reinterpretation into a general approach to structural ambiguity.

*The research presented in this paper was supported by the Deutsche Forschungsgemeinschaft in the project CHORUS within the SFB 378 „Ressourcenadaptive kognitive Prozesse“. Thanks to my colleagues for comments and discussion, in particular to Alexander Koller and Kristina Strignitz for their feedback on a first version of this paper.

1 Introduction

The topic of this paper is *reinterpretation*. Reinterpretation arises in particular in cases where there is a semantic incompatibility between the immediate subparts of a larger constituent. (Other instances of reinterpretation are discussed in Egg 2000.) Cases of aspectual reinterpretation like (1) exemplify this phenomenon: the aspectual properties of the modifier *for ten days* (semantically, the functor) and the modified expression (the sentence radical[1] *Amélie play- the sonata*; semantically, the argument) are at variance: the argument is bounded,[2] but the functor needs an unbounded argument.

(1) # Amélie played the sonata for ten days

Nevertheless, the sentence is acceptable. This is due to reinterpretation: in (1), the modified sentence radical can be reinterpreted *iteratively* (i.e., Amélie played the sonata repeatedly). This makes the sentence radical unbounded, which fits the restriction of the modifier. Sentences that are acceptable only after reinterpretation are marked by a 'hash' (#) sign.

Thus, reinterpretation is a feature of natural language that introduces a certain amount of flexibility into generating and understanding complex expressions. It must be integrated into a formal semantic framework. This is the goal of the analysis of reinterpretation that I am going to propose. Thus, the present paper pursues the same line of research as papers like Dölling (1992, 1995), Pustejovsky (1995), Moens and Steedman (1988), Pulman (1997), and de Swart (1998).

The proposed analysis is based on a view of reinterpretation where the full meaning of an expression that is to be reinterpreted (called a *reinterpretation candidate*) is determined by the *interaction of various knowledge sources*, which includes, but is not restricted to, the syntax and semantics of the expression. E.g., the iterative operator in the interpretation of (1) is provided by world knowledge that only the typical duration of an *iteration* of sonata playings is compatible with a duration of ten days. Since semantic construction cannot fully determine the meaning of a reinterpretation candidate, it must yield underdetermined, hence, *ambiguous semantic representations* for these expressions. These ambiguities 'leave room' for other knowledge sources to contribute to the meaning of the reinterpretation candidate. Integrating the material from the relevant other knowledge sources then disambiguates these representations. Thus, reinterpretation emerges as a *monotonic* process, not as a destructive process of undoing results of semantic construction.[3]

Consider a very simplistic example for the proposed analysis of reinterpretation. (It serves expository purposes only, see section 4.3.) The ambiguous (2a) models the semantics of (1). The dots indicate a position between functor and argument that is not determined by semantic construction and allows integrating information from other sources, e.g., an iterative operator. This results in (2b), which models the meaning of (1) after reinterpretation, viz., the proposition that Amélie played the sonata repeatedly for ten days:

(2) (a) $\mathbf{for\_10days}(\ldots\mathbf{play}(\mathbf{am\acute{e}lie}, \mathbf{the\_sonata}))$

(b) $\mathbf{for\_10days}(\mathrm{ITER}(\mathbf{play}(\mathbf{am\acute{e}lie}, \mathbf{the\_sonata})))$

The semantic ambiguities that emerge in the analysis of reinterpretation candidates are described non-disjunctively in the framework of *underspecification*: semantic representations do not give the meaning of an expression, but merely *descriptions* of its meaning. Selecting a reading of a reinterpretation candidate is then modelled as a monotonic process of adding further descriptions, which accords with the view on reinterpretation that is advocated in this paper.

The paper is structured as follows: After an exposition of patterns of reinterpretation in section 2, section 3.1 introduces the framework of underspecification. Section 3.2 outlines the representation language CLLS (Constraint Language for Lambda Structures), which is used for the analysis of reinterpretation. The main part of the paper is then section 4, where the analysis and the construction of reinterpretation candidates is presented in detail. I conclude with a brief outlook on further lines of research in section 5.

2 Reinterpretation phenomena

After the more general introduction of the phenomenon of reinterpretation in the preceding section, this section is devoted to an analysis of different patterns of reinterpretation and of the different knowledge sources that interact in reinterpretation.

2.1 Type coercion vs. landing site coercion

The first pattern of reinterpretation is exemplified in (1) and (3). (1) has already been expounded in section 1: reinterpretation uses an operator (preferredly, the iterative operator) that maps the bounded argument onto an unbounded expression. This expression then meets the selection restriction of the functor.

In (3), argument (the semantics of the verb, a property of *substances*) and functor (the semantics of the NP, a property of properties of *things*) do not fit, since the relevant sorts SUBST[ance] and THING are disjoint:

(3) # Every bottle froze

This sortal mismatch is resolved by an operator that maps the argument onto a property of things. The preferred operator for (3) introduces the notion of *content*. It is based on world knowledge about the use of bottles as containers for liquids. The resulting reading is that the content of every bottle froze.

Examples (1) and (3) follow the same reinterpretation pattern: to mediate between conflicting properties of functor and argument that are to be combined, an operator is applied to the argument first. This operator is not part of either functor or argument, hence, additional material - beyond the one provided by semantic construction - must be derived and integrated with the functor-argument pair. If F stands for the functor, A, for the argument, and Op, for the additional material, the general pattern of type coercion is $F(Op(A))$. This pattern is familiar from the literature, it has been dubbed *type coercion* by Moens and Steedman (1986, 1988). They discuss type coercion in the aspectual domain (as well as Pulman 1997 and de Swart 1998), while Dölling (1992, 1995) and Pustejovsky (1995) focus on type coercion in the nominal domain.

But reinterpretation cases like (4) call for a different explanation:

(4) # Amélie left for two hours

(4) exhibits the same aspectual incompatibility as (1). But the preferred way of reconciliating the conflicting properties of functor (the semantics of *for two hours*) and argument (the semantics of *Amélie leave-*) in (4) is different: the functor is applied not to the argument as a whole but to only a *part* of it. Reinterpretation augments the argument so that it comprises a part to which the functor may be applied alternatively (an additional 'landing site' for the functor).

For (4), such an additional part of the argument is introduced that refers to an *aftermath* associated with eventualities (this term refers to all kinds of states of affairs) in the extension of the argument. Since the property of being an aftermath of some other eventuality is unbounded, this new part of the argument fits the aspectual selection restriction of the adverbial. The resulting interpretation of (4) is that it is not Amélie's leaving but the aftermath of her leaving (her subsequent absence) that lasted for two hours.

This reinterpretation is based on conceptual knowledge. Following Moens and Steedman, it is assumed that conceptually the flow of eventualities is carved in small portions, called 'nuclei'. Nuclei consist of an eventuality together with associated preparatory phase and aftermath. These additional eventualities can be referred to in reinterpretation operators.

The abstract pattern of this kind of reinterpretation is $A \,\&\, a_1(F(a_2))$, where $a_1(a_2)$ is the augmentation of the argument A that introduces the additional landing site a_2 for the functor F ('&' is the generalized conjunction of Rooth and Partee 1982 that conjoins any pair of expressions that have the same type).

The pattern is called *landing site coercion.*

Landing site coercion is not restricted to the aspectual domain. It is shown in section 4 that the Pustejovsky example *fast road* also instantiates this phenomenon. Since roads (being immobile things) are themselves not fast, *fast road* must be understood differently, e.g., as 'road along which one can drive fast', where driving along the road is its preferred (but not its only possible) purpose.

In this subsection, two patterns of reinterpretation were distinguished, viz., type coercion and landing site coercion. An adequate account of these patterns is a first challenge for an analysis of reinterpretation.

2.2 Semantic and non-semantic parts of reinterpretation

The interpretation of reinterpreted expressions involves the integration of additional material. Its origin is neither functor nor argument, hence, syntax and semantics of reinterpretation expressions do not provide it. I.e., syntax and semantics cannot fully determine the meaning of these expressions, which brings forth ambiguity on the semantic level. Additional knowledge sources have to complement linguistic knowledge in order to obtain an unambiguous meaning for these expressions.

Reinterpretation thus instantiates the more general phenomenon that the meaning of some expressions can only be determined by the interaction of various knowledge sources. But due to the interaction between these knowledge sources, few examples illustrate this point clearly. Language understanding hardly ever proceeds in zero contexts, in particular, one cannot shut off world knowledge, which often immediately resolves ambiguities that arise in reinterpretation. E.g., nothing in the semantics of (3) bars one from understanding it in the sense of 'the substance constituting the bottle froze', but world knowledge makes one strongly prefer the 'container' reading over this reading. Similarly, knowledge about the typical duration of sonatas immediately triggers the choice of the iterative operator in the understanding of (1).

Examples that are clearly ambiguous are those with more than one equally plausible reinterpretation. For (5), one cannot decide out of context whether the argument is reinterpreted by inserting the progressive or the iterative operator. Both options meet the aspectual requirements of functor and argument. In contrast to (1), the modifier in (5) is too vague to contribute to the disambiguation. The Bierwisch (1983) style example (6) is analogously ambiguous in that reinterpretation may involve the *writings* or the *thoughts* of Faulkner.[4]

(5) # Amélie played the sonata for some time

(6) # It is difficult to understand Faulkner

These examples show that even if functor and argument in reinterpretation cases are not ambiguous, ambiguity arises as soon as they are combined. The resolution of this ambiguity is then the task of extralinguistic knowledge sources. Therefore, the semantic boundary must be delimited carefully in analyses of reinterpretation: the semantic representation of reinterpretation expressions must leave room for the contribution of additional knowledge sources to the full determination of their meaning. The method of representing these ambiguities

I adopt in this paper is *underspecification*, which is the topic of the following section.

3 The formal background

In the first part of this section, the notion of underspecification formalism is introduced along with its application to the task of accounting for ambiguity in natural language. The second part presents the chosen representation language CLLS, in particular, the adapted version of CLLS that is used for the account of reinterpretation in this paper.

3.1 Underspecification formalisms

The proposed analysis regards reinterpretation candidates as ambiguous on the semantic level. These ambiguities are captured in terms of *underspecification*. Underspecification formalisms were designed for the representation of semantically underdetermined expressions, i.e., expressions whose meaning can only partially be determined by semantic construction. As an example, consider (7), which exhibits quantifier scope ambiguity, since either quantifier may outscope the other one:

(7) Every woman loves a man

This sentence has two readings, glossed here as '$\forall\exists$' (for every woman, a [possibly different] man) and '$\exists\forall$' (one single man loved by every woman). An assertion of (7) is understood as $\forall\exists \vee \exists\forall$, a denial of (7), as $\neg\forall\exists \vee \neg\exists\forall$. Now these two interpretations could not be captured if one simply determined the meaning of the ambiguous (7) as the disjunction of its possible disambiguations: in that case, one would falsely obtain $\neg(\forall\exists \vee \exists\forall)$ as the interpretation of the denial of (7).[5] However, if one regards the meaning of the ambiguous expression as the set of its fully specified readings, and assumes that understanding such an expression proceeds by forming the disjunction of this set, these interpretations follow directly. For (7), the meaning is thus $\{\forall\exists, \exists\forall\}$. The assertion of (7) 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 interpretations (van Eijck and Pinkal 1996).

But, for more complex ambiguous expressions, generating and enumerating all their readings may become a very involved task, which is neither efficient nor cognitively adequate. Thus, to characterize the set of possible readings for ambiguous expressions without having to enumerate the readings, a number of formalisms were proposed that provide a compact description of the range of these readings in an *underspecified* representation. Examples for such formalisms are Quasi-Logical Form (QLF, Alshawi 1992), Underspecified Discourse Representation Theory (UDRT, Reyle 1993), Ambiguous Predicate Logic (Jaspars and v. Eijck 1996), Glue Language Semantics (Dalrymple et al. 1997), MRS (Minimal Recursion Semantics, Copestake et al. 1997), Underspecified Semantic Description Language (USDL, Pinkal 1996), or Constraint Language for Lambda Structures (CLLS, Egg et al. 1998; Egg et al. 2001).

These formalisms augment semantic representations with metavariables and subordination relations between fragments of such representations. E.g., a description of the semantics of (7) in such a formalism introduces the two quantifiers and the fact that they both subordinate (have scope over) the verb semantics (a two-place relation), and that the arguments of the verb semantics are variables that are bound by the quantifiers. The scope relation between the two quantifiers is left open. Ambiguity resolution can then be modelled as augmenting such descriptions, for (7), by information on the quantifier scope. This augmentation chooses one of the two feasible scope relations between the quantifiers. See (9) below for the formal account of the semantics of (7).

So far, analyses in underspecification formalisms focus on ambiguities arising from quantifier scope. Recently, they were extended to additional phenomena, e.g., ambiguities that arise for elliptical expressions. Reinterpretation is another field which crucially involves ambiguity (recall e.g. the leeway in interpreting (5), hence, it seems reasonable to apply underspecified approaches to this field, too. It is the goal of this paper, where reinterpretation is described in terms of the underspecification formalism CLLS, to show that this strategy proves fruitful.

Using an underspecification formalism for an account of reinterpretation has the additional advantage that one gets for free an analysis of scope ambiguity. Such an analysis is necessary since an account of reinterpretation must be able to model the interdependence between reinterpretation and issues of quantifier scope.

In this respect, the proposed underspecified account of reinterpretation is slightly more complex than - though similar in spirit to - approaches like

Dölling's (1995) or Pulman's (1997). They insert free variables between functor-argument pairs during semantic composition, which can be instantiated afterwards by suitable additional (conceptual) knowledge if necessary.

However, reinterpretation in its entirety cannot be captured with such an approach. This can be illustrated particularly well with two observations on the phenomenon of reinterpretation that show that the additional complexity of my approach is indispensable for an analysis of reinterpretation. For either phenomenon, an appropriate description would have to extend the expressive power of the frameworks of Dölling and Pulman.

The first observation is that reinterpretation and phenomena like quantifier scope may cooccur in a linguistic expression. As an example for such an expression consider (8), a transitive variant of example (3), where both subject and object NP introduce a quantifier. (8) exhibits not only the same sortal mismatch as (3) (between the sorts SUBST and THING), but there is also scope ambiguity between its two NPs:

(8) # A man froze every bottle

The semantic ambiguities that these phenomena introduce are independent of each other. One can perform reinterpretation operations on the resulting representations or determine scope relations between their quantifiers without affecting any ambiguities that arise from the respective other source (see Koller, Niehren, and Striegnitz 1999).

The cooccurrence of these phenomena calls for an integrated treatment of such kinds of semantic ambiguity, if one wants to be able to describe all cases of reinterpretation. But then an appropriate approach to reinterpretation must be expressive enough to cover scope ambiguities, too, i.e., its degree of complexity cannot fall below the one of the approach to reinterpretation that is proposed in this paper.

The second observation pertains to the phenomenon of landing site coercion. The analysis of this phenomenon as presented in section 2.1 turns out to require the very same kind of expressivity of the semantic description formalism that is needed for quantifier scope ambiguity: the semantic structure that results from landing site coercion is the same as for these scope ambiguities, which will be expounded in detail in section 4.3. Underspecification formalisms are hence well equipped for the description of landing site coercion, since they are of the requested expressivity (after all, quantifier scope ambiguities are *the* prototypical domain of application for these formalisms).

This concludes the brief sketch of underspecification formalisms, I will now move over to the the representation language that is used for the account of reinterpretation in this paper in the next subsection.

3.2 The representation language CLLS

The language CLLS (Constraint Language for Lambda Structures; Egg et al. 1998; Egg et al. 2001) was designed for classic cases of structural ambiguity, especially those that come with scope underspecification and/or ellipsis reconstruction. In this subsection I introduce CLLS in an adapted form, which suffices for the purpose of this paper. For the full information on CLLS, the reader is referred to the two papers cited above; the relation between the adapted and the full CLLS notation is discussed in chapter 3 of Egg (2000).

In CLLS, semantic representations are *constraints* on meanings. CLLS representations are based on *tree structures* that correspond to λ-terms. This basis makes CLLS a very powerful and flexible formalism.

In the adapted notation, CLLS constraints describe sets of λ-terms. They have four main ingredients:

- *fragments* of λ-terms ('LT fragments')
- not yet specified parts of LT fragments, which can be equated as a whole or in part with other fragments; they are called '*holes*' and are depicted by '$\Box$'
- *dominance* (or part-of) relations between holes and LT fragments, which are indicated by dotted lines[6]
- sort information, which is given by sorted variables

Holes and dominance relations introduce underspecification. Imagine a constraint where a hole in an LT fragment dominates another LT fragment. This expresses only that the first fragment outscopes the second. (Or, to put it differently: the second fragment is an (im-)proper part of the material represented by the hole in the first fragment.) Other fragments of the constraint or new material not yet present in the constraint may intervene scopally between the two fragments. Due to this possibility of integrating new material into constraints, which is crucial for the description of reinterpretation phenomena, the constraints are not partial orders of semantic fragments like e.g. in UDRT.

(9), the semantic representation of (7), illustrates this adapted CLLS notation. This constraint captures the common ground between the two readings of the sentence:

$$\Box$$

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

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

Such structures with holes and dominance relations provide partial descriptions of λ-terms. (9) introduces fragments for the two quantifiers. The scope of either quantifier in (9) is described by a hole, since it is not yet fixed. All that one knows right from the start is that the verbal nucleus (the bottom fragment $\mathbf{love'}(x, y)$) is part of either scope.[7] Similarly, the hole at the top shows that the semantics of the whole sentence is still open, but either quantifier fragment must be part of it. This 'dominance diamond' appears frequently in semantic representations for scopally ambiguous expressions.

All λ-terms that comply to (9) are *solutions* of the constraint. Solutions that comprise only material that is mentioned explicitly in the constraint are called *constructive*. For (9), there are two constructive solutions. Either quantifier fragment can receive widest scope, which forces the respective other quantifier fragment to fill in the scope of the first one, with the verbal nucleus being the scope of the second quantifier. These options give the λ-terms for the two readings of (9):

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

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

(9) can also be satisfied by λ-structures that contain material not yet mentioned in the description. One could rule out these solutions by a general *constructiveness* restriction on these descriptions. While such a restriction would be appropriate if one was only concerned with scope ambiguities, it would be too strict for my underspecified account of reinterpretation, which crucially involves the integration of additional material into semantic representations.

After this account of underspecification formalisms and the representation language CLLS, the next section is devoted to a detailed account of reinterpretation candidates and their derivation in semantic construction.

4 Representing reinterpretation

In this section, reinterpretation phenomena are analysed in detail after some preliminaries, first type coercion, and next, landing site coercion.

4.1 Preliminaries

Reinterpretation is seen as a monotonic process, where the full meaning of a reinterpretation candidate is derived by enriching a well-formed but not yet fully determined (hence ambiguous) result of semantic construction with material from extralinguistic sources.

To reach this goal, potential reinterpretation sites ('R-sites') must be distinguished. R-sites involve a functor-argument pair. In the cases discussed in this paper, functor and argument can be identified with the semantic contributions of a modifier/subcategorizer and of its modified/subcategorized expression, respectively. Verbs subcategorizing for NPs are exceptional, here the NP provides the functor, and the verb, the argument.

For functor-argument pairs that might be affected by reinterpretation, semantic construction avoids combining potentially conflicting material of functor and argument in one single LT fragment. The resulting semantic representations are constraints that consist of several fragments. Dominance relations connect these fragments. Schematic examples of such constraints are given in (12), where the $\mathbf{fr}_n$ are abbreviations for LT fragments:[8]

(12) (a) $\begin{array}{l} \mathbf{fr}_1(\ \square\ ) \\ \quad\ \vdots \\ \quad\ \mathbf{fr}_2 \end{array}$

(b) $\begin{array}{ccc} & \square & \\ \mathbf{fr}_1(\ \square) & & \mathbf{fr}_2 \end{array}$

This means in the first place that any incompatibilities between functor and argument cannot make the result of semantic construction ill-formed, since they are buffered by dominance relations. Semantic construction can proceed without having to bother about potential functor-argument mismatches.

Second, the dominance relations between the LT fragments allow additional material to show up in solutions for the constraints for reinterpretation candidates. Since there is always more than one solution of this sort, these constraints are compatible with more than one λ-term, hence, an underspecified representation of a semantic ambiguity.

Furthermore, the dominance relations between the LT fragments identify for reinterpretation candidates the positions where the material that is to be contributed by additional, extralinguistic sources must be integrated to fully specify the meaning of the expression. Thus, semantic construction strictly restricts itself to the semantic parts of such a meaning and leaves enough space for extralinguistically derived material. The integration of this additional information is hence monotonic, it need not undo results of semantic construction.

R-sites are determined in the syntax-semantic interface. The proposed approach shares this basic strategy with the analyses in Pulman (1997) and de Swart (1998). I.e., specific syntactic phrase structure rules are associated with reinterpretation potential in that they relate the semantic contributions of the involved immediate constituents only in terms of dominance relations. I will illustrate this with the reinterpretation candidate (3).

The general structure of the syntax-semantics interface is that phrase structure rules for complex constituents are coupled with constraints that relate the semantic contributions of the respective immediate constituents. These contributions (also in the form of constraints) are made available for the phrase structure rule e.g. as the value of some feature CONT[ent]|CON[straint] in an attribute-value matrix.

The constraint for a complex constituent then is simply the conjunction of the constraints for its immediate constituents with the constraint that is coupled with the relevant phrase structure rule.

The rules that are used in the construction of the semantics of (3) are given in (13)-(15) together with the corresponding syntax rule (in a simplified form that neglects the question of how to derive the 'top' holes in the dominance diamonds). In these rules, '$[\![\mathtt{XP}]\!]$' means 'semantic representation of constituent

XP'. In addition, the prefix '$[\![\texttt{XP}]\!]$:' before a fragment in the semantic representation means that this fragment is the main fragment in a complex constraint, which is addressed in semantic composition processes at subsequent steps in the derivation:

(13) $[_{\texttt{VP}}\ \texttt{V}]$ $\overset{\text{(SSI)}}{\Rightarrow}$ $[\![\texttt{VP}]\!] : \square$ … $[\![\texttt{V}]\!]$

(14) $[_{\texttt{NP}}\ \texttt{Det}\ \bar{\texttt{N}}]$ $\overset{\text{(SSI)}}{\Rightarrow}$ $[\![\texttt{Det}]\!]([\![\bar{\texttt{N}}]\!])(\lambda x.\ \square\ )$ … $[\![\texttt{NP}]\!]\text{: } x$

(15) $[_{\texttt{S}}\ \texttt{NP}\ \texttt{VP}]$ $\overset{\text{(SSI)}}{\Rightarrow}$ $[\![\texttt{S}]\!]\text{: } [\![\texttt{VP}]\!]([\![\texttt{NP}]\!])$

The crucial effect of these rules is that, in the semantic representation of (3), the semantic contribution of the verb is set off in a fragment of its own due to (13). Thus, any sortal restrictions that the verb might impose on its subject are buffered; the constraint is not ill-formed if the subject fails to comply to these restrictions.

Note also that rule (14) introduces a dominance relation into the semantic contribution of an NP: it sets off a (dominating) fragment that comprises the respective quantifier (such fragments are called *N-fragments*) from the NP fragment proper, which consists merely of a bound variable. This dominance relation is crucial for the modelling of scope ambiguities as in (9), since it allows other N-fragments to be dominated by the N-fragment of the subject NP.

This grammar fragment returns the semantic representation (16) for sentence (3) (ignoring matters of tense and sentence mood, hence, also not introducing eventuality arguments of any kind) on the basis of standard semantic representations for *every* ($\lambda Q \lambda P \forall x.Q(x) \to P(x)$), *bottle* ($\lambda x_{\textsc{thing}}.\mathbf{bottle}'(x)$), and *freeze* ($\lambda y._{\textsc{subst}}\mathbf{freeze}'(y)$). The incompatibility of the sorts THING and SUBST(ance) models the sortal mismatch between NP and verb:

(16)
$$\begin{array}{rl} \forall x_{\textsc{thing}}.\mathbf{bottle}'(x) \to & \Box \\ & \Box\ (x) \\ \lambda y_{\textsc{subst}}. & \mathbf{freeze}'(y) \end{array}$$

Since (16) is the representation of an expression with only one quantifier, its upper dominance relation is of no avail. Hence, to simplify matters, I ignore it from now on (i.e., the relation of (im-)proper part between the hole of the top fragment and the fragment in the middle is specified as identity), which yields the constraint (17), an instance of the pattern (12a). Here the scope of the functor (the semantics of the NP) is not yet determined, hence given in terms of a hole. The argument (the verb semantics) is part of the scope:

(17)
$$\begin{array}{rl} \forall x_{\textsc{thing}}.\mathbf{bottle}'(x) \to & \Box\ (x) \\ \lambda y_{\textsc{subst}}. & \mathbf{freeze}'(y) \end{array}$$

(17) indicates that reinterpretation must take place, since it has no constructive solution, due to the incompatible sorts in the two fragments. Yet this incompatibility does not render the constraint ill-formed (in that it has no solutions at all), since the dominance relation 'buffers' the incompatible sorts.

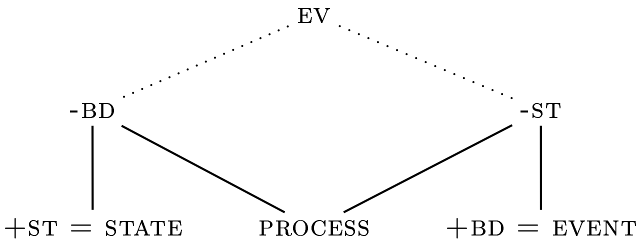
After these general comments on the derivation of the semantic representations for reinterpretation candidates, I will wind up this subsection with some remarks on the representation of aspect, before I can embark on the description of type coercion and landing site coercion.

Aspect is encoded in sorts of eventualities: different aspectual classes are modelled as incompatible sorts of the main eventuality arguments of predicates. The term 'predicate' refers to verbs and their projections (e.g., VPs or sentence radicals). In particular, *boundedness* and *unboundedness* are distinguished by the sorts ±BD.

This encoding of aspect has the consequence that no overlap is assumed between the eventualities in the extension of an unbounded expression and those in the extension of a bounded expression, although eventualities of different aspectual classes may spatiotemporally coincide. For instance, no entity can be in the extension of both *Amélie smile-* and *Amélie smile- for two hours*, since only the latter predicate is bounded. This kind of distinction reappears in the nominal domain, where e.g. groups and the pluralities of individuals that constitute them are distinguished in spite of their material identity.

The very simple aspectual classification that is used for the purpose of this paper[9] is based on the aspectual properties of boundedness and of *stativity*. Stative predicates present eventualities as involving no change of state; stativity and non-stativity are distinguished by the sorts ±ST. Stativity implies unboundedness. This groups unbounded predicates into (stative) *state predicates* (e.g., *to love*, *to be in the pub*) and (non-stative) *process predicates* (e.g., *to run*, *to shake*).[10]

In the sort hierarchy, eventualities (individuals with the sort EV) are cross-classified (this is indicated by the dotted lines) into unbounded and non-stative eventualities. Process eventualities emerge as the intersection of unbounded and non-stative eventualities:



Finally, the operators that show up in the following are defined. As usual, eventualities are represented within λ-terms (and in constraints on λ-terms) by variables $e, e', \ldots$ (rather than by the clumsier $x_{\mathrm{EV}}, x'_{\mathrm{EV}}, \ldots$).

First, the *progressive operator* PROG is defined. PROG(P) holds for e iff there is an eventuality e' for which P holds, and the spatiotemporal extensions of e and e' stand in the part-of relation. In other words, e' is ongoing at e. This definition ignores the so-called 'imperfective paradox' (which is of no avail for the present paper): assuming the existence of such an e' neglects the fact that PROG(P) may hold for an e even if the respective e' in the extension of P is never completed in the real world. PROG selects for non-stative predicates and returns a stative one.

The *maximality operator* MAX maps predicates P onto the bounded predicate of being a 'maximal phase' of P (Herweg 1991). MAX(P) holds for an eventuality e iff P holds for an e' that spatiotemporally coincides with e, but for no e'' of which e' is a proper part.

The *genericity operator* GEN$[e](R, C)$ is a quantifier over eventualities e, such that R generally implies C (see Krifka et al. 1995 for detailed discussion of GEN).

The semantics of durative adverbials is rendered in terms of the *S-quantity* *operator* PH. PH$(\mathbf{Q}, P)$ holds for an eventuality e iff there is an e' in the extension of P that spatiotemporally coincides with e, and a suitable measure function returns $\mathbf{Q}$ as the value of the length of e (see Egg 1995 for further discussion of the S-quantity). The argument of the durative adverbial must be unbounded, the resulting expression is bounded.[11]

4.2 Representing type coercion

Once R-sites for type coercion have been determined within semantic representations for reinterpretation candidates by the syntax-semantic interface, they are used for the integration of additional material into the semantic representation. Schematically, this is depicted in (18). We set out from (18a), the result of semantic construction, a structure in which functor and an argument are separated by a dominance relation. This dominance relation determines the position where additional material (in the form of an operator Op) must be integrated (18b). This material is added to the semantic representation (18c). Finally, this new representation has a constructive solution, the λ-term (18d):

(18) (a) $\begin{array}{c} F(\Box) \\ \vdots \\ A \end{array}$ (b) $\begin{array}{c} F(\Box) \\ \vdots \\ A \end{array} \Leftarrow Op(\Box)$

(c) $\begin{array}{l} F(\Box) \\ \quad\vdots \\ \quad Op(\Box) \\ \qquad\vdots \\ \qquad A \end{array}$ (d) $F(Op(A))$

I will flesh out this abstract schema with an account of the derivation of the preferred reading of (3) (that the content of every bottle froze). Semantic construction yields the constraint (17), an R-site for type coercion. (17) reappears on the left hand side of (19), which describes the augmentation of the result of semantic construction for (3):

(19)
$$\begin{array}{rl} \forall x_{\mathrm{THING}}.\mathbf{bottle}'(x) \rightarrow \square\ (x) & \\ \vdots\ & \Leftarrow \quad \lambda z_{\mathrm{THING}}\exists y_{\mathrm{SUBST}}.\mathbf{CONT}(z,y)\wedge\ \square\ (y) \\ \lambda y_{\mathrm{SUBST}}.\mathbf{freeze}'\ (y) & \end{array}$$

Since (17) has no constructive solution (this would impose conflicting sortal requirements onto the quantified variable), additional material, e.g., the right hand side constraint in (19), must be integrated at the position indicated by the dominance relation. The right hand side constraint introduces an operator that maps a property P of substances y onto a property of entities of sort THING, viz., the property of being a container of a substance for which P holds. E.g., if P is the property of freezing, we obtain the property of being a container whose content is freezing. At the heart of the reinterpretation operator is the relation **CONT** between containers and their content.

The combination of the two constraints in (19) has a constructive solution, viz., the λ-term (20). The λ-term expresses the desired interpretation of (3) that the content of every bottle froze.[12]

(20) $\forall x_{\mathrm{THING}}.\mathbf{bottle}'(x) \rightarrow \exists y_{\mathrm{SUBST}}.\mathbf{CONT}(x,y) \wedge \mathbf{freeze}'(y)$

The predetermination of R-sites in the syntax-semantics interface prevents overgeneration: resolving the sortal conflict in (3) by deriving an R-site between the semantics of the verb and the one of the NP (rather than between determiner and noun semantics) correctly predicts that the universal quantification in (3) is on bottles and not on their contents.

Aspectual type coercion is illustrated by (21), which features an incompatibility between the progressive, which selects for nonstative predicates, and its stative argument *Amélie be- silly*.

(21) # Amélie was being silly

For this type of example, we assume the interface rule (22), which states that the semantics of an auxiliary merely dominates the fragment of its argument (the semantics of a sentence radical):

(22) $[_{\mathtt{VP}_1}\ \mathtt{Aux}\ \mathtt{VP}_2] \overset{\text{(SSI)}}{\Rightarrow} [\![\mathtt{VP}_1]\!] : [\![\mathtt{Aux}]\!](\ \square\ )$; $\square$ dominates $[\![\mathtt{VP}_2]\!]$

The semantics of the sentence radical of (21) is given by the left hand side in (23). Just like in the case of (3), it is an R-site, where the functor (here, the progressive operator) provides the dominating fragment, the argument (here, the semantics of *Amélie be- silly*), the dominated one.

(23) $\lambda e_{+\mathrm{ST}}.\mathrm{PROG}(\lambda e'_{\text{-ST}}\ \square\ (e'))(e)$; $\square$ dominates $\lambda e_{+\mathrm{ST}}.\mathbf{silly'}\ (\mathbf{amélie'})(e)$ $\quad\Leftarrow\quad \lambda e''_{+\mathrm{BD}}.\mathrm{MAX}(\ \square\ )(e'')$

Short of imposing conflicting aspectual requirements on the λ-abstracted variable e', there is no constructive solution for the left hand side of (23). The preferred augmentation of this structure is a constraint for the operator MAX, as depicted on the right hand side of (23). It mediates between the conflicting aspectual properties of functor and argument.

Integrating these two constraints yields a new constraint with the constructive solution (24) for the set of eventualities at which maximal phases of Amélie's being silly are ongoing:

(24) $\lambda e_{+\mathrm{ST}}.\mathrm{PROG}(\lambda e'_{+\mathrm{BD}}.\mathrm{MAX}(\lambda e''_{+\mathrm{ST}}.\mathbf{silly}'(\mathbf{am\acute{e}lie}')(e''))(e'))(e)$

For (21), this augmentation models the intuition that - in spite of her silliness in the eventualities referred to in (21) - Amélie is not a silly person in general: if her silliness is bounded, it is only temporary, and hence, none of her traits.

However, this kind of analysis raises the question of how to account for examples in the style of Nunberg's (1993) notorious (25). These examples show that - at least in some professional jargons - additional patterns of type coercion are possible, which introduce a certain coercion potential for nouns. This potential seems to be restricted to nouns that refer to entities highly relevant for the respective context (e.g., dishes for a restaurant context, and musical instruments for a music context).

E.g., in (25), quantification is on customers, not on sandwiches, which calls for an R-site where the $\bar{\mathrm{N}}$ semantics is the argument in order to make room for an operator from properties of food onto properties of its consumers.

(25) # Every ham sandwich complained

The analysis of (25) is given in (26). The incompatible sorts PERSON and FOOD model the relevant distinction between persons and food. Again, the result of semantic construction is on the left, the extralinguistically derived operator, on the right. The left constraint is an R-site whose argument is the $\bar{\mathrm{N}}$ semantics, the right constraint introduces an operator that maps properties of food P onto the property of being the consumer of something in the extension of P:

(26) $$\begin{array}{l} \forall x_{\text{PERSON}}.\ \Box\ (x) \rightarrow \mathbf{complain'}(x) \\ \qquad\qquad\qquad\qquad\qquad\qquad \Leftarrow \quad \lambda z_{\text{PERSON}} \exists y_{\text{FOOD}}.\mathbf{consume'}(z, y) \wedge\ \Box\ (y) \\ \lambda y_{\text{FOOD}}.\ \mathbf{hamsandwich'}(y) \end{array}$$

Combining both constraints yields a new constraint with the constructive solution (27). In prose, every consumer of a ham sandwich complained:

(27) $$\forall x_{\text{PERSON}}.(\exists y_{\text{FOOD}}.\mathbf{consume'}(x, y) \wedge \mathbf{hamsandwich'}(y)) \rightarrow \mathbf{complain'}(x)$$

Augmenting the semantics of (25) like in (19) (by an operator constraint between the NP and the verb semantics) would lead to a well-formed yet wrong analysis with the quantification on ham sandwiches.

But this does not mean that we can do without the R-site between NP and verb semantics in Nunberg-style examples. E.g., (28) shows that type coercion of the verb may occur even in these cases (since *be parked out back* is a property of vehicles). In (28), both noun and verb must be coerced in order to obtain a quantification on *persons* (roughly, the preferred meaning is 'every consumer of a ham sandwich has a car that is parked out back'):

(28) # Every ham sandwich is parked out back

This shows that cases like (25) do not pose counterexamples to the analysis as advocated in this paper, rather, these cases show that the regular pattern of type coercion can be complemented by additional coercion possibilities. While there are some speculations about possible causes for these additional possibilities (e.g., Nunberg 1993 assumes contextual relevance as potential cause), it is still fair to say that further work is necessary to determine the full range of factors that control reinterpretation in these cases and to integrate them into a formal account of reinterpretation.

The approach proposed here (as well as the one of Pulman 1997) hard-wires reinterpretation potential for expressions regardless of whether it actually has type-theoretically or sortally mismatching constituents. I.e., for cases where there is no reinterpretation, R-sites are introduced in the syntax-semantics interface, too. This contrasts with the analysis of de Swart (1998): she proposes restricting the assignment of reinterpretation potential to only those expressions that actually have semantically mismatching constituents. To this end, potential R-sites are checked for conflicts between their immediate constituents.

However, there are two areas where this strategy may turn out problematic. On the one side, expressions whose literal reading would be well-formed may call for reinterpretation, too, to reconcile them with world or background knowledge. E.g., (29) has a well-formed literal reading, yet it is reinterpreted (so that the $\bar{\mathrm{N}}$ of the subject NP refers e.g. to *recordings* of opera singers) to fit in (29) with our knowledge that opera singers are highly unlikely to occupy shelves:

(29) All opera singers are on the top shelf

On the other side, anaphoric references may enforce reinterpretation of sentences that by themselves do not exhibit any semantic mismatches between their constituents:

(30) (a) The ham sandwich was nice.

(b) He gave me a big tip.

E.g., only the anaphoric reference of (30b) is responsible for the need to reinterpret (30a). Therefore, precluding reinterpretation potential for a sentence like (30a) merely because it exhibits no semantic mismatches is too rash: from the perspective of larger texts such sentences have reinterpretation potential, too. Thus, it seems advisable not to make the determination of the reinterpretation potential of an expression conditional on a check for semantic mismatches between constituents of the expression.

To sum up, type coercion is modelled by determining the positions where it is possible in the syntax-semantics interface. The interface introduces dominance relations at these positions, which allows for a straightforward semantic construction for instances of type coercion and for their further specification by extralinguistic knowledge sources.

4.3 Representing landing site coercion

This subsection is devoted to landing site coercion. I will analyse this phenomenon and the mechanisms of semantic composition for landing site coercion cases. There are three main problems for such an analysis.

The first issue is the question of *compositionality.* The representation of landing site coercion must be capable of inserting a functor within its argument without violating the maxim that the argument is opaque for the process of semantic construction.

Second, reinterpretation should be triggered *in the same fashion* for both type coercion and landing site coercion, since within semantic construction one cannot decide whether a given functor-argument pair might undergo type coercion, landing site coercion, or no kind of reinterpretation at all. Thus, once again it is the task of the syntax-semantics interface to determine the R-sites for landing site coercion.

The issue of *control* is the third problem for this reinterpretation pattern: the observation is that *both* functor and argument can block landing site coercion. This is illustrated by two variants of the landing site coercion example (4). Neither of them can undergo landing site coercion, although each comprises either the modifier or the modified expression of (4):[13]

(31) Amélie left yesterday

(32) Amélie laughed for two hours

This shows that the syntactic identification of R-sites for landing site coercion must be complemented by assigning the involved expressions lexical representations that avoid overgeneration in that they rule out landing site coercion lexically if this coercion would not be appropriate.

The proposed account of landing site coercion starts with a related phenomenon, the modification of nomina agentis like in *fast eater* ('agentive modification'). This is no instance of landing site coercion (since it belongs wholly to the realm of compositional semantics), but is extremely useful for introducing a pattern of semantic composition which is needed for landing site coercion cases.

When modifying agentive nouns, modifiers like *fast* may pertain to the noun as a whole or only to the verb stem that is part of the noun. The two readings are 'fast person with the habit of eating' and 'person with the habit of eating fast'. If one represents habituality by the genericity operator GEN, the two readings are represented by the λ-terms in (33):[14]

(33) (a) $\lambda x.\mathbf{fast'}(x) \wedge \mathrm{GEN}[e](x\ \mathbf{in}\ e, \mathbf{eat'}(x)(e))$

(b) $\lambda x.\mathrm{GEN}[e](x\ \mathbf{in}\ e, \mathbf{fast'}(e) \wedge \mathbf{eat'}(x)(e))$

To derive these readings, the noun semantics is represented by a twopartite constraint that consists of two fragments, with a dominance relation between them. The embedded fragment is an additional landing site for the semantics of modifiers like *fast*:

(34)
$$\begin{array}{l} \lambda x.\mathrm{GEN}[e](x\ \mathbf{in}\ e,\ \square\ (e)) \\ \qquad\qquad\qquad\quad \vdots \\ \qquad\qquad \lambda e.\mathbf{eat'}\ (x)(e) \end{array}$$

The top fragment is the semantics of the suffix *-er*, the embedded fragment comes from the semantics of the verb stem *eat*. The constraint for the semantics of the whole noun is given as the value of the feature CONT|CON as introduced above. In addition, the embedded fragment is also specified as the value of another feature CONT|EMCON (for 'embedded constraint'). If the constraint for the semantics of a lexeme (e.g., of *to walk*) has no such twopartite structure, the value of this second feature is not specified.

Now the syntax-semantics interface may pertain to the value of CONT|EMCON as well as to the value of CONT|CON. This is what happens in the rule for adjective and adverbial modification. The rules for adjective modification are spelt out in (35)-(36), the rules for adverbial modification of sentence radicals are analogous. The fact that the CONT|EMCON value of a constituent is addressed in a rule is indicated by the subscript E:

$$(35) \quad [_{\bar{\mathtt{N}}}\ \mathtt{AP}\ \mathtt{N}] \qquad \overset{\text{(SSI)}}{\Rightarrow} \qquad \begin{array}{ccc} & [\![\bar{\mathtt{N}}]\!] : \square & \\ [\![\mathtt{AP}]\!](\ \square\ ) & & [\![\mathtt{N}]\!] \\ [\![\mathtt{N_E}]\!] & & \end{array}$$

$$(36) \quad [_{\mathtt{AP}}\ \mathtt{Adj}] \qquad \overset{\text{(SSI)}}{\Rightarrow} \qquad [\![\mathtt{AP}]\!]\text{: } [\![\mathtt{Adj}]\!]$$

Here the modifier constraint is related to its argument constraint, the one of its modified expression, in terms of dominance over the embedded fragment in the argument (the CONT|EMCON value of the modified expression). This strategy allows inserting a functor within its argument. It will be reused in the analysis of landing site coercion, which answers the first of the issues for landing site coercion analyses as noted above.

For *fast eater*, the resulting constraint (37) is a dominance diamond, since the semantics of *fast* is related to the CONT|EMCON value of its argument, and this value is specified in the argument. The functor dominates the embedded fragment of the argument (the verb stem semantics), but the relation between the functor and the part of the argument contributed by the suffix is open. The two constructive solutions of (37) are the λ-terms (33a) and (33b) for the two readings of *fast eater*.

$$(37) \quad \begin{array}{ccc} & \square & \\ \lambda y.\ \mathbf{fast'}(y) \wedge \square\ (y) & & \lambda x.\ \mathrm{GEN}[e](x\ \mathbf{in}\ e, \square\ (e)) \\ & \lambda e.\ \mathbf{eat'}(x)(e) & \end{array}$$

Structures like (37) are the goal of landing site coercion. In contrast to agentive modification, these structures are not the result of semantic composition in landing site coercion cases, but emerge from augmenting the result of semantic composition.

Landing site coercion is similar to agentive modification in that the functor constraint must dominate the value of CONT|EMCON of the argument. Furthermore, either phenomenon is triggered by the syntax-semantics interface, which relates the semantics of modifying and modified expression only in terms of the CONT|EMCON of the latter. The difference between landing site coercion and nomina agentis cases is that in the former the argument fails to specify the value of CONT|EMCON, since its constraint has no embedded fragment that may serve as landing site for the functor. Thus, semantic construction cannot decide the scope of functor and argument, it can only express that both are dominated by a common hole.

As an example for landing site coercion, consider (38), the semantics of (4). Semantic construction (along the lines of (35 and 36)) merely states that functor and argument are dominated by a common hole at the top of the constraint for (4). This is due to the fact that (4) is a case of adverbial modification of a sentence radical, but the value of CONT|EMCON of *Amélie leave-* is not specified in the semantics of the modified expression.

(38)

$$\boxdot$$

$$\lambda e_{+\mathrm{BD}}.\ \mathrm{PH}(\mathbf{2h}, \lambda e'_{-\mathrm{BD}}.\ \Box(e'))(e) \qquad \lambda e'_{+\mathrm{BD}}.\mathbf{leave'}(\mathbf{am\acute{e}lie'})(e')$$

(38) has no constructive solution, since there are conflicting aspectual requirements in functor and argument. (38) can be augmented in terms of a landing

site coercion that bridges the aspectual conflict.[15] One can insert an operator directly above the argument that introduces an additional landing site for the functor. For (38), the relevant operator is (39):

(39)
$$\lambda e_{+\mathrm{BD}}.\Box(e) \wedge \exists e'.\ \boxdot\ (e')$$
$$\lambda e''_{+\mathrm{ST}}.\ \mathrm{RES}(e, e'')$$

(39) applies to bounded predicates like *Amélie leave-*. This is expressed in the sort +BD of the λ-abstracted variable e. In an embedded fragment, (39) explicitly introduces the notion of aftermath: RES relates eventualities with their aftermaths. The property of being an aftermath of some eventuality is stative. (39) maps properties of eventualities P onto the set of eventualities e in the extension of P that have an eventuality e'' associated with them whose description includes the property of being an aftermath of e. The embedded fragment of (39) fills in the missing value of CONT|EMCON for the predicate to which (39) applies.

Inserting (39) directly above the argument in (38) results in another dominance diamond:

(40)
$$\boxdot$$
$$\lambda e^{*}_{+\mathrm{BD}}.\ \mathrm{PH}(\mathbf{2h},\ \lambda e^{**}_{\text{-BD}}.\ \boxdot(e^{**}))(e^{*}) \qquad \lambda e_{+\mathrm{BD}}.\mathbf{leave}'\ (\mathbf{am\acute{e}lie}')(e) \wedge \exists e'.\ \boxdot\ (e')$$
$$\lambda e''_{+\mathrm{ST}}.\ \mathrm{RES}(e, e'')$$

(40) has one constructive solution, in which the functor is inserted between the two parts of the argument. This λ-term (41) models the desired reading of (4) that the aftermath of Amélie's leaving (her absence), but not the leaving itself lasted two hours.

(41) $\lambda e_{+\mathrm{BD}}.\mathbf{leave}'(\mathbf{am\acute{e}lie}')(e) \wedge \exists e'_{+\mathrm{BD}}.\mathrm{PH}(\mathbf{2h}, \lambda e''_{+\mathrm{ST}}.\mathrm{RES}(e, e''))(e')$

This representation of landing site coercion highlights the common ground between landing site coercion and agentive modification in that it represents both phenomena in terms of the same kind of semantic structure.

The structural similarity between (40) and (9) proves the claim that a comprehensive account of reinterpretation phenomena must use a semantic description formalism that is at least as complex as a formalism that can handle scope underspecification.

Landing site coercion is not restricted to aspectual semantics but reappears in the nominal domain, e.g., in *fast road.* Here *fast* (as opposed to e.g. *red*) refers to the value of CONT|EMCON, but the noun does not specify it. The resulting constraint for the semantics of *fast road* is (42):

(42)

$$\boxed{\cdot}$$

$$\lambda x_{\text{-STAT}}.\mathbf{fast}'(x) \wedge \boxed{\cdot}(x) \qquad \lambda y_{+\text{STAT} \,\sqcap\, \text{ARTE}}.\mathbf{road}'(y)$$

(42) has no constructive solution, since there is once again a sortal conflict between functor (the adjective semantics) and argument (the noun semantics). The functor selects for properties of entities with (im-)proper parts that may undergo some change (typically, a change of location), but the argument is a property of entities that have no such parts. This incompatibility is modelled in terms of the sort ±STAT[ionary]. (For eventualities, the distinction coincides with ±ST, i.e., +ST $\sqsubseteq$ +STAT.) The sort ARTE stands for 'artefact'.

The constraint (42) can be augmented along the lines expounded above for (4). The augmentation is given in (43) and introduces the notion of purpose in a fragment of its own. The property of being a purpose for a given entity is not yet restricted with respect to the sorts $\pm$STAT. The operator (43) is restricted to artefacts, as indicated by the sort on the λ-abstracted variable. (43) is provided by world knowledge: artefacts typically are created to serve some purpose.

(43)
$$\lambda x_{\text{ARTE}}.\Box(x) \wedge \exists e.\ \Box\ (e)$$
$$\lambda e.\ \mathbf{purpose'}(e, x)$$

Combining the two constraints (with the embedded fragment of (43) filling in the CONT|EMCON value in the semantics of *fast*) yields the constraint (44):

(44)
$$\Box$$
$$\lambda x_{\text{-STAT}}.\ \mathbf{fast'}(x) \wedge\ \Box(x) \qquad \lambda y_{+\text{STAT}\ \sqcap\ \text{ARTE}}.\mathbf{road'}(y) \wedge\ \exists e.\ \Box\ (e)$$
$$\lambda e.\ \mathbf{purpose'}(e, y)$$

Here the potential structural ambiguity is once again directly resolved by the sortal information: in a constructive solution of (44), the functor can only dominate the dominated fragment of the argument. This at the same time rules out stative purposes. The constructive solution (45) of (44) is the λ-term for the set of roads whose purpose is nonstative and fast:

(45) $\lambda y_{+\text{STAT}\ \sqcap\ \text{ARTE}}.\mathbf{road'}(y) \wedge \exists e_{\text{-ST}}.\mathbf{fast'}(e) \wedge \mathbf{purpose'}(e, y)$

This analysis deliberately leaves open the exact nature of the purpose of the road (in contrast to Pustejovsky, who assumes that this purpose is invariably driving on the road). It takes into account that roads may serve a different purpose in different contexts.

This account of landing site coercion provides a handle on the issue of control. As noted for examples like (31) and (32), the global strategy of introducing landing site coercion in the syntax-semantics interface must be complemented by lexical means of blocking the coercion in order to avoid overgeneration. Both the functor (e.g., the semantics of the modifier in (31)) and the argument (e.g., the semantics of the modified sentence radical in (32)) in a functor-argument structure can block landing site coercion. Either possibility must be modelled in the semantics of these expressions.

First, I discuss how expressions like *yesterday* can block landing site coercion. The idea here is that these expressions introduce an additional dominance relation in their semantics, which states wide scope of their constraint over the CONT|CON value of their modified expression.[16] This is compatible with the dominance relation as introduced in the syntax-semantics interface (i.e., the one of the modifier fragment over the CONT|CON value of the modified expression), but immediately resolves the potential structural ambiguity inherent in this relation.

E.g., (31) has no landing site coercion interpretation, since the fragment for the semantics of the modifier *yesterday* dominates the semantics of its modified expression as a whole. The constraint for (31) looks like this:

(46)
$$\begin{array}{l} \lambda e.\mathbf{yesterday}'(e) \wedge \square\ (e) \\ \qquad\qquad\qquad \vdots \\ \qquad \lambda e'_{+\mathrm{BD}}.\ \mathbf{leave}'\ (\mathbf{am\acute{e}lie}')(e') \end{array}$$

I.e., even an augmentation of the semantics of the modified sentence radical (the bottom fragment in (46)) can never result in an embedding of the modifier semantics (the top fragment in (46)) in the semantics of the modified expression.

The second possibility of blocking landing site coercion lies in the argument, as illustrated by (32). In the proposed approach, this possibility is modelled in the following way: for landing site coercion, the argument must allow augmentation by an operator that provides an embedded fragment (and thereby determines the value of CONT|EMCON). If the argument is incompatible with such an operator, landing site coercion is blocked.

This accounts for the difference between (32) and (4), even though (47), the result of semantic construction for (32), has the same structure as (38), the semantic representation of (4). In this constraint, functor and argument are dominance daughters, since again the modified expression fails to specify the value of CONT|EMCON, while the modifier makes reference to this value:

(47)

$$\square$$

$$\lambda e_{+\mathrm{BD}}.\ \mathrm{PH}(\mathbf{2h}, \lambda e'_{\text{-BD}}.\ \square(e'))(e) \qquad \lambda e'_{\text{-BD}}.\mathbf{laugh'}(\mathbf{am\acute{e}lie'})(e')$$

But the argument in (47) (the semantic contribution of the modified expression) is a constraint for an unbounded predicate (due to the semantics of *laugh*). This unboundedness prevents inserting the operator (39) directly above the argument. Thus, the verb lexically determines that its projections disallow landing site coercion like in (4). This ties in with the intuition that unbounded predicates do not involve the notion of aftermath.[17]

For nominal modifiers, the issue of lexical control of landing site coercion is treated in an analogous way. Those that do not engage in landing site coercion (e.g., *red*) specify this by introducing a dominance relation between their constraint and the CONT|CON value of the modified expression. Hence, *red car* and the like are no reinterpretation candidates.

In addition, nouns can indicate in their semantic contribution whether they participate in specific kinds of landing site coercion, if they are modified by adjectives. E.g., the sorts of their main λ-abstracted argument variables indicate whether the purpose landing site coercion (43) is possible or not: this kind of landing site coercion presupposes that these variables have a sort that is compatible with ARTE. Hence there is no such reinterpretation for *fast comet* and the like, as desired.

The lexical control of landing site coercion in the proposed approach complements the basic strategy of determining R-sites for landing site coercion by syntactic constituent structure. This fine-grainedness is needed because information beyond the syntactic constituent structure is needed to delimit the range of R-sites for landing site coercion: two expressions may share the same immediate constituent structure and yet differ with respect to their landing site coercion potential. In sum, introducing landing site coercion in the syntax-semantics interface and controlling it lexically provides an efficient means of avoiding overgeneration for reinterpretation analyses.

This concludes the presentation of the proposed analysis of reinterpretation phenomena. In this section I have shown how reinterpretation patterns can be accounted for in a flexible semantic analysis that is designed to interact with additional knowledge sources in the full determination of the meaning of a reinterpretation candidate.

5 Conclusion and further work

This paper presented an analysis of reinterpretation phenomena and the challenges they pose for semantic construction. The proposed representation of reinterpretation candidates is couched in a suitable underspecification formalism, which allows for an adequate description of the phenomenon of reinterpretation and integrates the analysis of reinterpretation into a larger current of research on ambiguities in natural language.

The presented work focuses on those parts of the semantics of reinterpretation candidates that are fixed by semantic construction. Hence, it delimits that part of the interpretation of reinterpretation expressions that can be derived in terms of a compositional semantics and shows how the other non-linguistic sources may contribute to the determination of a fully-fledged interpretation for reinterpretation candidates. Eventually, it must be complemented by an account of the other sources and the way they provide information for our understanding of reinterpretation candidates.

It is an advantage of the proposed analysis that it is flexible enough to be integrated into a variety of analyses whose goal it is to cover these other, non-semantic parts as well. It might serve as the non-defeasible part of a default approach to reinterpretation phenomena, or provide one of the knowledge sources that are adduced by an abductive system along the lines of Hobbs et al. (1993) in order to determine the meaning of reinterpretation candidates.

But the presented approach also raises questions on the limits of the proposed strategy. Modelling reinterpretation by deliberately relaxing semantic constraints is just one phenomenon that can be modelled by relaxation. Many more ambiguities (e.g., those that arise from ellipsis reconstruction, see Egg et al. 1998; Egg et al. 2001) are modelled in the very same fashion. The combination of these relaxations might lead to unwanted interactions, in particular, it might give rise to unintended readings. Koller et al. (1999) discuss this problem. They identify positions in constraint structures where further relaxation is possible without running the risk of unwanted ambiguities in terms of a general *safety criterion*. They show that the kind of semantic representations for type coercion expressions that was presented in this paper does not introduce such unwanted ambiguities.

A further line of research that is suggested by the analyses presented in this paper is relating landing site coercion to the phenomenon of *partial modification*. Modification of agentive nouns like in the case of *beautiful dancer* illustrates the phenomenon that sometimes modifiers pertain only to a *part* of the expressions they modify. E.g., lexical NPs like *something* or *everyone* may be modified by adjectives, PPs, or relative clauses, yet semantically, these modifiers pertain only to the restriction of the quantification that is introduced in these modifiers (as in *everyone who knows English* 'the set of properties such that everyone who knows English has them'). Similarly, in the so-called *restitutive* readings of *again*-sentences, *again* pertains only to that part of the meaning of a change-of-state predicate that specifies the *aftermath* of the change of state. E.g., the restitutive reading of *Max left again* is 'Max left, and thereby restituted a preceding absence'. It may well be that landing site coercion turns out to be just one variant of a more global phenomenon of partial modification, though further research is needed at this point.

References

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

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

Copestake, Ann, Dan Flickinger, and Ivan Sag (1997). Minimal Recursion Semantics. An introduction. Available from `ftp://ftp-csli.stanford.edu/linguistics/sag/mrs.ps.gz`.

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

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

Dölling, Johannes (1992). Flexible Interpretation durch Sortenverschiebung. In I. Zimmermann and A. Strigin (eds), *Fügungspotenzen*, 23–62. Berlin: Akademie-Verlag.

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

Dowty, David (1979). *Word meaning and Montague grammar*. Dordrecht: Reidel.

Egg, Markus (1995). The intergressive as a new category of verbal aktionsart. *Journal of Semantics 12*, 311–356.

Egg, Markus (2000). *Flexible semantic construction: the case of reinterpretation.* Habilitation thesis, Universität des Saarlandes.

Egg, Markus, Alexander Koller, and Joachim Niehren (2001). The constraint language over lambda-structures. to appear in *Journal of Logic, Language, and Information* 10; available from `http://www.ps.uni-sb.de/Papers/abstracts/clls2000.html`.

Egg, Markus, Joachim Niehren, Peter Ruhrberg, and Feiyu Xu (1998). Constraints over lambda-structures in semantic underspecification. In *Proceedings of ACL/COLING '98*, Montreal, Canada, 353–359.

Galton, Antony (1984). *The logic of aspect.* Oxford: Clarendon Press.

Herweg, Michael (1991). Temporale Konjunktionen und Aspekt. *Kognitionswissenschaft 2*, 51–90.

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

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

Koller, Alexander, Joachim Niehren, and Kristina Striegnitz (1999). Relaxing underspecified semantic representations for reinterpretation. In *Proceedings of MOL6*, Orlando.

Krifka, Manfred, Francis Pelletier, Greg Carlson, Alice ter Meulen, Gennaro Chierchia, and Godehard Link (1995). Genericity: an introduction. In G. Carlson and F. Pelletier (eds), *The generic book*, 1–124. University of Chicago Press.

Moens, Marc and Mark Steedman (1986). Temporal information and natural language processing. Technical Report Research Paper RP-2, CCS, University of Edinburgh.

Moens, Marc and Mark Steedman (1988). Temporal ontology and temporal reference. *Computational Linguistics 14*, 3–14.

Nunberg, Geoffrey (1993). Indexicality and deixis. *Linguistics & Philosophy 16*, 1–43.

Parsons, Terence (1990). *Events in the semantics of English.* Cambridge: MIT Press.

Piñón, Christopher (1997). Achievements in event semantics. In A. Lawson (ed.), *Proceedings of SALT VII*, Cornell. CLC Publications.

Pinkal, Manfred (1996). Radical underspecification. In P. Dekker and M. Stokhof (eds), *Proceedings of the 10th Amsterdam Colloquium*, Amsterdam, 587–606. ILLC.

Pollard, Carl and Ivan Sag (1994). *Head-driven Phrase Structure Grammar.* CSLI and University of Chicago Press.

Pulman, Steve (1997). Aspectual shift as type coercion. *Transactions of the Philological Society 95*, 279–317.

Pustejovsky, James (1995). *The generative lexicon.* Cambridge: MIT Press.

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

Rooth, Mats and Barbara Partee (1982). Conjunction type ambiguity and wide scope "or". In *Proceedings of the First West Coast Conference on Formal Linguistics*, 353–362. Stanford University.

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

Notes

[1]Throughout the paper, issues of sentence mood and tense are neglected. Sentences minus mood and tense (called 'sentence radicals') are written with an uninflected main verb.

[2]Bounded expressions present a state of affairs as having inherent boundaries (here e.g. the first and the last note of the sonata). Formally, bounded expressions can be defined as those that have in their extension no proper parts of elements of their extension.

[3]Unfortunately, the very term 'reinterpretation' suggests an explanation of the phenomenon in terms of non-monotonic, destructive processes. But since the expression 'reinterpretation' is so well-established, I will continue to use it, although I do not subscribe to this explanation of the phenomenon.

[4]The reading of *to understand* that is at issue here is 'comprehend'. It requires an object NP whose head refers to abstract information structures. Reinterpretation is called for if the object NP is *Faulkner*.

[5]This shows that the meaning of an ambiguous expression cannot be identified with its weakest reading, either, if there is one (for (7), it would be $\forall\exists$). Negating the weakest reading would yield the *strongest* reading for the denial of (7), viz., $\neg\forall\exists$.

[6]In the following, I will often abbreviate 'fragment 1 comprises a hole that dominates fragment 2' as 'fragment 1 dominates fragment 2'.

[7]CLLS proper does not use variable names to model the binding of variables, in contrast to the adapted notation. Instead, binding is explicated by a function λ. Roughly, this function relates variables to their binders. This point is of no importance here, hence neglected in the adapted notation.

[8]These types of constraints are characteristic for type coercion and landing site coercion, respectively. They will be discussed extensively in sections 4.2 and 4.3.

[9]In particular, the group of bounded predicates must be subdivided further, but this is of no importance for the present discussion. See Egg (1995) and Piñón (1997) for (conflicting) opinions on this issue.

[10]Aspectual classification is based on so-called *compatibility tests.* Compatibility (without reinterpretation) with durative adverbials distinguishes unbounded from bounded predicates, with time frame adverbials like *in three hours*, the pattern is the reverse. Stative predicates disallow the progressive and modification by adverbials like *quickly.* See Dowty (1979) for an extensive list of such tests.

[11]Note that the complexity of this definition is not a consequence of the chosen approach to reinterpretation. It emerges for *any* account of the durative adverbial semantics, if this account is part of an analysis that introduces some sortal distinction between stative/nonstative (e.g., Parsons 1990) or bounded/unbounded predicates (e.g., Galton 1984).

[12]The intuitions that underlie this analysis were considerably influenced by

the work of Dölling (1992, 1995).

[13]Otherwise, they would have to have readings which they have not: (31) has no reading 'Amélie left (sometime in the past) and was away yesterday'; similarly, (32) cannot mean 'Amélie laughed and the aftermath or result of this laughing lasted for two hours'.

[14]The same ambiguity would emerge if *fast* were analysed as a non-intersective modifier. The expression 'x **in** e' means that x participates in e.

[15]Type coercion might bridge the gap, too, e.g., integrating an iterative operator would yield the reading that Amélie left repeatedly during the two hours.

[16]Since modifiers exert selectional restrictions on their modified expression, they must specify them somewhere in their linguistic representation (e.g., in the MOD feature of HPSG, Pollard and Sag 1994). This is also the position where it is possible for the modifier to refer to the CONT|CON of the modified expression.

[17]Rather than by landing site coercion, (47) is resolved by letting the functor directly dominate the argument. This yields a constraint whose constructive solution is a λ-term for the set of eventualities that last two hours and are spatiotemporally coextensive to an eventuality of Amélie laughing.