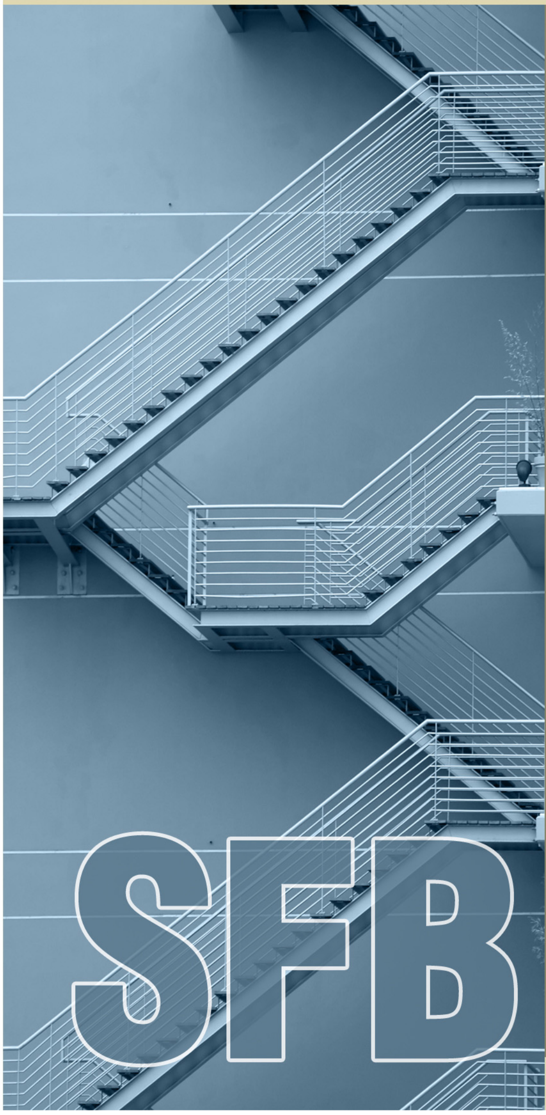
Universität Stuttgart

Antje Roßdeutscher (ed.):

Disambiguation and Reambiguation



SinSpeC

Working Papers of the SFB 732
“Incremental Specification in Context”

Universität Stuttgart

SinSpeC 06 (2009) ISSN 1867-3082

SinSpeC issues do not appear on a strict schedule.

Volume **06 (December 2009)**

Volume Editor: Antje Roßdeutscher
Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
Azenbergstraße 12
70174 Stuttgart
Germany
antje@ims.uni-stuttgart.de

Series Editors: Artemis Alexiadou
Universität Stuttgart
Institut für Lingusitik/Anglistik
Heilbronner Strasse 7
D-70174 Stuttgart

Hinrich Schütze
Universität Stuttgart
Institut für maschinelle Sprachverarbeitung
Azenbergstrasse 12
D-70174 Stuttgart

Published by **O**nline **P**ublikationsverbund der **U**niversität **S**tuttgart (OPUS)

Published 2009

ISSN **1867-3082**

About *SinSpeC*

SinSpeC are the Working Papers of the Sonderforschungsbereich (SFB) 732 "Incremental Specification in Context". The SFB 732 is a collaboratory research center at the University of Stuttgart and has been funded by the German Research Foundation (DFG) since July 1, 2006.

The SFB 732 brings together scientists from the areas of linguistics, computational linguistics and signal processing at the University of Stuttgart. Their common scientific goals are to achieve a better understanding of the mechanisms that lead to ambiguity control/disambiguation as well as the enrichment of missing/incomplete information and to develop methods that are able to fully describe these mechanisms.

For further information about the SFB please visit:

http://www.uni-stuttgart.de/linguistik/sfb732/

SinSpeC aims at publishing ongoing work within the SFB in a fast and uncomplicated way in order to make the results of our work here known to the scientific community and strengthen our international relationships. It publishes papers by the staff of the SFB as well as papers by visiting scholars or invited scholars.

SinSpeC is available online on the above website.

Contact Information:

Director of the SFB 732:

Prof. Dr. Artemis Alexiadou
artemis@ifla.uni-stuttgart.de

Coordinator of the SFB 732:

Dr. Sabine Mohr
sabine@ifla.uni-stuttgart.de

SFB 732
Universität Stuttgart
Keplerstraße 17
D-70174 Stuttgart

Phone: 0711/685-83115
Fax: 0711/685-83120

Ambiguation and reambiguation: Introduction to the volume

Antje Roßdeutscher (antje.rossdeutscher@ims.uni-stuttgart.de)
Institute for Natural Langauge Processing
University of Stuttgart

The papers in this volume developed as part of the two projects *The Role of Lexical Information in the Context of Word-formation, Sentence and Discourse* and the project *Representation of Ambiguities and their Resolution in Context*. In the former, a theory of -*ung*-nominalisation in German has been developed. German -*ung*-formation is productive but it is restricted. This restriction can be made precise in an account of word-formation where the syntax and semantics of verbs emerges from their sub-lexical units. Whether a verb has a corresponding -*ung*-noun or not, depends on the semantics of those units. If the kernel predicate of a verb describes manner of action, like *arbeiten* (*to work*) there is no *ung*--noun **Arbeitung*. If the kernel predicate describes an entity like German *Sperre* ('barrier') in *sperren* ('to bar, to block sth.') or *absperren* ('to cordon off'), the verb has an -*ung*-noun. Moreover the sub-lexical units allow us to predict which readings – event, result state or entity interpretations are productively derived – can be expected for an -*ung*-nominal. The details of how such a theory of word-formation is motivated and can be defended can be found in Roßdeutscher and Kamp. (2010) and in Roßdeutscher (2010).

The two papers presented in this volume focus on the second part of the joint enterprise of the two projects, namely on disambiguation of -*ung*-nouns in context. Hamm and Kamp study a proto-typical example, *die Absperrung der Botschaft* 'the cordoning-off of the embassy', which is three-way ambiguous. This DP can denote a material object (the fence used for cordoning-off), an event (the process of cordoning-off) or a result state (the embassy being cordoned off). Formally, this three-way ambiguity is represented by an underspecified DRS, (cf. Reyle, 1993, and subsequent work). The paper contributes a partial answer to the general question which contextual factors are responsible for the (partial) disambiguation of this DP in discourse. The disambiguation process is described on the level of DRT.

However, apart from problems concerning disambiguation, there are other factors affecting the possible denotations of DPs like *die Absprerrung der Botschaft*. These factors concern inferences which are triggered by certain verbs and blocked by others. In the verbal context of *behindern* ('to hamper'), for instance, we may infer (non–monotonically) that the result state – the embassy being cordoned-off – was successfully established albeit this aim was not achieved in a smooth way. Like *behindern*, the verb *verhindern* 'prevent' selects the event reading of *die Absperrung der Botschaft*, but in the latter case the inference that the result state was established, is clearly blocked. The matter is more complicated in the verbal context of *unterbrechen* ('to interrupt'): Intuitively, the time profile of *unterbrechen* is as follows: the process in question (in our case the cordoning-off) started at some time t and continued for some indefinite time span. At some time later than t the process was stopped for some indefinite time span and resumed after-

Antje Roßdeutscher (ed.): Reambiguation and Disambiguation (SinSpec06)
Working Papers of the SFB 732 *Incremental Specification in Context*,
University of Stuttgart, 2009: i – iii

wards. As with *behindern*, we get as a non-monotonic inference that the result state was eventually established although the process was interrupted for some time. This is the by now familiar kind of non-monotonicity. However, (1) shows that the time profile of *unterbrechen* is defeasible itself. It contains a necessary part roughly corresponding to *abbrechen* 'cancel', and a part which may be resumed but which may also be cancelled in certain contexts.

(1) *Die Absperrung der Botschaft wurde unterbrochen und nie wieder aufgenommen.*
'The cordoning-off of the embassy was interrupted and never resumed again.'

In order to capture both the disambiguating and the inference triggering effects of verbal contexts, Hamm and Kamp argue for a combination of underspecified DRT and an event calculus considered as a logic program. This combined formal system was developed in Hamm et al. (2006).

Building on the results in the first paper, the second paper by Hamm and Solstad focuses on problems that arise in anaphora resolution of pronouns with ambiguous nouns like *die Absperrung der Botschaft* as antecedent. What happens if the selection restriction of the verb in the antecedent sentence and that of the consequent sentence are incompatible? This situation is exemplified in (2):

(2) *Die Absperrung des Botschaft wurde vorgestern von Demonstranten behindert. Wegen anhaltender Unruhen wird* ***sie*** *auch heute aufrecht erhalten.*
'The cordoning-off of the embassy was hampered by protesters the day before yesterday. Due to continuing unrest, it [the state of being cordoned off] is sustained today as well.'

Behindern 'to hamper' filters out both the entity-reading and the result state reading of *Absperrung*, but the verb *aufrecht erhalten* 'to sustain' requires the result state as its argument. Thus, in order for the anaphoric pronoun *sie* to be resolved successfully, the first sentence should provide a result state which, however, is not available, if the result state reading has been erased.

Hamm and Solstad show that the required result state can be reconstructed – even under the assumption that *behindern* erases the result state reading of the first sentence in (2). This is achieved in a process of *reambiguation*. Reambiguation involves a non-monotonic inference process. The question arise what triggers this process and what its restrictions are. Hamm and Solstad provide formally precise answers to these questions. Again a combination of UDRT and the event calculus provide the framework where these puzzles can be solved.

References

Hamm, F., Kamp, H. and van Lambalgen, M. (2006): There is no opposition between formal and cognitive semantics, *Theoretical Linguistics* **32**(1), 1–40.

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

Roßdeutscher, A. (2010):, German *-ung*-formation. An explanation of formation and interpretation in a root-based account, *in* S. Olsen (ed.), New Impulses in Word-Formation, Special issue 117 of *Linguistische Berichte*, Hamburg: Helmut Buske Verlag.

Roßdeutscher, A. and Kamp., H.: 2010, Syntactic and Semantic Constraints in the Formation and Interpretation of *ung*-Nouns, *in* A. Alexiadou and M. Rathert (eds), *Nominalisations across Languages and Frameworks*, Mouton de Gruyter, Berlin.

Ontology and Inference

The Case of German *ung*–Nominals

Fritz Hamm (fritz.hamm@ims.uni-stuttgart.de)
Institute for Natural Langauge Processing
University of Stuttgart

Hans Kamp (hans.kamp@ims.uni-stuttgart.de)
Institute for Natural Language Processing
University of Stuttgart

1 Introduction

Our linguistic competence has two parts, that of being able to express what we want to say and that of being able to understand what we are told. The point of the second part (and indirectly therefore also of the first) is that understanding yields information which can be put to further use. Almost always this further use takes the form of drawing inferences that involve the new information as premise.

A good theory of language interpretation should tell a story about how this works. At a minimum it should show not only how we extract interpretations from linguistic input, but also that the interpretations we extract have the structural properties that allow them to play their part as premises in human inferencing processes.

What such a theory should look like in detail will depend on what we assume about the nature and forms of human reasoning. For example, someone who thinks of reasoning along the lines of deduction systems for first or higher order predicate logic would want an account of interpretation that transforms linguistic input into interpretations whose form allows them to serve as premises in deduction systems of this sort. Such interpretations should be like formulas of predicate logic (or, failing that they should be readily convertible into such formulas). But of course, we need not think of the inference modules of human cognition as deduction engines for predicate logic. In fact, we definitely shouldn't think of them as *just* that. Most of the inferences people draw in ordinary life are defeasible inferences: conclusions depending on implicit assumptions, statements on which the premises only confer a high probability, abductions, educated guesses etc. There is considerable latitude here for alternative conceptions. In particular, there seems room for conceptions on which premises and conclusions come in forms that are very different from predicate logical formulas. Someone who embraces an alternative conception of this kind will also put different demands on the theory of interpretation. He will want a theory of interpretation to show how natural language input can be converted into representations that are suitable as premises and conclusions in inferences that fit his conception of inference.

In this paper we look at some implications of one such conception of human inferencing. This conception is based on the method of Constraint Logic Programming (CLP); a detailed presentation of the approach can be found in van Lambalgen and Hamm (2005). In this approach the premises of inferences are given as programs and constraints ex-

Antje Roßdeutscher (ed.): Reambiguation and Disambiguation (SinSpec06)
Working Papers of the SFB 732 *Incremental Specification in Context*,
University of Stuttgart, 2009: 1 – 67

pressed in a certain Constraint Logic Programming Formalism. The inferences that can be drawn from such combinations of programs and constraints within the Constraint Logic Programming framework are statements that are verified by models of the completions of those programs. This gives an inference relation that is more powerful than classical deductive validity, and it is non-monotonic[1]. The notion is particularly well adapted to deal with inferences that people are prone to draw from narrative texts and other text types in which chains of causally connected events, states and processes play a prominent role: interpreters assume that events that are neither mentioned explicitly nor unequivocally entailed by what is explicitly mentioned did not take place at all. Many of the inferences we draw reflect this understanding in that they would not be true had such events actually occurred. The CLP approach captures this aspect of human inferencing because the completions of the programs that represent narrative texts exclude the problematic events; in the models of those completions no such events are present.

It is not easy to develop a theory of natural language interpretation that turns linguistic input directly into suitable representations at CLP level. We have therefore been experimenting with an indirect method, in which natural language inputs are first converted into Discourse Representation Structures; in a second step these DRSs are then converted into CLP level representations (Hamm et al., 2006). This two-stage procedure has several advantages. A first practical advantage is the considerable coverage of natural language constructions that has been accumulated within DRT in the course of the past few decades (van der Sandt, 1992; Kamp and Roßdeutscher, 1994; Geurts, 1999; Roßdeutscher, 2000; Kamp, 2003; Kamp and Roßdeutscher, 2010; Genabith et al., 2010; Kamp and Reyle, 2010). For a wide variety of sentences DRSs can now be built systematically (and even generated automatically from NL text by machine, see e.g. Kamp and Reyle (1993), Blackburn and Bos (2009). Furthermore, converting DRSs into CLPs is a more streamlined and therefore simpler procedure than going from natural language to CLPs directly. But there is also a second advantage. As things stand, DRT has little to say on the topic of defeasible inference. But all DRS formalisms come equipped with model theories and thus with model-theoretic notions of valid inference. Moreover, first order fragments of DRS-languages (as in Kamp and Reyle, 1993, Chs. 1,2) can be straightforwardly translated into first order predicate logic, and so can take advantage of the theorem provers for first order logic that are currently on the market[2]. These classical inference relations and their implementations fall short of giving us a credible account of human reasoning. But even so they cover a substantial part of the inferences that human beings are able to draw, and often do draw.

In a two stage interpretation system of the sort we have just described inferencing is possible at two distinct levels, that of DRT and that of the CLP. This creates an opportunity for two-pronged accounts of reasoning processes, in which some inferences happen at the DRT- and some at the CLP-level.

Not only do we draw inferences from our interpretations of the linguistic input we receive. We also have to rely on inferencing in order to get to those interpretations. Many of those inferences are needed for resolving ambiguities, and a large majority of those

[1]There are some CLPs P_1 that can be extended to CLPs P_2 with the property that some of the models of its completion are not models of the completion of P_1.

[2]Kamp and Reyle (1996) presents a deduction system for first order DRT that is fitted to the distinctive structural properties of DRSs.

ambiguities are lexical ambiguities. The vocabularies of natural languages are rife with ambiguity, both in that they contain large numbers of ambiguous words, and in that many of those have more than two readings, often many more. One of the striking aspects of the efficiency of natural language is that ambiguities are almost always resolved in context. Moreover, for the resolution of lexical ambiguities usually not much contextual information isl required; mostly the purely linguistic local context – the sentence in which the ambiguous word occurs – is all that is needed.

When a word is multiply ambiguous, its full disambiguation may take several steps, in which the readings that the given context excludes are successively eliminated. In such cases each step may involve its own inferences and moreover, these inferences may belong to different levels. The cases of disambiguation that we will consider here only involve inferencing at the DRT level. This has to do with the special features of sortal ambiguity resolution and does not generalise to ambiguity resolution in general. Moreover, even in cases where disambiguation takes place at the level of DRS construction, processing may requre further inferences that belong to the CLP level. In fact, part of the paper will be devoted to a detailed treatment of an example for which inferences at the DRS level (needed for the resolution of sortal ambiguities) are followed by a further inference at the level of CLP.

The structure of the paper is as follows. (Don't read this if you don't want to, you will find out in any case.)

Section 2 gives a brief account of our interest in *-ung*-nominalisation and in the ambiguities of *-ung* nouns to which it gives rise. We then present the sentences that will preoccupy us throughout the paper. Each of these sentences involves a transitive verb and all of them have the same theme DP – *die Absperrung der Botschaft* (roughly: *the fencing off of the embassy*) – in which our paradigm noun, the three ways ambiguous *Absperrung*. is the lexical head.

Section 3 deals with the DRT part of our story. 3.1 presents lexical entries for *Absperrung* and for the verb *Absperren* from which it is derived, as well as a semantic representation for the DP *die Absperrung der Botschaft*. 3.2 presents lexical entries for most of the verbs of the sentences of Section 2 and the DRS construction for some of these sentences. These constructions show how a noun like *Absperrung* can get wholly or partly disambiguated in the course of sentence interpretation. (The length of this section is due to the unusual semantic properties of the chosen verbs. These are outside the standard repertory that has been studied in extant work in formal semantics and require careful discussion.)

With Section 4 we reach the second part of the paper. Section 4.1 reviews our reasons for wanting a combined DRT-CLP architecture. Section 4.2 given a succinct introduction to Logic Programming, emphasising the features that are important for the use that is made of it here. Sections 4.3-4.5 the system of Constraint Logic Programming that is used here - a CLP-based version of the Event Calculus, as developed in van Lambalgen and Hamm (2005). Of special importance for our purpose are the *integrity constraints* discussed in 4.4 - it is they, and not the Logic *programs* themselves, that correspond to the DRSs of Section 3 - and the method of *reification* presented in 4.5.

In Section 5 the two halves of our proposal - the DRT half and CLP half - are at last brought together. The general problem we are facing here is to define the transition from DRSs to representations in the format of our version of the Event Calculus. As indicated already in Section 4, DRSs of episodic sentences and texts do not translate into CL-programs but into integrity constraints. Constraint Logic programs also play a part, but they arise as translations of lexical entries (most notably, as translations of the lexical entries of verbs). The logic of semantic representations at this level is defined by the CLP-behviour of integrity constraints that translate sentence and text DRSs against the backgropund provided by the CL programs that serve as lexical entries for the verbs (and other words) occurring in the given sentence or text.

Section 6 (the Conclusion) concludes.

2 German *ung*–Nominals

The word that will be central to this paper is the deverbal German noun *Absperrung* (from the verb *absperren*, Engl. *fence off*, *block off*; the meaning of the noun will be explained below.) The choice of *Absperrung* reflects our general interest in the processes of "-*ung*-formation", a German word formation process in which nouns are derived from verbs by attaching the suffix -*ung*. -*ung*-formation is a challenge to linguistic theory because

- there are many verbs that allow for the formation of -*ung*-nouns and many others that do not
- the -*ung* nouns that exist vary in that some are ambiguous while others aren't, and the ambiguous ones vary in how they are ambiguous.

To a large extent these phenomena appear to be systematic und thus force two general questions upon us:

1. What is responsible for the possibility of -*ung*-formation? and
2. What determines the range of possible readings of -*ung*-nouns that can be formed?

Finding answers to these questions is one of the principal tasks of the research project that has also included the study reported here.

The general question of the present study is a different one. Our central concern is the resolution of lexical ambiguity in context. But the connection with questions (1) and (2) above is obvious enough. We too are concerned with aspects of -*ung*-nominalisation. But rather than asking where the ambiguities of -*ung*-nouns come from we focus on when and how those ambiguities disappear again when the nouns are used as parts of sentences.

The ambiguities of -*ung*-nouns we will be dealing with are ambiguities of a special kind. They are what we call *sortal* ambiguities. To see what that means consider the word *Absperrung*. *Absperrung* is ambiguous between describing

1. events of fencing off something (a building, a plot of land, a street);
2. the states which result from such actions, i.e. the state of the building, plot or whatever having become inaccessible on account of such an action;
3. the fences or barricades that are erected in the course of *Absperrung*-actions.

Thus, each of the three readings involves a different ontological category – the category 'event', the category 'state' and the category 'physical object' (or 'entity'). Something like this is true more generally: The sortal ambiguities of -*ung*-nouns come in different 'patterns', where each pattern is given by a set of ontological categories (or 'sorts', as we will also call them). The ambiguity pattern of *Absperrung* is given by the set {event, state, entity}. Other ambiguous -*ung*-nouns have patterns that are two element subsets of this set. (The -*ung*-nouns whose 'ambiguity patterns' are given by singleton sets are the ones that are not sortally ambiguous.)

Sortally ambiguous nouns allow for a form of disambiguation that is specific to this type of ambiguity. When nouns occur in sentences, they do so mostly as the lexical heads of noun phrases that occupy an argument position of some lexical predicate – a verb, a preposition, an adjective or another noun. In such cases the predicate is referred to as the *container predicate* (of the noun phrase and, by proxy, also of the head noun of that noun phrase). As a rule, argument positions of lexical predicates impose *selectional restrictions* on their arguments – conditions that delimit the sorts of entities that can fill them. One way in which the selection restrictions for an argument position manifest themselves is by excluding from it any argument phrase whose lexical head is incompatible with them. And, by the same token, when the head of the argument phrase is ambiguous, it may be that the selectional restrictions for the argument position are compatible with some of its readings but not with others. In such cases the selectional restrictions will have a disambiguating effect; the head will become either fully disambiguated or else its ambiguity will at least be reduced.

Prominent among the container predicates of noun phrases are verbs, and it is on these that we will concentrate. (1) lists a few sentences in which the noun phrase die *Absperrung* occurs in the theme position of different transitive verbs[3].

(1) a. Die Absperrung der Botschaft wurde abgebaut.
The blocking-off of the embassy was removed.
b. Die Absperrung der Botschaft wurde behindert.
The blocking-off of the embassy was obstructed.
c. Die Absperrung der Botschaft wurde verhindert.
The blocking-off of the embassy was prevented.
d. Die Absperrung der Botschaft wurde unterbrochen.
The blocking-off of the embassy was interrupted.
e. Die Absperrung der Botschaft wurde aufgehoben.
The blocking-off of the embassy was lifted.
f. Die Absperrung der Botschaft wurde ignoriert.
The blocking-off of the embassy was ignored.

The sentences in (1) differ from each other in that the theme positions of their verbs come with different selection restrictions. Each set of selection restrictions has a different effect on the disambiguation of the head noun *Absperrung*. In what follows we will

[3]For reasons connected with the second part of the paper our sample sentences are passives, in which the theme phrase plays the part of grammatical subject, and not that of direct object, as it would in the corresponding active sentences.

be looking carefully at these effects and at the ways in which they interact with other disambiguating factors.

The constraints that the verbs in (1) impose on the interpretation of *Absperrung* can be accounted for at the level of DRS-construction and that is where we will account for these effects here. But in many instances the representations that result from DRS construction-cum-disambiguation invite further processing at the level of CLP. Among the sentences in (1) there is one, viz. (1)d with its verb *unterbrechen* ('interrupt'), which illustrates this possibility in an interesting way. In the second part of the paper we will use this sentence as an illustration of how DRS construction can provide input to further inferential processing at the level of CLP.

We will discuss the relevant features of the different sentences in (1) at our leasure in Section 3. But as a foretaste of what is to come, we conclude this section with a few informal remarks about the semantics of the verb *unterbrechen*. These should give a first impression of what it is that makes (1)d particularly relevant to the two level inference architecture we are proposing.

The theme arguments of *unterbrechen* are activities. *unterbrechen* says of such a theme argument that it is stopped, but with a presumption that it will be resumed at a later time. That the resumption part is only a presumption is shown by the coherence of examples like (2).

(2) Die Absperrung wurde unterbrochen und dann nie wieder aufgenommen.
(Literally: The blocking was interrupted and then never taken up again.)
They stopped erecting the barricades and then never resumed the work.

But while the resumption can be cancelled, it is present as a presumption no less: only explicit cancellation, as in (2), will make it go away.

That activities which are described as 'unterbrochen' (i.e. as 'interrupted') are subsequently resumed is, then, a default aspect of the meaning of *unterbrechen*. But in addition to this we can observe in some cases a further, 'secondary' default effect. This arises when the theme argument phrase of *unterbrechen* describes a goal-directed action. An example is the phrase *die Absperrung der Botschaft* when understood in its event reading: the events the phrase describes are actions that result in the embassy being inaccessible. With such theme phrases *unterbrechen* not only carries an implication that the activity which is meant to lead to the embassy's being inaccessible is resumed after the interruption, but also that the activity is then eventually completed (so that the embassy *is* fenced off and inaccessible).[4]

[4]It has been observed that this second implication – that of the embassy's fencing off being completed after resumption – is rather weak. (Our informants vary on how weak.) The implication appears to be somewhat stronger, however, with a sentence like (ia), in which there is an explicit mention of how long the interruption is meant to be for. As shown by (ib), the implication that the activity will be completed can be cancelled just as well as in the case of (2).

(i) a. Die Absperrung wurde für einige Stunden unterbrochen.
They interrupted the erecting of the barricades for several hours.
b. Die Absperrung wurde für einige Stunden unterbrochen, aber dann nie wieder aufgenommen.
They interrupted the erecting of the barricades for several hours, but then never resumed the work again.

This second implication – that the fencing off will be completed after resumption – is of a different sort. Unlike the implication it presupposes – that the activity will be resumed – it is not part of the lexical meaning of *unterbrechen*, but an instance of the general discourse principle referred to in Section 1: events that are neither mentioned in the discourse nor follow from what is explicitly mentioned are not part of the episode described. The second implication thus falls within the province of the Event Calculus.

This means that when all is told, (1)d involves, first, inferences at the level of DRT – those needed for the disambiguation of *Absperrung* as well as the (default) inference that the activity will be resumed – and a subsequent inference at our second level, one that would not have been possible without the ones that precede it, but which could only be drawn at the level where it is drawn. Here we see the synergetic effects of inferencing by different modules that operate according to radically different principles.

3 DRS-Constructions for *ung*–Nominals and Container Verbs

The first ingredients we need to account for the disambiguation effects in (1) are lexical entries for the noun *Absperrung* and its container verbs. Since the account reconstructs the disambiguations at the level of DRSs, the entries will have to be given in a form suitable for DRS-construction. We will be using the format first introduced in (Kamp and Roßdeutscher, 1994).

3.1 The Noun *Absperrung*

As noted in Section 2, *Absperrung* is three ways ambiguous. Its three readings emerge naturally from the internal syntactic and semantic structure that *Absperrung* shares with the verb *absperren*. (3) gives the lexical entry we assume for *absperren*.

(3)

(E0) **absperren**

absperren	verb	nom	acc	(*mit*-PP)
(i)	e	x	y	[z]
SEL.RESTR.	event(e)	agent(x)	[+z] region(y) $\vee$ [-z] utility(y)	barrier(z)

Semantic Representation

$s_0 \quad s_1 \quad [s_2] \quad [s_3]$
$s_0 \supset\!\subset e \quad e \supset\!\subset s_1 \quad [e \supset\!\subset s_2] \quad [e \supset\!\subset s_3]$ s_0: accessible(y) s_1: $\neg$ accessible(y)   CAUSE(e,s_1)   Agent(e) = x [s_2: present(z)] [s_3: sperr(z,y)] [CAUSE(e,s_2)] [CAUSE(e,s_3)]

But in cases where it isn't, the implcation that the resumption will be followed by completion appears to be reasonably robust.

Details of how such entries should be read and how they are used in the construction of sentence representations will be explained in Section 3.2 in relation to the verbs that occur in the sentences listed under (1). (For those familiar with the notation of Kamp and Roßdeutscher (1994): the use of the square brackets around 'z', s_2, s_3 and the conditions involving these discourse referents, which are novel here will be explained following the entry for *Absperrung* below.)

What matters most at this point is that the argument structure of *absperren*, presented in the top tier of (3), contains the three elements that can become the referential argument of *Absperrung*: e, s and z.[5]

The lexical entry for *Absperrung* that we will be using derives from (3) in an obvious fashion. We represent the referential argument of *Absperrung* as α and represent the ambiguity of *Absperrung* via the disjunction $\overset{!}{\vee}$.

$$\alpha = \mathrm{e} \overset{!}{\vee} \alpha = \mathrm{s}_1 \overset{!}{\vee} \alpha = \mathrm{z}.$$

The symbol $\overset{!}{\vee}$ is used to list alternative interpretations. It is not to be confused with the disjunction operator $\vee$. When A and B are formulas (or DRSs), then $A \vee B$ is a formula (or DRS condition) which is true if at least one of the disjuncts is true. In contrast, $A \overset{!}{\vee} B$ is not a formula (or DRS condition) in the usual sense. It is an instruction to choose between A and B: the interpretation under construction will either have to be A (or contain A in the position occupied by $A \overset{!}{\vee} B$) or it will have to be or contain B. In other words, before an interpretation can be considered complete, every $\overset{!}{\vee}$-disjunction should be eliminated in favour of one of its disjuncts. Representations containing $\overset{!}{\vee}$-disjunctions are thus *underspecified* interpretations, which can be turned into proper (fully specified) representations only by eliminating all occurrences of $\overset{!}{\vee}$-disjunctions occurring in them (Reyle et al., 2008).

Apart from its $\overset{!}{\vee}$-condition the semantic representation of the lexical entry for *Absperrung* is like that for *absperren*.

(4)

(E1) **Absperrung**

Absperrung (i)	noun α	(*durch*-PP) x	(DP-gen) y	(*mit*-PP) [z]
SEL.RESTR.	event(e)	agent(x)	[+z] region(y) $\vee$ [-z] utility(y)	barrier(z)

Semantic Representation

[5] For an account of why only these three discourse referents from (3) are possible as referential arguments of *Absperrung* see Kamp and Roßdeutscher (2010).

$s_0 \quad s_1 \quad [s_2] \quad [s_3]$
$s_0 \supset\subset e \quad e \supset\subset s_1 \quad [e \supset\subset s_2] \quad [e \supset\subset s_3]$ s_0: accessible(y) s_1: $\neg$ accessible(y)   CAUSE(e,s_1)   Agent(e) = x [s_2: present(z)] [s_3: sperr(z,y)] [CAUSE(e,s_2)] [CAUSE(e,s_3)] $\alpha = e \overset{!}{\vee} \alpha = s_1 \overset{!}{\vee} \alpha = z$

It is time for the promised explanation of the square brackets around z etc., which (4) shares with in (3). The square brackets around ‘z’ in (3) indicate that there is a participant z in the event structure described by *absperren* only under certain conditions, which are reflected in the sort of the theme argument y: When y is a ‘region’ (a territory, a road, a plot of land, a building, a part of a building etc.), then ‘absperren’ involves a barrier (a fence, wall, barricade or the like), and, with it, the state s_2 of the barrier being present and the state s_3 of its making y inaccessible; if the theme argument is a utility (water, gas, electricity), then no barrier is involved. In this case none of z, s_2, s_3 and the conditions in which they appear as arguments are relevant, and the entry is then to be treated as not including them. This difference is made explicit in (3) by marking the first sortal option for y as [+z] and the second option as [-z] and by placing those discourse referents and conditions that are only relevant for the first option between brackets. This dependence of the participation of z and what comes in its wake on the sort of y carries over to the semantics of (4): When the theme of *Absperrung* (optionally realised by a genitive DP) is a utility, then *Absperrung* allows for only two readings, viz. event and result state; when y is a region, then the entity reading is possible a well.

We get a first glimpse of the role that the $\overset{!}{\vee}$-condition of (4) plays in the constructions for expressions involving *Absperrung* by looking at the construction of a semantic representation for the DP *die Absperrung der Botschaft*. Like its head noun when taken by itself, this phrase is three ways ambiguous. This is a fact that merits explicit notice and a brief discussion. The entries for all three readings specify an argument y that is optionally realised as a genitive DP. So the parser can use each of the three readings to assign a structure to the complex DP in which the embedded DP *ier Botschaft* fills the y argument slot of *Absperrung*. Moreover, in each of the three cases this structure is coherent since the selectional restrictions of the theme argument of *Absperrung*, which are the same for all three readings, are compatible with the head noun *Botschaft* of the embedded DP.[6]

To give a detailed account of how the ambiguous semantic representation of the NP *Absperrung der Botschaft* results from the semantic representations of its parts we would need a lexical entry for the noun *Botschaft*. This entry would have to be such that it would enable us to detrermine its compatibility with the selectional restrictions on the theme argument of *Absperrung*. In fact, we encounter here an instance of the same phenomenon that will preoccupy us throughout Seection 3: The ambiguity reduction of the lexical head of an argument phrase through the selectional constraints imposed by the container predicate.

[6]Among the possible interpretations of the noun *Botschaft* there is one according to which it describes the premises of the embassy in question or, alternatively, the building in which the embassy is housed.

The ambiguities of *Botschaft* are of two sorts. First there is the distinction between *Botschaft* in the sense of embassy and *Botschaft* in the sense of message. There is a fairly obvious historical connection between these two meanings, but it is unlikely that this connection plays any significant part in our current understanding of the word. Secondly there is the polysemy that *Botschaft*, when taken in its embassy sense, shares with other nouns that denote institutions We can use *Botschaft* to refer to the institution as such (a branch of the ministry of foreign affairs), to the building where the institution is situated, to its personnel and so forth.[7]

As regards the combination of *die Botschaft* with *Absperrung* the situation before us is as follows: *Botschaft* can be used to denote a considerable range of different sorts, but most of these are incompatible with the sortal restrictions that *Absperrung* imposes. The one or two possibilities that remain belong to the family of polysemous meanings of *Botschaft* in its sense of embassy: in *Absperrung der Botschaft Botschaft* can either stand for the building or building part in which the embassy is housed or for the entire embassy compound including the grounds. (Whether these should be counted as two possibilities or one may be a matter for debate.)

It should be noted that for its part *Botschaft* also has a kind of disambiguating effect on *Absperrung*. Since *Botschaft* cannot be used to refer to a utilitiy, it can be inferred that *Absperrung der Botschaft* must involve the participant z. So in the representation of *Absperrung der Botschaft* the square brackets that occur in the ledical entry of *Absperrung* can be removed.

We do not present a lexical entry for *Botschaft* here and thus cannot present a detailed account of the compositional semantics of *(die) Absperrung der Botschaft*. Instead, what follows is a bit of a stopgap. We adopt a very much simplified representation for the DP *die Botschaft* (see (5) below) and assume that when this representation is combined with the entry for *Absperrung* the resulting representation is as in (6).

Before we present this representation one further point should be mentioned. It concerns the two definite articles that occur in this DP. There is widespread agreement that singular definite descriptions come with an existence presuppositon and a uniqueness presupposition. The precise form and status of these presuppositions has been for many decades the topic of a debate that remains unresolved to this day. For our own views on this matter see Genabith et al. (2010) and Kamp (2008).

To go into this issue here would be counterproductive. So we will make do with the assumption that the existence and uniqueness presuppositions of both DPs – *die Botschaft* and *die Absperrung der Botschaft* – can be justified in context. Given this assumption, the DPs can be treated as contributing their referential arguments in the form of discourse referents that represent the unique satisfiers of their descriptive contents.

(5)

$$\left\langle y'_{ref} \mid \boxed{\text{botschaft(y')}} \right\rangle$$

[7]The apparently systematic character of this type of polysemy has been noted as long ago as Bierwisch (1993). Other examples that manifest it are *Schule* (school), *Ministerium* (ministry), *Krankenhaus* (hospital), *die Hypovereinsbank* (a Bavarian Bank that made an exceptional mess of it even by current standards) and so on.

Combining (5) with the semantic representation from the entry for *Absperrung* takes the form of substituting the referential argument y' of (5) for the argument slot symbol $\underline{y}$ in the semantic representation for *Absperrung* and merging the DRS of (5) with the result of this substitution. Furthermore, the arguments of *Absperrung* that are not realised by argument phrases in *Absperrung der Botschaft* are also replaced by real discourse referents. (In the representation (6) below we have retained the same symbols that appear in the top tier of (4).) Of these the discourse referents that represent potential referential arguments of *Absperrung der Botschaft* - viz. e', s_1 and z - are placed in the store of the new representation, together with the 'dummy' referential argument α and the referential argument y' of the argument DP *der Botschaft*.[8] The remaining discourse referents - those which neither represent potential referential arguments nor are realised by an argument phrase - are then existentially bound by entering them into the universe of the new (merged) DRS. In the present instance there is just one such discourse referent, viz. the agent argument x'.[9]

Lastly, the selection restrictions from the 'container' predicate *Absperrung* are incorporated into the representation of *Absperrung der Botschaft*, in the form of *selection presuppositions*, which are collected into a presupposition set that is adjoined to the left of the non-presuppositional component of the representation (Kamp and Reyle, 2010).[10]

The selectiomal restrictions for those arguments that are put into the store are encoded as *selection presuppositions*, those for arguments that are entered into the universe of the DRS are added as conditions of that DRS. We have chosen to keep the restrictions on the different arguments separate, which gives us one presupposition for each argument in the store. Moreover, in the representation below we have omitted the selection restrictions associated with the argument y, since these have done their work already. (They have been used in eliminating the 'message' reading of *Botschaft*, which is incompatible with them. We do not offer a formal reconstruction of this process here). The result is that we end up with three presuppositions, for the arguments e', s_1 and z.

It should be stressed that incorporation of the selectional restrictions as selection presuppositions is to be seen as a general feature of the use of lexical entries of predicates in the DRS construction for sentences in which they occur as container predicates. This applies in particular to the 'container verbs' of *die Absperrung der Botschaft* in the sentences in (1). In the DRS constructions of the next section we will make use of this principle without mentioning it again.

(6)

[8] Recall that the argument phrase *die Botschaft* disambiguates *Absperrung* to the interpretation in which a physical barrier is involved.

[9] When at some later point the represented phrase *Absperrung der Botschaft* will become subject to disambiguation, the discourse referent(s) corresponding to the discarded option or options will be bound in the same way, by being transferred from the store to the universe of the non-presuppositional DRS. From now on we will rely on this principle without mentioning it again explicitly.

[10] In the sequel we will sometimes be less strict on this last point, allowing ourselves to merge two presuppositions into one, for easier reading or for reasons of grahical display.

$$\left\langle \alpha_{ref}\ e'\ s_1\ z\ y' \mid \left\langle \left\{ \begin{array}{l} \boxed{\text{event(e')}}, \\ \boxed{\text{state(s}_1\text{)}}, \\ \boxed{\text{barrier(z)}} \end{array} \right\}, \begin{array}{|c|} \hline x' \ \ s_0 \ \ s_2 \ \ s_3 \\ \hline \text{s}_0 \supset\!\subset \text{e'} \\ \text{s}_0\text{:accessible(y')} \\ \text{s}_1 : \neg \text{ accessible(y')} \\ \text{s}_2\text{: present(z)} \\ \text{CAUSE(e, s}_1\text{)} \\ \text{s}_3\text{:sperr(z,y')} \\ \text{agent(x')} \\ \text{Agent(e') = x'} \\ \text{CAUSE(e', s}_2\text{)} \\ \text{CAUSE(e', s}_{3)} \\ \text{embassy(y')} \\ \alpha = \text{e'} \overset{!}{\vee} \\ \alpha = \text{s}_1 \overset{!}{\vee} \\ \alpha = \text{z} \\ \hline \end{array} \right\rangle \right\rangle$$

(6) has been presented as representation of the NP *Absperrung der Botschaft*. However, since we have decided to ignore the complications connected with the definite article of the DP *die Absperrung der Botschaft* our representation of the DP will also just have the referential argument of the phrase in its store. Thus (6) will serve equally as representation for *die Absperrung der Botschaft*.

In the interest of easier readability we eliminate from (6) all reference to the states s_2 and s_3, as these will play no further part in our deliberations. So from now on we will use (7) as representation for *die Absperrung der Botschaft* rather than (6).

(7)

$$\left\langle \alpha_{ref}\ e'\ s_1\ z\ y' \mid \left\langle \left\{ \begin{array}{l} \boxed{\text{event}(e')}, \\ \boxed{\text{state}(s_1)}, \\ \boxed{\text{barrier}(z)} \end{array} \right\}, \begin{array}{|c|}\hline x'\ s_0 \\ \hline s_0 \supset\subset e' \\ s_0:\text{accessible}(y') \\ s_1 : \neg\ \text{accessible}(y') \\ \text{CAUSE}(e, s_1) \\ \text{agent}(x') \\ \text{Agent}(e') = x' \\ \\ \text{embassy}(y') \\ \alpha = e' \overset{!}{\vee} \\ \alpha = s_1 \overset{!}{\vee} \\ \alpha = z \\ \hline \end{array} \right\rangle \right\rangle$$

3.2 Verbs

3.2.1 abbauen

The first verb is *abbauen* (tear down/take down). *abbauen* is something that can be done to physical objects, especially constructions erected for a temporary purpose, like tents, stands for an open air performance, or – relevant here – fences, walls or barricades erected to ensure the inaccessibility of something. These are not the only kinds of things to which *abbauen* can be applied. Some other possible themes are non-physical entities such as debts or surpluses – here the English equivalent would be 'reduce' rather than 'take down' – and there are other possibilities as well. But what matters for present purposes is that *abbauen* never takes events or states as theme arguments. This entails that in (1)a *Absperrung* can only have its third reading.

It is not all that easy to come up with an accurate and comprehensive statement of the selectional restrictions on the theme argument of *abbauen*. A partial statement, which includes some types of entities as possible themes and excludes certain other types, would be enough to account for the disambiguation in (1)a, but we refrain from formalisation in this case.

3.2.2 behindern

With regard to the selection restrictions for its theme, the verb *behindern* (obstruct, interfere with) differs considerably from *abbauen*. The only possible theme arguments of *behindern* are (i) activities and goal directed actions and (ii) the agents of actions or activities. Moreover, its themes are restricted to intentional actions and activities – things that are done by an agent with a conception of what he is doing or trying to do.

As regards the content of *behindern*, just a few observations will suffice for our needs. (It is the selection restrictions that really matter.) In the lexical entry (E2) for *behindern* that is given below we hide all aspects of its meaning behind the constant ‘behindern’ of the DRS language in which our semantic representations are couched.

Important for our present purposes is that in cases where its theme is a goal-directed action, *behindern* carries no implications about the completion of that action: for all that *behindern* says, the action may be completed – in which case the interference may have been a nuisance but no worse. But it is also possible that the interference was so crippling that it forced the agent to give up. Our entry (E2) captures this intuition by virtue of omission: it simply doesn’t say anything about the completion prospects of actions that occur as theme arguments of *behindern*. The effect of this omission is that when the DRS for sentence (1)b has been built with the help of this entry and the underspecified DRS for *die Absperrung der Botschaft* and is then converted into an integrity constraint, the inference machinery of the Event Calculus will make it possible to infer that the action did reach completion.

(E2) **behindern**

behindern	verb	nom	acc
	e	x	y
SEL.RES:	event(e)	agent(x) ∨ event(x)	action(y) ∨ activity(y) ∨ agent-of-an-action-or-activity(y)
SEMANT:	e: behindern(x,y)		

The lexical entries for *Absperrung* and *behindern* are the principal resources we need to construct the DRS for (1)b. We assume that it is during the construction of this DRS that the selectional restrictions of *behindern* exert their disambiguating effect on *Absperrung*. To see how this might work in some detail is useful, we believe, in that it provides insight into what must be involved even in the comparatively simple disambiguation procedures that rest exclusively on selectional restrictions. As will become clear later on in this section, the case of (1)b is simple even as cases of disambiguation via selection restrictions go. It is a good first introduction not only to this kind of disambiguation but also to the more general topic of online disambiguation in the course of DRS construction.

First, a minor complication arises from the fact that the sentences in (1) are passives. We have chosen passive rather than active sentences in order to avoid certain irrelevant complications we would otherwise have to deal with when we turn to semantic representation at the level of CLP. But a small price has to be paid for this decision, and it has to be paid right here. The reason why it has to be paid here is simply that it hasn’t been paid before: To our knowledge there exists no treatment of passives within DRT that suits our present purposes.[11] What suits our current interests best – even if it isn’t the best possible account of passives from a theoretical perspective – is to assume a general rule that transforms transitive verbs into the corresponding passive forms, which then behave

[11] Roßdeutscher (2000) discusses a compositional analysis of the passive as built transparently from past participles and the verb *werden* (the German passive auxiliary). We think that such a deep analysis of passives in German and other languages is probably the right way to go. See also Sternefeld (1995). But to compute the DRS for (1)b from a syntactic structure which reflects such an in depth account of passive formation would defeat the point of making things simpler for ourselves when we come to the transition from DRSs to integrity constraints.

syntactically like compound intransitive verbs. The transformation also carries information relevant to 'linking theory': the argument of the verb that is realised as direct object when the input verb is used in the active vioce is realised as subject of the output verb (the passive form of the input verb), whereas the argument that is realised as grammatical subject in active uses is optionally realised by a *von*- or *durch*-PP.[12] We will not state the rule which transforms verbs into their passive forms in general terms, but just show what it does to the verb *behindern* of (1)b. The operation is given in (8).

(8)

behindern	verb	nom	acc
	e	x	y

⇒

behindert werden	verb	(*von*- or *durch*-PP)	nom
	e	x	y

We haven't bothered to make specific in (8) how selectional restrictions and semantic representation of *behindern* are transferred to *behindert werden*, but this is done in the obvious way,

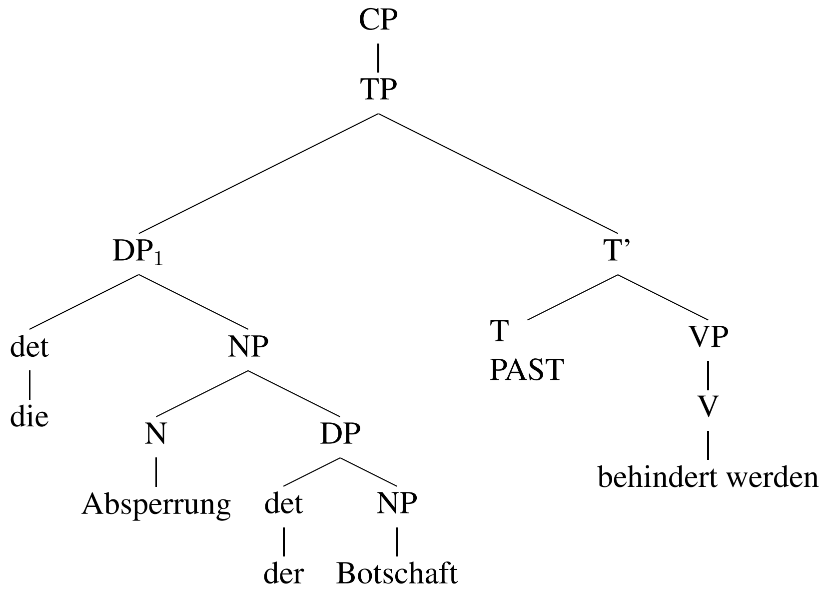
Passivisation of the other verbs that occur in (1) is assumed to follow the exact same pattern. We will not mention passivisation again

The selection restrictions and the semantic representation of *behindern* are transferred unaltered to *behindert werden*. This is a general feature of the passive transformation which (8) exemplifies. Because of this general specification it is unnecessary to mention selectional restrictions and semantic representation explicitly in any particular instance of the general rule (such as (8)).

Since the syntax treats the output of (8) as an intransitive verb, it will analyse (1)c as an intransitive sentence. We assume its syntactic structure to be as in (9).

(9)

[12] By 'linking theory' we understand that component of the grammar which assigns the phrases that occur in a syntactically well-formed clause or phrase to argument slots of the main predicate of that clause or phrase. As indicated earlier, we assume that the lexical entries of verbs mark which arguments of the verbal predicate are obligatorily and which are optionally realised. Part of verifying a string as a well-formed expression is to assign DPs and PPs in the string to argument slots such that each obligatorily realised argument slot is assigned a phrase, (Optionally realised arguments can but need not be assigned a phrase; phrases that are not assigned to any argument are identified and treated as adjuncts.) We rely on the principle that a parser which verifies an input string as well-formed by assigning it a syntactic structure must implement linking theory in any case. (It must establish, as part of verifying the string's well-formedness, the right correspondences between argument phrases and argument slots.) The links between phrases and slots that are determined in the proicess can be included in the parser's output and thus be made available for the semantiic interpretation component.



The first step in the construction of the DRS for this syntactic structure consists in replacing the occurrence of the verb *behindert werden* by its semantic representation (as given in the lexical entry for *behindern*). Since for the next construction steps the selection restrictions will be crucial, it is necessary to include these as part of the information that is passed on for combining with the semantics of the subject phrase. In doing so we rely once more on the intuitive principle that selection restrictions act as presuppositions which are left-adjoined to the non-presupppositional component of the representation of the sentences or sentence parts within which the presuppositions are triggered. In the semantics which replaces *behindert werden* in (9) the presupposition set will consist just of the selectional presuppositions contributed by the entry for *behindert werden* (including the uniqueness presupposition concerning the theme argument y).

As in other recent work on DRS construction (Kamp, 2001; Genabith et al., 2010; Kamp and Reyle, 2010), we assume that some of the discourse referents that are introduced in the process of constructing the semantic representation need not be bound instantly, but can be put into a store to await binding at some later stage of the construction.[13] In the representations we are using the store is presented to the left of the presupposition set.

Lexical predicates such as verbs and nouns are assumed to have one referential argument and zero or more non-referential arguments. The referential argument is introduced by the predicate word itself, and is never realised by an argument phrase. When the pred-

[13] If all types of binding are cast in the form of operations on lambda-abstracts, and these operations always occur in the same, predictable order, then stores could be dispensed with in favour of lambda-abstraction over discourse referents that are not bound instantly. But this is an assumption that seems to us unwarranted even if it were technically possible to set the syntax-semantics interface up in a manner which allows us to treat all cases of binding as operations of this particular form. On the other hand, the possibility to put discourse referents into the store is only one half of the story. The other, much harder, half is to spell out in precise detail when a discourse referent is to be taken out of the store again and how it may or must be bound at that point. This is not the place to go into this part of the story, but at one or two points we will refer to binding principles that would have to be part of it. (For the record we should mention that at this point in time no comprehensive description of store retrieval for a substantial explicitly defined natural language fragment exists).

icate is a verb, the referential argument is the event or state described by it. The verb's non-referential arguments are those which in traditional treatments are simply called its 'arguments' – subject, direct object, indirect object, .. . According to the present account of argument structure a verb or other kind of predicate word introduces its own referential aregument (a discourse referent), but makes its non-referential arguments available only in the form of *argument slots*. When a predicate word occurs as head of a well-formed phrase or clause, then all its argument slots must eventually be filled by discourse referents. Those argument slots that are linked to argument phrases will be filled by the discourse referents that act as referential arguments of these prhases. Slots without links to argument phrases must be filled by same kind of default procedure'implicit argument interpretation'. For the construction of semantic representations the difference between arguments and argument slots is important and needs to be marked explicitly. To this end we distinguish between 'real arguments' (discourse referents) and *argument slot symbols*. For the latter we are using underlined letters (e.g, $\underline{x}$), while symbois for discourse referents are as usual plain letters (without underlining). In lexical entries this distinction is not being enforced. There is no need for it here, since the relevant information can be directly recovered from the form in which our entries for predicate words are specified. Lexical entries of this form distinguish between the referential argument, which is presented directly below the entry's lemma, and zero or more non-refrerential arguments, presented to its right. When the semantic representation from the lexical entry of a predicate word is inserted for the word in the course of constructing the representation for a sentence in which the word occurs, then the referential argument from the entry is replaced by a discourse referent (which is usually put into the store of the representation under construction, as well as being inserted into the relevant argument slot), whereas the entry's non-referential arguments are replaced by argument slot symbols. (Eventually these slot symbols will have to be eliminated through replacement by discourse referents.)

The representation that replaces *behindert werden* under the V-node in (9) is given in (10).

(10)

$$\left\langle \text{e}_{ref} \mid \left\langle \left\{ \boxed{\text{event(e)}}, \boxed{\begin{array}{c}\text{agent}(\underline{\text{x}})\\ \vee\ \text{event}(\underline{\text{x}})\end{array}}, \boxed{\begin{array}{c}\text{action}(\underline{\text{y}}) \vee\\ \text{activity}(\underline{\text{y}})\\ \vee\ \text{agent}(\underline{\text{y}})\end{array}} \right\}, \begin{array}{|c|}\hline \\ \hline \text{e: beh'ern}(\underline{\text{x}},\underline{\text{y}}) \\ \hline\end{array} \right\rangle \right\rangle$$

In the passage from V to VP nothing happens; in the transition from VP to T' e is located in the past of the utterance time n, via a location time t introduced by the information attached to T. For reasons that we need not go into here, we assume that t joins e in the store. Both will be bound existentially in the final construction step, the transition from TP to CP. This gives (11) as representation associated with the T' node.

(11)

$$\left\langle \text{t}_{ref}\ e \mid \left\langle \left\{ \boxed{\text{ev't(e)}}, \boxed{\begin{array}{c}\text{agent}(\underline{\text{x}})\\ \vee\ \text{event}(\underline{\text{x}})\end{array}}, \boxed{\begin{array}{c}\text{action}(\underline{\text{y}}) \vee\\ \text{activity}(\underline{\text{y}})\\ \vee\ \text{agent}(\underline{\text{y}})\end{array}} \right\}, \begin{array}{|c|}\hline \\ \hline \text{t} < \text{n} \quad \text{e} \subseteq \text{t} \\ \text{e: beh'ern}(\underline{\text{x}},\underline{\text{y}}) \\ \hline\end{array} \right\rangle \right\rangle$$

We now come to the decisive construction step, in which the representation of the subject DP *die Absperrung der Botschaft* is combined with that of the T' node. This step is comparatively simple in that only one of the two representations (that for *die Absperrung der Botschaft*) is ambiguous.[14] But even so there is work to be done. The procedure that combines the two part representations has to take each of the alternatives of the ambiguous representation and try to combine it with the representation of its sister node (here the T' node), test for sortal consistency and keep the combined representation if and only if the test is positive. The result is either an aborted representation (when all combinations fail) or else a new set of one or more alternatives (that or those which survive the consistency test).

We have set up the representations in a way which makes sortal consistency testing comparatively straightforward. Semantically, the combination of the representation of the subject DP and the structure under T' is a case of argument insertion. In this respect the construction step before us resembles the combination of the DP *die Botschaft* with the semantics of the noun *Absperrung*: the referential argument (α) of the argument phrase gets substituted for the symbol that represents the slot of the container predicate with which the phrase has been linked ($\underline{y}$), whereupon the stores, presupposition sets and the non-presuppositional DRSs of the two representations get merged. (This time both representations have non-empty presupposition sets. The representation that results is shown in (12).[15]

[14] In this respect the present case differs from that of *Absperrung der Botschaft* as discussed in Section 3.1, where both constituent representations, the one for *Absperrung* and that for *die Botschaft*, were ambiguous. Cases where both of two representations that have to be combined in the course of DRS construction are ambiguous are common enough. Another example, which will be discussed below, is sentence (1)e.

[15] The semantic representation associated with the T' node contains not only the information contributed by the verb *behindern* but also that contributed by the past tense of (1)b. Tense has contributed the discourse referent t, which at this point plays the role of referential argument (hence the subscript *ref* to its occurrence in the store). None of this is of any direct concern to us here, so we do not elaborate. For a presentation of the treatment of tense in DRT that we are assuming here, see Reyle et al. (2008) or Genabith et al. (2010).

(12)

$$\left\langle \mathrm{t}_{ref}\ e\ \alpha\ e'\ \mathrm{s}_1\ z\ y' \mid \left\langle \left\{ \begin{array}{l} \boxed{\text{event(e')}}, \\ \boxed{\text{state(s}_1\text{)}}, \\ \boxed{\text{barrier(z)}}, \\ \boxed{\text{event(e)}}, \\ \boxed{\begin{array}{c}\text{agent}(\underline{\text{x}}) \vee \\ \text{event}(\underline{\text{x}})\end{array}}, \\ \boxed{\begin{array}{c}\text{action}(\alpha) \vee \text{activity}(\alpha) \\ \vee\ \text{agent}(\alpha)\end{array}} \end{array} \right\}, \begin{array}{|c|}\hline s_0\ \ s_1\ \ x' \\ \hline \text{t} < \text{n} \ \ \text{e} \subseteq \text{t} \\ \text{s}_0 \supset\!\subset \text{e'} \ \ \text{e'} \supset\!\subset \text{s}_1 \\ \text{s}_0\text{:accessible(y')} \\ \text{s}_1 : \neg \text{ accessible(y')} \\ \text{CAUSE(e, s}_1\text{)} \\ \text{agent(x')} \\ \text{Agent(e') = x'} \\ \text{embassy(y')} \\ \text{e: behindern}(\underline{\text{x}}\text{, e')} \\ \alpha = \text{e'} \overset{!}{\vee} \\ \alpha = \text{s}_1 \overset{!}{\vee} \alpha = \text{z} \\ \hline \end{array} \right\rangle \right\rangle$$

The compatibility check that now has to be performed on (12) consists in determining which of the options for α that are offered by the $\overset{!}{\vee}$-disjunction in the DRS on the right in (12) is compatible with the selection restrictions that *behindern* imposes on α. (This selecton restriction is the condition 'action(α) $\vee$ activity(α) $\vee$ agent(α)', which occurs as one of the presuppositions in (12).) The effect of this check is that when α is identified with one of e', s_1, z) and the selection constraint imposed on the chosen discourse referent by trhe relevant presupposition in (12) is incompatible with those that (12) imposes on α, then that choice is eliminated. Only choices which do not lead to incompatibility are retained as possible options. It is clear that in the case at hand there is only one compatible choice, viz. e'. So we infer that in (1)b *die Absperrung der Botschaft* is fully disambiguated, with only the event reading remaining. In the DRS (13) for (1)b the results of the compatibility check have been incorporated: the $\overset{!}{\vee}$ condition has been discarded and all other occurrences of α have been replaced by e'.[16]

To obtain the DRS for (1)b from (12) some further operations are needed. First, the discourse referents in the store need to be bound and the argument slot ($\underline{\text{x}}$) must be filled in some way. We assume that in cases like the one before us, where the remaining syntactic transition is that from TP to the CP node of a main clause, both slot filling and store binding are existential operations. For the discourse referents in the store existential binding can be implemented by transferring them from the store to the universe of the non-presuppositional DRS of the representation. The unfilled argument slot requires a double

[16]While it seems intuitively plain that the consistency check will discard the state reading and the object reading of *die Absperrrung der Botschaft*, a word of caution is called for. What are the formal principles behind these results - the two inconsistency and the one consistency result? Evidently, a formal reconstruction of these inferences must rely on some kind of formal ontology, which captures the logical relations – of subsumption, incompatibility and so on – between such 'onto-semantic' primitives as 'action', 'activity', 'event', 'physical object' and the like. In other words, a strictly formal version of the account we are presenting would have to include as one of it components a formal ontology, of the kind developed by, for instance, the Trento group led by Nicola Guarino or the ontologies of systems like WordNet, Framenet or Cyc. Guarino and Welty (2000); Lenat (2006). Whether any existing formal ontology will provide what is needed here is a topic for further research.

operation: first the slot must be filled with a fresh discourse referent, and this discourse referent is then bound existentially through placement in the relevant DRS universe.[17]

The one remaning operation concerns the presuppositions of (12). Now that, as a result of the compatibility check, the sortal presuppositions have been established as consistent, they can be regarded as satisfied and incorporated into the representation of the non-presupppositional content of the sentence. In accordance with this principle the prsuppositions of (12) that have survived the consistency check have been incorporated into th DRS (13) for the sentence (1)b.

(13)

$e\ \ t\ \ x\ \ e'\ \ x'\ \ y'\ \ z\ \ s_0\ \ s_1$
t < n event(e) $e \subseteq t$ agent(x) $\vee$ event(x) Agent(e) = x event(e') action(e') $\vee$ activity(e') $\vee$ agent(e') $s_0 \supset\subset e'$ agent(x') Agent(e') = x' fence(z) $\vee$ wall(z) $\vee$ barricade(z) s_0: accessible(y') s_1:$\neg$ accessible(y') CAUSE(e,s_1) embassy(y) e: behindern'(x,e')

3.2.3 verhindern

The verb *verhindern* (Engl. *prevent*) has certain features in common with *behindern*, but in the light of our concerns the differences are at least as important as the similarities. The first difference concerns the selection restrictions on the theme argument. *behindern* selects for agents as well as actions and activities. (There is a sense in which activities are the primary themes; even when the theme phrase refers to an action or an agent, it is an associated activity - the preparatory activity that is part of the action or the activity in which the agent is engaged - that is directly involved in *behindern*. But nevertheless, theme phrases that describe goal-directed actions sound perfectly natural with *behindern*.) *verhindern* differs in two ways. On the one hand it is more liberal with regard to its posible themes in that it also admits theme phrases that denote states.[18] But on the other hand it

[17] One problem with the existential default binding of unfilled slots is that the discourse referents that are used in this process do not become available as antecedents for subsequent anaphoric pronouns. This is a more pervasive problem in DRT, which arises for many extensions of the early DRS languages presented in Kamp (1981) or Kamp and Reyle (1993), but is usually ignored. Various technical solutions have been suggested for this problem, among them (i) creating sub-DRSs in whose universes the fillersof the slots are placed, making them thereby inaccessible to subsequent pronouns later on, and (ii) the use of different types of discourse referents. Here we follow a somewhat reprehensible tradition of paying the problem no attenton after having pointed out that it exists.

[18] For instance, among the entities that can be themes of *verhindern* are water shortages, famines and other disastrous conditions; in this respect *verhindern* is just like the English verb *prevent*. Actually, it is hard in this connection to know where to draw the line between states and events. Is a case of torrential rain fall an event or a state? If we classify it as a state, what about the shower that caught me just as I stepped outside, but was over half a minute later; if we classify it as an event, what about the drought that preceded it? There is, we believe, an ineliminable element of ambivalence here, which is closely connected with the choice between an internal and an external viewpoint that often presents itself to us as an option when we

seems to admit theme phrases that denote persons only marginally, and to the extent that it does, it treats them as non-agents.[19] Connected with the fact that a properly agentive interpretation does not seem possible for theme phrases of *verhindern* is the way we interpret theme phrases that denote activities. The combination of *verhindern* with such a theme phrase means that the activity did not take place at all; it cannot mean that the activity was started but then cut short.

As far as the agent argument is concerned agent *behindern* and *verhindern* seem to behave alike. Both admit as 'agents' conscious agents (individuals or groups of people) as well as events (both natural events: rains, storms, floods, and man made ones: demonstrations, decisions by governing boards) and also rules, laws and regulations. Important in connection with our next verb, *unterbrechen*, is that the 'agent' of *behindern* and *verhindern* must always be distinct from the agent of the action or activity that is being interfered with or prevented, or from the bearer of the prevented state. Both *behindern* and *verhindern* describe interference from the outside. For instance, though it is perhaps not inconceivable that someone should prevent his own action, the verb *verhindern* just isn't made to describe such a situation.[20]

What matters most for the disambiguation of *Absperrung* in (1)c is that *verhindern* accepts both events and states as theme arguments. This means that *Absperrung* in (1)c can be interpreted both as the event of fencing off and as the state of being fenced off. Thus, unlike in (1)b, disambiguation is partial - only the physical object interpretation of *Absperrung* is excluded.[21]

But although (1)c is still ambiguous in that *die Absperrung der Botschaft* can be read either as referring to an event or to a result state, this ambiguity is of no great consequence, since in either case we end up with the same truth conditions for (1)c. In order to secure the truth-conditional equivalence of these two interpretations of (1)c, our semantics must make explicit what the connection is between preventing states and preventing the events that lead to them. We propose to state this connection in the form of a Meaning Postulate (see (14) below). But before we are in a position to formulate this postulate there is another matter about the entry for *verhindern* that must be sorted out first.

This brings us to what is for us the most important respect in which *verhindern* differs from *behindern*. This difference has to do with what might be called the 'privative' di-

give a verbal description of an given episode. (For the distinction between internal and external viewpoint see Smith, 1991).

[19] *Sie haben Lafontaine verhindert* (lit.: 'They prevented Lafontaine') can be used to say that they prevented Lafontaine from becoming prime minister, but could hardly be used to describe a situation in which he is forced to abandon some goal-directed activity, such as the painting of his garden fence, in mid-stream, so that the painting job remains incomplete.

[20] In English you can say 'He prevented himself from ...'. But no corresponding construction is possible for *verhindern*.

[21] Arguably, even the physical object interpretation is not fully excluded in (1)c. It seems just about possible to say 'Sie verhinderten den Zaun' ('They prevented the fence.') with the intended meaning that they prevented the planned erection of the fence. This is similar to the case of Lafontaine mentioned in the one but last footnote.

We believe that the interpretation of theme phrases of *verhindern* of the sorts at issue here (those that denote neither events nor states) are best dealt with via coercion of the kind discussed in Pustejovsky (1995). We haven't included a module that deals with coercions of this kind from the proposal we make here. As a consequence the physical object reading of *Absperrung* in (1)c is excluded. This would change if a coercion module were added.

mension of the meaning of *verhindern*. The concept denoted by *verhindern* is 'privative' in the sense that 'x verhinderte α' entails that there was no α.[22] The privative character of *verhindern* brings a new complication to the formulation of this verb's lexical entry. The combination of *verhindern* with a theme argument phrase α, we have just noted, asserts that the event described by *verhindern* produces the effect that there was no token eventuality instantiating the description provided by α. This means that the verb cannot be treated as involving a relation between its referential argument (the event it describes) and a token eventuality (described by its theme argument phrase); for there is no such token eventuality. Rather, we must treat *verhindern* as denoting a relation between the event that it describes and the eventuality *type* described by its theme argument phrase - a relation that hold between the described event e and an eventuality type EV when e has the effect that EV is not instantiated.[23]

The way we implement this 'second order' analysis of the theme argument of *verhindern* is as follows. We represent *verhindern* as a relation between its referential argument e and an event or state type A which obtains iff e results in the non-instantiation of A. When *verhindern* occurs as part of a sentence S that is used in a context C, A must be retrieved from sentence and/or context. In a sentence like (1)c, in which the theme argument is a description, it will be the descriptive content of the phrase that furnishes A. (Thus in the case of (1)c it is the type described as 'Absperrung der Botschaft'.) Formally, the descriptive part of the description will yield a DRS K and A can then be identified with the λ-term λev.K, where *ev* represents the 'fence off' event or 'fenced off' state that *Absperrung* is taken to describe in the given context. But the theme argument phrase of *verhindern* isn't always a DP with an explicit descriptive content - for instance, it could be a pronoun - and in such cases the mechanism for retrieving A will be more indirect. We represent the need for A to be retrieved from the context in the form of an 'anaphoric presupposition' (Genabith et al., 2010). (The underlining of A in the universe of the DRS representing this presupposition indicates that resolution of the presupposition must yield a specific type denoting term that can replace A - see the entry E3 for *verhindern* be-

[22] That 'x verhinderte α' can be true only if no α was realised is uncontroversial when α is of the sort of an activity or a state. But the principle also holds, we claim, when α describes a telic event. For example, consider the action description *die Botschaft absperren* ('to fence off the embassy'). *verhindern* can be used correctly in combination with this verbal description – for instance in the form of the infinitival complement construction *Sie verhinderten das Absperren der Botschaft* ('They prevented the fencing off of the embassy') or, alternatively, that of the corresponding *-ung*-noun, as in (1)c – even in a situation where a start has already been made with the fencing off but where a decisive intervention then puts an end to it, so that the state of the embassy being fenced off is not reached. In this case there will have been an instance of the activity that is associated with the action type denoted by the theme phrase. But that token activity is *not* an instance of the action type itself. (It is not an event that results in the embassy being fenced off.) So here too *verhindern* entails that the description that is used to characterise its theme is not instantiated.

[23] Our proposal for dealing with the semantics of *verhindern* is reminiscent of that of for the treatment of intensional verbs such as *suchen* ('seek'). Zimmermann analyses such verbs as relations between the subject and a property. (Thus the relation denoted by *suchen* holds between agent x and property P iff (roughly) x wants to find a token that instantiates P.) Our analysis of *verhindern* is like Zimmermann's analysis of *suchen* except that in our case the subject is not concerned to find a token of the type provided by the theme phrase but instead prevents a token from coming into existence.
A discussion of privative verbs like *prevent* can be found in Condrovadi et al. (2001). This paper makes the same central point that emphasized in our discussion: the argument of *prevent* needs some form of second order analysis. The details of their proposal and ours, however, are, due to differences in the general representational/logical form formalisms employed quite different.

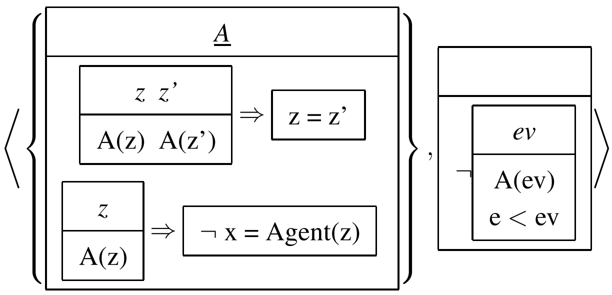
low.) Our statement of this presupposition will be incomplete in that it fails to articulate what mechanisms are availble for the retrieval of A in the different contexts in which *verhindern* can occur.[24]

One constraint that *verhindern* imposes on A is that A be uniquely instantiated, in the sense that it can have no more than one instance. When the theme phrase is a singular definite description and A is extracted from it in the way indicated above, then the existence-and-uniqueness presupposition associated with the descriptive phrase will guarantee uniqueness of A. But the constraint seems to hold generally, and we have built it into the anaphoric presupposition for A.

(E3) **verhindern**

verhindern	verb	nom	acc
(i)	e	x	A
SEL.RES:	event(e)	agent(x) $\vee$ event(x)	event type(A) $\vee$ state type(A)

SEMANT:



Among the event types that can be theme arguments to *verhindern* there are, we noted, in particular goal-directed actions and also result-oriented non-agentive event types. Such action and event types are sometimes described as 'target state' types: with the event type E is associated a state type S such that whenever E is instantiated by a token event e, there is an instance s of S that is the result of e. We denote the target state relation between event types and state types as 'TARGETST'. Thus 'TARGETST(E,S)' says that every instance e of E results in an instance s of S. We will assume that uniqueness of E entails uniqueness of S: if there can be at most one instance of E, then TARGETST(E,S) entails that there also can be at most one instance of S, so if e is the unique instance of E, then the unique instance of S will be the state that e results in.

The predicate 'TARGETST' enables us to state the Meaning Postulate that connects interpretations of *verhindern* in which the theme argument is a 'target state event type' with interpretations in which the theme argument is the corresponding result state type.[25]

[24] This is a general shortcoming that can be found in most current presupposition accounts: little if anything is said about the 'accommodation' available possibilities for presuppositions that cannot be resolved in the context as is. An exception is the extensive work on the limited accommodation options for anaphoric pronouns.

[25] Stating this postulate in DRS notation is a bit awkward. But the content of the postulate is simple enough: for the given E and S it is the case for any instance e of E and subject x that: 'e: verhindern'(x,E) if and only if e: verhindern'(x,S)'.
Here 'e: verhindern'(x,A)' is short for something like 'Cause(x,K)', where K is the DRS

(14) a.

$$\begin{array}{|c|}\hline e\ \ x \\ \hline \text{TARGETST(E,S)} \\ \text{e: verhindern'(x,E)} \\ \hline \end{array} \Rightarrow \boxed{\text{e: verhindern'(x,S)}}$$

b.

$$\begin{array}{|c|}\hline e\ \ x \\ \hline \text{TARGETST(E,S)} \\ \text{e: verhindern'(x,S)} \\ \hline \end{array} \Rightarrow \boxed{\text{e: verhindern'(x,E)}}$$

(14) guarantees that it won't make any difference to the truth conditions that get assigned to (1)c whether *Absperrung* is taken in its state or its event sense. Using (14) we can deduce the sentence representation that results from the first choice from the one that results from the second, and vice versa.

3.2.4 Shifting die Absperrung der Botschaft

The entry of *verhindern* can be used to construct a DRS for (1)c in much the same way as we constructed the DRS for (1)b using the entry for *behindern*. But there is one crucial difference. The theme argument y specified in the entry for *behindern* is of the type of an (activity, action or agent) *token*. But in the case of *verhindern* we saw that it was necessary to interpret the theme argument (A) to be an (event or state) *type*. This theme argument specification leads to a clash with the referential argument of *die Absperrung der Botschaft* and it is one that arises irrespective of which reading of *Absperrung* we choose. Clashes of this kind - let us refer to them as 'type-token clashes' - are no reason for dismissing the combination as incoherent. Rather, they trigger instances of legitimate *type shifting* in the spirit of e.g. Partee (1987), Partee and Rooth (1983) – or, as we will call this operation here in order to forestall terminological confusion, *token-to-type shifting*

In the case before us the only permissible token-to-type shifts that restore compatibility between argument phrase and argument slot are those that lift the token category of the entity denoted by the argument phrase to the corresponding type category.[26] We will

---

(i)

$$\begin{array}{|c|}\hline \\ \hline \neg \begin{array}{|c|}\hline ev \\ \hline \text{A(ev)} \\ \text{e} < \text{ev} \\ \hline \end{array} \\ \hline \end{array}$$

Note well: the relation 'Cause' is different from the relation 'CAUSE' that can be found in the DRSs we have shown up to this point. 'Cause' is not a relation between token eventualities, but between 'causers' (token eventualities or agents) and facts. Here we understand by 'fact' something like 'that which is responsible for the truth of a proposition' (formalisable for instance as a pair consisting of a possible world and a proposition that is true in it). The logical connection between the predicates 'CAUSE' and 'Cause' is a non-trivial matter, but it is not one that we want to go into here.

[26]Our use of the term 'category' in this paper may need some elucidation. We are using 'category' strictly in the sense of 'semantic category' and in fact interchangeably wth the term 'sort'. Both terms are to be understood as relating to what one tries to capture in conceptual hierarchies like those of *Wordnet* or *Cyc* (Fellbaum, 1998; Lenat, 2006). In particular, selectional restrictions for argument slots are constraints on the categories or sorts – as we use the terms – of the arguments that may fill those slots, and the expression

assume that this operation is always possible in case an argument slot of a predicate word is specified (in the lexical entry for that word) as being of a certain *type* category and the argument phrase linked with that slot is of the corresponding *token* category. In all such cases the argument phrase is lifted to the corresponding type level (i.e. to the level of the type of its token category) and it is the referential argument of the raised argument phrase that is then inserted into the argument slot. It is only after this token-to-type shift has taken place that compatibility checks between argument category and selection restrictions on the slot come into play.

For the construction of the DRS for (1)c the implications are clear: In order to combine the representation of the subject DP with that of the T' node we first need to subject the DP to token-to-type shifting. To describe the general procedure for this will be our next task.

The task is straightforward when the representation that presents its referential argument for slot insertion is not ambiguous and has no presuppositions. In such cases the representation will consist of a store followed by a DRS K; the store will contain the referential argument - let us assume this is the discourse referent y - as one of its elements (possibly as its only element). The new representation, of the type, which is to replace the old representation of the token, is obtained as follows:

1. form the λ-*abstract* of K with respect to y; this results in a λ-term which we denote as 'λy.K'.[27]
2. replace y in the store by 'λy.K' with the subscript 'ref' to indicate that this term now functions as referential argument.[28]

'sortal' ambiguity' as it applies to nouns like *Absperrung* is also used in this sense. Exactly what 'sorts', or 'categories' in this sense of the terms are – by what criteria they are identified, how many of them there are, what subsumption relations (and other structural relations perhaps) obtain between them – is an important but difficult conglomerate of questions: we do not address these here, even though they are clearly relevant to the disambiguation issues that are our central concern here.

What matters in particular about sorts/categories at this point of our discussion is that for us the hierarchical organisation of sorts/categories that endeavours such as Wordnet and Cyc try to capture is in a certain sense orthogonal to the type-token distinction that is alluded to in the term 'token-to-type shifting'. Both sorts/categories and types, as we use the terms here, are properties of individuals/tokens. But 'type' is the more general term. For instance, λ-abstraction with respect to a variable (or discourse referent) for individuals over a formula (or DRS) always yields a term which denotes a type, but not necessarily all of these are sorts. In what follows we will need not only token sorts (those that populate sortal hierarchies like those of Wordnet of Cyc) but also 'type sorts'– higher order sortal predicates that apply to types rather than to tokens. Thus, corresponding to the token sort predicate 'event' we will need a type predicate 'event type' which is true of a type E iff all instances of E necessarily satisfy 'event'. (More about this below.)

A simple and proper way of expressing these various distinctions can be given within a framework of Higher Order Many Sorted Logic. But that is not for here.

[27]The possibility of abstracting over DRSs is central to λ-*DRT* (Kohlhase et al., 1995; Kuschert, 1996). See also Blackburn and Bos (2009). For a modest use of λ-abstraction within DRT, which is commensurable with the use we make of λ-abstraction here, see Genabith et al. (2010).

[28]The new representation differs from those we have seen so far in that its store contains not only discourse referents; to construct DRSs for a sentence like (1)c we also need to be able to put certain λ-terms there. However, the only reason for putting a λ-term into the store is to have it available as referential argument of the represented phrase, so that it can be inserted into the argument slot with which the phrase is linked when the time for this has come. We observe in this connection that the denotation of a λ-term is fully determined by its structure. So there is no further need for binding, in the sense in which a discourse

3. replace the DRS K by the empty DRS $\emptyset$.

These operations transform the initial representation λy.K into one of the form '$\langle \lambda y.K_{ref} \mid K' \rangle$', where (in this case, but see below) K' = $\emptyset$.

Note that, as was assumed implicitly in the last footnote, argument insertion after token-to-type shifting of the argument phrase involves substitution of a λ-term and not of a plain discourse referent.

The token-to-type problem we are facing differs in two respects from the case described above: the semantic representation that needs lifting – representation (7) for the phrase *die Absperrung der Botschaft* – is (a) ambiguous and (b) it has presuppositions. First consider the ambiguity disjunction 'α = e' $\overset{!}{\vee}$ α = s_1 $\overset{!}{\vee}$ α = z'. Since the token-level referential argument α is now going to be lifted and replaced by a new argument term of the form '$\lambda\alpha$.K of the corresponding type, the ambiguity condition has to lifted accordingly - it now is to get the form: '$\lambda\alpha$.K = E' $\overset{!}{\vee}$ $\lambda\alpha$.K = S_1 $\overset{!}{\vee}$ $\lambda\alpha$.K = Z', where E', S_1 and Z are discourse referents representing types corresponding to the token discourse referents e', s_1 and z. These type discourse referents replace e', s_1 and z in the store, while e', s_1 and z are transferred from the store to the universe of the non-presupositional DRS K. Moreover, conditions which say that e', s_1 and z are instances of the types represented by the new discourse referents - viz the conditions 'E'(e')', '$S_1(s_1)$' and 'Z(z)' – are added to the condition set of K.

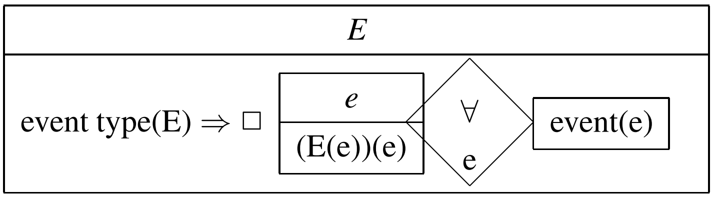
The selection presuppositions that involve e', s_1 and z also need lifting to selection presuppositions pertaining to the corresponding types E',S_1 and Z. (This is crucial since the selectional restrictions on the theme argument of *verhindern* are constraints on types (rather than constraints on tokens) and it is against these that the presupopopsitonal constraints on the referential argument of *die Absperrung der Botschaft* have to be checked for consistency.) Lifting of these presuppositions requires sortal predicates of types corresponding to the sortal predicates in the original presuppositions (those given in (7)). Consider for instance the presuppositioon 'event(e')'. Its sortal predicate 'event' is now to be replaced by a predicate of types. We call this predicate (which by the way has already been used in the lexical entry for *verhindern*) 'event type'. So the presupposition 'event(e')' gets replaced by the condition 'event type(E')'.

There is an obvious connection between the prediates 'event' and 'event type' that we already alluded to in footnote 24: the latter predicate is true of those types which are necessarily instantiated only by tokens of which the former predicate is true. This connection is expressed in the Meaning Postulate (15).[29]

referent occurring in the store has to be bound at some point. (In this sense λ-terms are 'costants' whereas discourse referents are 'variables'.) This means that when a λ-term occurring in the store of an argument phrase is inserted into the relevant argument slot, it can be simply removed from the store - its denotation will be secure in any case. It also follows from this that λ-terms occurring in stores always play the part of referential arguments; so they always bear the subscript 'ref'.

[29]Meaning Postulates like (15) are relevant in that they enter into the consistency checks that are responsible for the cases of sortal disambiguation discussed here. For instance, the incompatibility of the constraints 'event type' and 'state type' derives, via the Meaning Postulates for 'event type' and 'state type', from the incompatibility between the predicates 'event' and 'state'; and so forth. These Meaning Postulates show their real use only in conjunction with a formal ontology, in which the incompatibility of, for instance, the

(15)



The presuppositional conditions ‘state(s_1)’ and ‘barrier(z)’ are litfed in analogous fashion to the conditions ‘state tpye(S_1)’ and ‘barrier type (Z)’.

This completes our informal description of the operations involved in the token-to-type shifting of (7). We summarise the different operations for a better overview of what is necessary,

Consistent with our description above of lifting unambiguous representations without presuppositions, let K be the non-presuppositional DRS of (7). Before we perform λ-abstraction over K with respect to α K has to be subjected to the changes specified in (a)-(c):

(a) the $\overset{!}{\vee}$-condition is removed from K.

(b) the discourse referents that are mentioned in the $\overset{!}{\vee}$-condition - e’, s_1 and z - are transferred from the store to the universe of K_0.

(c) added to the condition set of K are conditions that relate the discourse referents e’, s_1 and z to the corresponding type discourse referents E’, S_1 and Z by stating that the former are instances of the latter. We express these conditions as ‘E’(e’)’, ‘S_1(s_1)’ and ‘Z(z)’.

Next the following operations must be performed:

1. form the λ-term $\lambda\alpha$-K”, where K” is the DRS obtained from K via the operations (a)-(c), and place this term, with the subscript *ref*, in the store of the new representation;

2. enter the new type discourse referents E’, S_1 and Z into the new store;

3. enter the new ambiguity disjunction ‘$\lambda\alpha$-K” = E’ $\overset{!}{\vee}$ $\lambda\alpha$-K” = S_1 $\overset{!}{\vee}$ $\lambda\alpha$-K” = Z’ into the condition set of the new non-presuppositional DRS K’ (which is thus non-empty in this case).

For easier readability we drop the primes from ‘K” ’; so we will write ‘K’ where just now we have been writing ‘K” ’. (But keep in mind that ‘K’ now stands for the result of applying (a)–(c) to the non-presuppositional DRS of (7).)

When we apply this modified token-to-type shifting procedure to (7), we obtain the structure in (16).

predicates ‘event’ and ‘state’ is explicitly encoded, but not that of ‘event type’ and ‘state type’.

(16)

$$\left\langle \lambda\alpha.K_{ref}\ E'\ S_1\ Z\ y' \mid \left\langle \left\{ \boxed{\begin{array}{c}\text{event type (E')}\\ \text{state type (S}_1\text{)}\end{array}}, \boxed{\text{barrier type(Z)}} \right\}, \boxed{\begin{array}{c} \lambda\alpha.\mathrm{K} = \mathrm{E}' \overset{!}{\vee} \\ \lambda\alpha.\mathrm{K} = \mathrm{S}_1 \overset{!}{\vee} \\ \lambda\alpha.\mathrm{K} = \mathrm{Z}\end{array}} \right\rangle \right\rangle$$

where K is the DRS (17).

(17)

$$\begin{array}{|c|}\hline e'\ \ s_1\ \ z\ \ x'\ \ s_0 \\ \hline \mathrm{s}_0 \supset\!\subset \mathrm{e}' \\ \mathrm{s}_0\text{:accessible(y')} \\ \mathrm{s}_1 : \neg\ \text{accessible(y')} \\ \text{CAUSE(e, s}_1\text{)} \\ \text{agent(x')} \\ \text{Agent(e')} = \text{x'} \\ \text{embassy(y')} \\ \text{E'(e')}\ \ \mathrm{S}_1(\mathrm{s}_1)\ \ \text{Z(z)} \\ \hline \end{array}$$

3.2.5 Completing the DRS construction for (1)c

At last we return to the DRS construction for (1)c. (18) gives the representation for the T' node of the syntactic tree for this sentence.

(18)

$$\left\langle \mathrm{t}_{ref}\ \ \mathrm{e} \mid \left\langle \left\{ \boxed{\text{event(e)}}, \boxed{\begin{array}{c}\text{agent}(\underline{\text{x}}) \\ \vee\ \text{event}(\underline{\text{x}})\end{array}}, \boxed{\begin{array}{c}\text{event type}(\underline{\text{A}}) \vee \\ \text{state type}(\underline{\text{A}})\ )\end{array}} \right\}, \begin{array}{|c|}\hline \\ \hline \mathrm{t} < \mathrm{n}\ \ \mathrm{e} \subseteq \mathrm{t} \\ \text{e: verh'rn'}(\text{x},\underline{\text{A}}) \\ \hline \end{array} \right\rangle \right\rangle$$

Argument insertion now involves substitution of the referential argument term $\lambda\alpha$.K from (16) for the slot symbol $\underline{A}$ in (18). The result is given in (19).

(19)

$$\left\langle E'\ \mathrm{S}_1\ Z\ y'\ \mathrm{t}_{ref} \ \middle|\ \left\langle \left\{ \begin{array}{l} \boxed{\text{event-type(E')}}, \\ \boxed{\text{state-type(S}_1\text{)}}, \\ \boxed{\text{agent(}\underline{\text{x'}}\text{)}}, \\ \boxed{\text{barrier type(Z)}}, \\ \boxed{\text{event(e)}}, \\ \boxed{\begin{array}{c} \text{agent(}\underline{\text{x}}\text{)} \vee \\ \text{event(}\underline{\text{x}}\text{)} \end{array}}, \\ \boxed{\begin{array}{c} \text{event type(}\tau\text{)} \vee \\ \text{state type(}\tau\text{)} \end{array}} \end{array} \right\}, \boxed{\begin{array}{c} \mathrm{t} < \mathrm{n} \quad \mathrm{e} \subseteq \mathrm{t} \\ \text{e: verhindern'(}\underline{\text{x}},\tau\text{)} \\ \tau = \text{E'} \overset{!}{\vee} \\ \tau = \mathrm{S}_1 \vee \\ \tau = \mathrm{Z} \end{array}} \right\rangle \right\rangle$$

Here 'τ' is short for '$\lambda\alpha$.K', with 'K' asa in (17).

Consistency must now be checked between the different presuppositional constraints on τ that result when τ is identified with E', S_1 or Z, respectively. It should be intuitively clear that identification with E' and identification with S_1 both yield consistency and that identification with Z does not. Elimination of this last possibility means that the $\overset{!}{\vee}$-condition is now reduced to the binary disjunction τ = E' $\overset{!}{\vee}$ τ = S_1, which means that the sentence representation we end up with for (1)c is still ambiguous. To obtain this representation we once again take the presuppositions as satisfied and bind the discourse referents in the store existentially by transferring them to the universe of the DRS. The result is given in (20).

(20)

$$\begin{array}{|c|} \hline t\ \ e\ \ E'\ \ S_1\ \ Z\ \ y'\ \ x \\ \hline \mathrm{t} < \mathrm{n} \quad \mathrm{e} \subseteq \mathrm{t} \\ \text{e: verhindern'(x,}\lambda\alpha\text{.K')} \\ \text{event-type(E')}\ \ \text{state-type(S}_1\text{)} \\ \text{barrier type(Z)} \\ \text{agent(x)} \vee \text{event(x)} \\ \lambda\alpha\text{.K'} = \text{E'} \overset{!}{\vee} \\ \lambda\alpha\text{.K'} = \mathrm{S}_1 \\ \hline \end{array}$$

Here K' is just like the K of (17), except that the slot symbol $\underline{\text{x}}$ has been replaced by the discourse referent x. For a full appreciation of what (20) says it is important to keep the actual form and content of K' in mind.

We note for later use that the local ambiguity of (20) can also be 'multiplied out' into a $\overset{!}{\vee}$-disjunction of two complete representations, each representing one of the two remaining readings of *die Absperrung der Botschaft*. The first of these, representing the event reading, is obtained from (20) by:

1. replacing the second argument of ‘verhindern’ (viz. ‘$\lambda\alpha$.K) by ‘λe’.K”) where K” is like K except that the conditions ‘E’(e’)’, ‘S_1((s_1)’ and ‘Z(z)’ have been removed;

2. eliminating from the universe of (20) the discourse referents E’, S_1 and Z and from its condition set all conditions involving these discourse referents.

3. adding to the condition set of (20) the condition ‘E’(λe’.K”)’.

The second disjunct of the new representation is obtained in the same way, except that this time the argument term ‘$\lambda\alpha$.K’ is replaced by ‘λs_1.K” and the new condition is ‘state type(λs_1.K’). The resulting disjunction is given in (21).

(21)

$$
\begin{array}{|c|}
\hline
t\ \ e\ \ y'\ \ x \\
\hline
\mathrm{t} < \mathrm{n}\quad \mathrm{e} \subseteq \mathrm{t} \\
\mathrm{e{:}\ verhindern'(x,}\lambda\mathrm{e'.K'')} \\
\mathrm{agent(x)} \vee \mathrm{event(x)} \\
\mathrm{event\ type(}\lambda\mathrm{e'.K'')} \\
\hline
\end{array}
\quad \stackrel{!}{\vee} \quad
\begin{array}{|c|}
\hline
t\ \ e\ \ y'\ \ x \\
\hline
\mathrm{t} < \mathrm{n}\quad \mathrm{e} \subseteq \mathrm{t} \\
\mathrm{e{:}\ verhindern'(x,}\lambda \mathrm{s}_1\mathrm{.K'')} \\
\mathrm{agent(x)} \vee \mathrm{event(x)} \\
\mathrm{state\ type(}\lambda \mathrm{s}_1\mathrm{.K'')} \\
\hline
\end{array}
$$

We already noted that (1)c is formally speaking ambiguous according to the analysis which yields (20) or (21) as output, but that this ambiguity is of no consequence because the two alternatives of (20)/(21) assign equivalent truth conditions. The reason for this is that *verhindern* is true of a goal-directed action type iff it is true of the corresponding result state type. So, in particular, *verhindern* is true of the action type described by *die Absperrung der Botschaft* on its event reading iff it is true of the state type described by *die Absperrung der Botschaft* on its state reading.

3.2.6 unterbrechen

We now come to the verb *unterbrechen*. *unterbrechen* is of particular importance for us, since it is the verb that occurs in our illustration of how inferences at the CLP level can build on inferences at the DRS level. *unterbrechen* shares some of its salient features with *verhindern* and some others with *behindern*. In common with *behindern* are the selection restrictions for the theme argument: the theme can be an activity, a goal directed action or an agent. However, in the case of *unterbrechen* there is a notable lack of parity between these three categories. When you interrupt an action (such as fencing off the Embassy, writing a letter to the President etc) then what you interrupt is strictly speaking the activity that the agent is engaged in while trying to reach the goal of his action. Likewise, when you interrupt an agent, you interrupt an activity in which he is engaged. Here too, you might say, the activity is the ‘primary’ sort. One way to account for this asymmetry would be to adopt selection restrictions that only admit the category ’activity’ as theme of *unterbrechen*, while supplementing our theory of lexical and supra-lexical semantics with a coercion module. This module would let arguments of other categories in through the back door: for instance, the module would license an agent as theme of *unterbrechen* provided the context would allow one to see this agent as engaged in a certain activity, and to take this activity as the ’true’ theme of the verb, i.e. as the activity that is actually interrupted. Likewise, a goal-directed action can often be seen as determining an associated activity - the activity that is being performed in order to achieve the goal of the action – and can thus be justified as theme of *unterbrechen* because the predication expressed by verb and theme phrase can be understood as expressing interruption of the associated activity.

On closer reflection, however, the agent case and the action case do not really seem all that similar. Associating with an agent a particular activity that this agent is currently engaged in is a process that tends to be situation-dependent and ad hoc. In contrast, the relation between actions and the activities that agents engage in as part of performing those actions is a systematic one – one doesn't need to know much about a given situation to feel entitled to the assumption that an agent who is trying to perform a goal-directed action in that situation must be engaging in some sort of 'preparatory' activity that should eventually lead to the action's goal. These considerations have led us to the decision to treat actions, like activities, as a 'primary' theme category of *unterbrechen*. Thus in the lexical entry for *unterbrechen* we have adopted selection restrictions for the theme argument which admit both activities and goal-directed actions. On the other hand, cases in which the ostensive theme argument of *unterbrechen* is an agent seem the kind for which a coercion account is more appropriate.[30]

One feature that *unterbrechen* shares with *verhindern* is that it selects for theme arguments that are types rather than tokens. However, the reasons why this is so are different for the theme category 'activity' and the theme caegory'goal-directed action'. First consider the case of activities. Interruption of an activity is always interruption of something that is actually in progress – of an individual activity – and it carries an implication of resumption: the activity that was interrupted for some time is taken up again at the end of the interruption. But what is to count as a resumption? How are we to distinguish between a case where the initial activity can be said to have been *resumed* and one in which one would rather have to say that the agent engaged on a new activity some time after the first one was terminated, which may resemble the original activity in some ways but does not qualify as a resumption of it? That should depend, we think, on the way in which the activity is being *conceived*. If the original activity and the later one, which starts after the interruption, are both instances of this conception, then we have a case of resumption, otherwise not.

If this is the correct explication of what we understand by resumption, then an activity concept, or activity type, must be part of our understanding of the meaning of *unterbrechen*. Thus, if the theme of *unterbrechen* is an activity, then it must be available as an activity type and not just as an activity token.[31]

[30]We are convinced that coercion is a genuine aspect of natural language interpretation, and we are among those who believe that a coercion component will have to be introduced as part of any viable theory of lexical and supra-lexical semantics eventually. But to add such a component to the framework we are using in this paper would be a major undertaking and it wouldn't help us with the issues that this paper is about. In this case, it wouldn't make any difference to what can be said about the interpretation of (1)d. Adding a coercion component would make agents admissible as themes for *unterbrechen*. But since agents are not among the possible readings of *Absperrung*, whether agent themes are admitted or not is not going to make any difference to the disambiguation of *Absperrung* by its container verb *unterbrechen*.

[31]There is the further question whether in cases of resumption the activity before and the one after the interruption are the same *token* activity. (This would be an activity stretching from the start of the activity before the interruption all the way to the end of the activity after the interruption and which has the

When the theme is a goal-directed action, the considerations which indicate that it must be understood as a type are quite different. The argument for this case closely resembles the one we gave to establish that the themes of *verhindern* are types. As already observed in Section 2, when an action is said to be interrupted, then there is a kind of default implication that it will be resumed, and a further (if perhaps rather weak) implication that it will be completed after resumption. But we saw that the implication that the action will be completed is easily overwritten. So, while *unterbrechen* differs from *verhindern* in that interruption doesn't *entail* that the action won't be completed, non-completion is certainly a possibility: It is perfectly coherent, and even natural, to use *unterbrechen* in order to speak of a situation in which the action described by the theme argument phrase is *not* completed. And in such a situation there is, as we explained in our discussion of *verhindern*, no token action that instantiates the action type. In other words, predications involving *unterbrechen* and a goal-directed action as theme are compatible with there being no individual action that instantiates the theme description. Thus we find ourselves driven, just as we were in the case of *verhindern*, to the conclusion that action themes of *unterbrechen* cannot simply be tokens, but that they must be types.

When the theme of *unterbrechen* is an action type, then there must be an actual activity connected with that type that the interruption puts a (temporary) end to. To capture this aspect of the meaning of *unterbrechen* we assume that with each action type A is associated an activity type A' the tokens of which are activities that are performed as part of attempts to perform actions that instantiate A. We call this relation between A and A' 'PREP' (for 'Preparatory'); 'PREP(A',A)' means that A' is an acticvity type whose instances qualify as attempts to realise A – i. e. as activities that will grow into, or might have grown into, complete actions that are instances of A. We assume that it is a general property of goal-directed action types A that there is an activity type A' which stands to A in the relation PREP. So when *unterbrechen* is used with A as theme, there will be an actual instance a' of the associated activity type A' that is terminated by the described interruption.

As far as we can tell, the action and activity types that serve as themes to *unterbrechen* come with a requirement of unique satisfaction, just as we assumed in connection with *verhindern*. In particular we assume that when the theme is a goal-directed action type A and A' is the associated activity type, then A' also satisfies this uniqueness requirement.

As we argued in Section 2, the implication of a subsequent resumption that seems to be part of the meaning of *unterbrechen* is a defeasible implicature, and one that is defeated easily. Still, this implicature *is* part of the lexical meaning of *unterbrechen*, so it must be included in the semantics of its lexical entry. But because it is a defeasible part, it should not simply be amalgamated with the non-defeasible parts of the meaning of the word. We do this by marking the defeasible part of the meaning with the label 'DEFEAS' (for 'defeasible').

In the course of our discussion of *verhindern* we observed that the agent of the event e that *verhindern* describes cannot be at the same time the agent of what e prevents. *unterbrechen*, we said there, has the opposite property. Here the agent of the described event e must be identical with the agent of the interrupted action or activity.[32] Concomitant

interruption as a hole somewhere in its middle.) We do not know how to settle this question.

[32] Actually, put in terms as general as this the claim is not quite true. For there is one salient exception, where one speaker interrupts another. (*Entschuldige, dass ich Dich unterbreche, aber ..* ('Excuse me for

with this identity requirement are stricter selection restrictions on the subject and object of *unterbrechen*. The subject must be a true agent and the object can only be an activity or an action (i.e. an eventuality that has a true agent).

Whether the theme argument phrase of an occurrence of *unterbrechen* contains the description of an activity or a goal-directed action, the predication expressed by *unterbrechen* is essentially the same: in each case what gets interrupted is an activity, either the activity that the argument phrase describes directly or an activity that can be seen as 'preparatory' to an action of the kind the phrase describes. In principle these two possibilities could be captured by an entry that treats the theme argument as having to be an activity, together with a principle that allows for coercion of action descriptions into descriptions of the corresponding preparatory activities. Here we have opted instead for an entry in which the two possibilities are treated separately: the theme can be either an activity type or an action type, but with the qualification that in the latter case it is really an instance of the associated activity type that gets interrupted properly speaking.

The way in which we have represented these two options in our entry E4 below is somewhat informal. The selection restrictions for the theme argument A mentions two possibilities: A can be an activity type or a goal-directed action type, and for each of these possibilities the entry specifies a separate semantic representation. It should be clear how an entry of this form is used in sentence representation: depending on the theme argument phrase with which *unterbrechen* is being combined, the appropriate representation is to be inserted for the given occurrence of the verb. We defer the question whether a better notation could be found that captures what is common to the two options in a more elegant and insightful manner.

The time has come to present the entry we propose for *unterbrechen*.

(E4) unterbrechen

unterbrechen	verb	DP:nom	DP:acc
	e	x	A
	event(e)	agent(x)	g-dir.action type(A) $\vee$ activity type(A)

Semantic Representation:

---

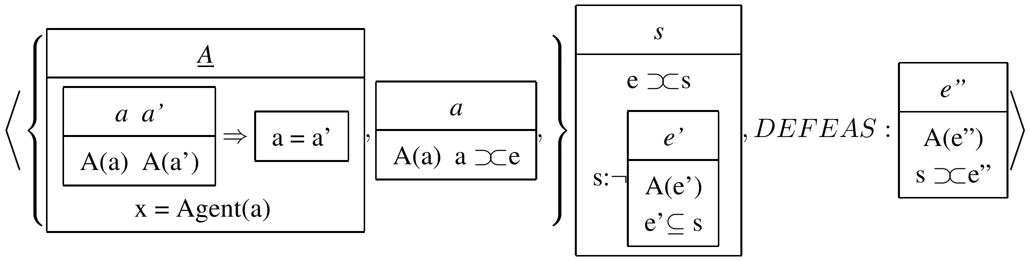
interrupting you, but ..'). As far as we can tell this possibility is specific to the use of *unterbrechen* in relation to conversation. It is not covered by ur entry for *unterbrechen*.

While we are at it, we might as well mention onoe further possibility that our entry for *unterbrechen* might be accused of having missed, and that could be relevant to the disambiguation potential of *unterbrechen* vis-a-vis *Absperrung*. It might be thought that *unterbrechen* not only admits the event reading of *Absperrung* but also its physical object reading. After all we can say things like (i) to mean that there were several breaches in the fence around the embassy (or several parts of the embassy's perimeter that the fence did not cover).

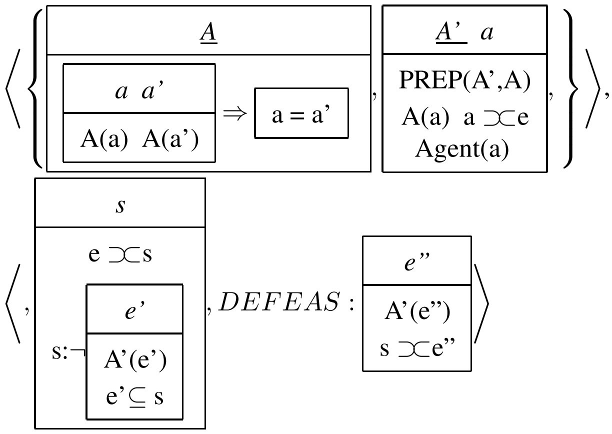
(i) Die Absperrung der Botschaft war an mehreren Stellen unterbrochen.
(The fencing of the embassy was interrupted in several places.)

However, we ourselves cannot get such a physical object reading for (1)d, on which the sentence would mean that people *made* openings in the fence. Exactly why there is this difference between *unterbrechen* and its past participle *unterbrochen* we do not know. But whatever the reason, we just do not think that the transitive verb *unterbrechen* can be used this way in the active voice. Hence we haaven't included the possibility in our lexical entry.

(i) activity type(A)

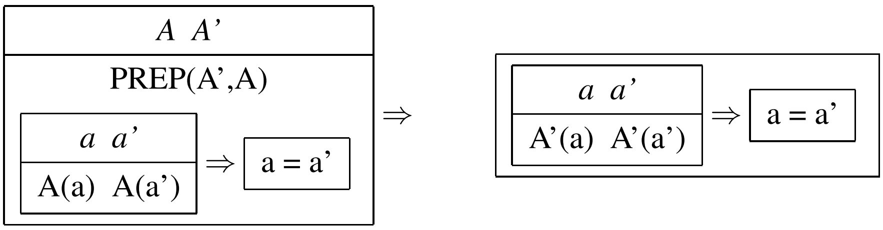


(ii) goal-directed action type(A)



We state the principle that when PREP(A',A), then unique instantiation of A implies as unique instantiation of A' in the form of a Meaning Postulate.

(22)



The construction of the representation for (1)d proceeds in much the same way as that for (1)c. The main difference is that this time the compatibility check will retain only the event reading of *Absperrung*. This means that the $\overset{!}{\vee}$-disjunction reduces to the single disjunct τ = E' where again τ is short for $\lambda\alpha$.K and $\lambda\alpha$.K is constructed as described in our discussion of (1)c. We can simplify the resulting interpretation by removing the occurrence of E' in the universe and by replacing all other occurrences of E' by τ. Moreover we can also remove, without substantial loss of information, all occurrences of 'S_1' and of 'Z' together with the conditions involving them ('$S_1(s_1)$', 'Z(z)','state type(S_1)' and 'barrier type(Z) ').

One difference with the representation for (1)c is that the new representation consists, like the component of the entry E4 from which it is in part derived, of a non-defeasible and a defeasible part. It is easy to see, however, that this does not change much. In particular, the token-to-type shifting of the representation of *die Absperrung der Botschaft* is in no way affected by it. The resulting representation is shown in (23).

(23)
$$\left\langle \begin{array}{|c|} \hline t\ e\ x\ A'\ a\ s \\ \hline \mathrm{t} < \mathrm{n}\quad \mathrm{e} \subseteq \mathrm{t} \\ \mathrm{Agent(e) = x} \\ \mathrm{PREP(A',}\ \tau) \\ \mathrm{A'(a)}\quad \mathrm{Agent(a) = x}\ \mathrm{a} \supset\!\subset \mathrm{e}\quad \mathrm{e} \subseteq \mathrm{s} \\ \mathrm{s}:\ \neg \begin{array}{|c|} \hline e' \\ \hline \mathrm{A'(e')} \\ \hline \end{array} \\ \hline \end{array}, DEFEAS \begin{array}{|c|} \hline e'' \\ \hline \mathrm{A'(e'')} \\ \mathrm{s} \supset\!\subset \mathrm{e''} \\ \hline \end{array} \right\rangle$$

For easy reference in the second half of the paper we summarise the salient properties of this representation and its potential for use in further processing.

1. The distinction between the non-defeasible DRS on the left and the defeasible DRS on the right (marked ‘DEFEASIBLE’) manifests itself when (1)d occurs embedded within a wider discourse context. If this discourse context contradicts the right hand DRS but not the left hand one, then the right hand one is dropped while the left hand one is retained. When there is no contradiction, then the right hand side DRS is retained as well. (That is, the marker ‘DEFEAS’ is dropped and the DRS it marked is merged with the non-defeasible DRS.) When the context contradicts the left hand DRS then the discourse is prima facie incoherent; if no context repair is possible either, it wll have to be dismissed as incoherent.

2. The condition ‘PREP(A’,τ)’ in the left hand DRS indicates that the subject phrase *die Absperrung der Botschaft* has been interpreted as denoting a goal-directed action (or, more accurately, since token-to-type shifting is involved, as denoting the type of a goal-directed action). This means that A’ now stands for the corresponding activity type, the type of an activity that is preparatory to the culmination of an action instantiating τ (i.e. of an action that results in the embassy being properly fenced off).

3.2.7 aufheben

The penultimate verb of those in (1) is *aufheben*. *aufheben* has a number of different readings. Two of these correlate with uses of the English verb *lift*: a ‘concrete’ meaning – that of lifting a physical object, in the sense of physically moving it to a higher position – and an ‘abstract’ meaning, in which the theme is not a physical object, but some kind of state, typically a state of prohibition or ban.[33] In addition, *aufheben* has a third meaning that has no obvious connection with its two ’lift’ meanings. This meaning is captured

[33]When used in this second way *aufheben* appears to be somewhat more liberal than *lift* in that it can be applied to laws - something like ’they lifted the law on taxation of luxury goods’ sounds awkward in English but its literal German translation is perfectly acceptable. These are subtleties that our analysis is not designed to capture.

approximately by the English verb 'keep', but only in its inchoative use, when a decsion is made to keep something, as when I say 'We keep that for later' to you who are intent on throwing it out right now. (*aufheben* cannot be used as a state verb.) Like the 'concrete lift' meaning, the 'keep' meaning selects for themes that are physical objects. (These may be of pretty much any kind, as long as they are the sorts of things that can be moved and put or stowed away.)[34]

This classification leads to an analysis for *aufheben* according to which it is 3-ways ambiguous, with the readings:

(i) 'concrete lift';

(ii) 'abstract lift';

(iii) 'keep'.

Of these, (i) and (iii) require the theme to be a physical object, while (ii) restricts it to states. (The states must be of a fairly specific kind, but the kind includes the states that can be described by *Absperrung*.) This means that between its different readings *aufheben* allows for two of the three possible interpretations of *Absperrung*, excluding only the event reading. In this respect the disambiguation of the selectional restrictions in (1)e resembles that in (1)c: Disambiguation of *Absperrung* is partial; of the three possible readings only one is eliminated while two remain. But apart from this similarity the disambiguation problems presented by the two sentences are very different. In (1)c, we saw, the remaining ambiguity of *Absperrung* is of no consequence, since irrespective of which of the two readings of *Absperrung* is chosen we obtain the same truth conditions for the sentence. With (1)e that is clearly not so. Taking *Absperrung* in its state sense in (1)e leads to a clearly different sentence interpretation from what we get when we interpret *Absperrung* as describing a physical object. Moreover, when this second interpretation is chosen for *Absperrung*, then we are facing a new ambiguity, between the 'lift' reading of *aufheben* and its 'keep' reading. So after the selection restrictions have done their work we are still stuck with a 3-way ambiguity, but now at the sentence level.

The case of (1)e is interesting in particular because it involves the confrontation of ambiguities that come from opposite sides - both the representation of the subject DP and that of the T' node are ambiguous. In all likelihood the way in which these ambiguities interlock in this case is just one of many different forms such interactions can take. In fact, the comparison with the only other case of this sort that we have encountered in this paper - that of combining the lexical semantics of *Absperrung* with the semantics of *die Botschaft* - is instructive. Both in that case and in the present one the main disambiguating effect is that of the container predicate on its argument phrase. In *Absperrung der Botschaft Absperrung* disambiguates *Botschaft* while its own ambiguity remains unaffected. (Except that the selected meaning of *Botschaft* confirms the entity reading of *Absperrung* as a genuine possibility; this, one might say, is resolving a 'second order ambiguity', between *Absperrung* as twofold and as threefold ambiguous.) The case of (1)e is similar in that here too it is the argument phrase *die Absperrung der Botschaft* that gets

[34] Apparently the historical connection between the third meaning of *aufheben* and its second meaning is that 'lifting', in the sense of physically taking away, (as in Engl, 'shoplifting') often leads to, and presumably is done for the purpose of, keeping what was taken (in a safe place). (Grimms Wörterbuch).

(partially) disambiguated, while the container predicate *aufheben* remains strictly speaking just as ambiguous as it was: each of the three readings we have distinguished remains an option.

However, strict reduction of ambiguity is only one aspect of ambiguity resolution. For even when *die Absperrung der Botschaft* doesn't reduce the ambiguity of *aufheben* in an absolute sense, surely the probabilities have shifted. True, you can lift and/or keep a fence or wall or barricade. But that is just one way in which a blockade can be lifted, and it is not the way that would most naturally come to mind - it is not a 'prototypical' way of lifting a blockade (although what is perceived as prototypical here may vary somewhat as a function of cultural context or individual experiences, in the way prototypicality often does). In any case, the reading of (1)e according to which its direct object is the embassy's inaccessibility is - we conjecture - for most people its default interpretation. (There probably is also some stacking of the deck when *Absperrung* is combined with *die Botschaft*, though as far as we can tell the effects are much less dramatic in that case.)

We suspect that the probabilistic effects of disambiguation are at least as important in the practice of actual language use as disambiguation of the strict, logically based kind that we are investigating in this paper. As things stand we have little idea of the methods that could be developed and used to deal with probabilistic disambiguation (in large part because this is not our area of expertise), but we see the exploration of such methods as both conceptually important and potentially of great value in NLP applications.[35]

The disambiguation examples discussed in this paper were chosen more or less at random.[36] If so much variation can be found in so few examples, collected in such a haphazard manner, surely the full range of ambiguity and disambiguation patterns must

[35]Weighting of 'soft disambiguation factors' – factors which alter the likelihoods of different readings of an expression without eliminating any one of them definitively – has been used in a project closely related to the one within which we have been doing the work reported here. (Both projects are part of the Sonderforschungsbereich 732 'Incremental Specification in Context' at the University of Stuttgart.) In this project – Disambiguierung von Nominalisierungen bei der Datenextraktion aus Korpora: Morphologisch verwandte Wörter' (Engl. 'Disambiguation of nominalisations as part of data extraction from corpora: morphologically related words') – has been investigating disambguation phenomena that are very similar to the ones we have been considering here. The disambiguation equally concerns *-ung* nouns that are sortally ambiguous. An example of the nouns studied is *Behauptung* ('assertion'), which, like its English counterpart, is ambiguous betweeen an event reading (the event of making an assertion) and a 'proposotiional object reading' (the assertion as entity, which continues to exist after the event has come to its end and is identified largely in terms of its content). The container predicate for nouns of this sort that has thus far been considered is the preposition *nach* rather than a verb. *nach* is itself ambiguous between a temporal reading ('after') and a 'propositional ' reading ('according to') and these two readings correlate in an obvious way with the event reading and the propositional object reading of nouns like *Behauptung* when they are the heads of noun phrases governed by *nach*. The project has looked at many more factors that affect the interpretaton of such *nach*-PPs than we have done here, but what we have just noted about the disambiguating influence of *Absperrung* on *aufheben* gives a inkling of the direction of some of those factors. See in particular Eberle et al. (2009a) and Eberle et al. (2009b).

[36]Our starting point for this investigation was the decision to look at a few cases in which *Absperrung* (as a salient three ways ambiguous *-ung* noun) can get wholly or partly disambiguated through the selectional restrictions contributed by a container verb. The verbs - those of the sentences in (1) - were chosen just for the different ways in which they do this, with no attention to their further properties. It was only when we set about working out the details of the semantics of the sentences in (1) that we became aware of the complexities that are connected with the verbs in these sentences. And the ambiguity of *Botschaft* was another twist whose interest we came to recognise only well after we had settled on *die Absperrung der Botschaft*.

be vastly richer. To find more of them we only have to keep looking and be prepared to dig deeper.

As far as (1)e is concerned, the details of the construction of its (multiply ambiguous) semantic representation will, just as in the cases we have already looked at in detail, depend to a large extent on the content and form of the lexical entries of its words, and the entry of *aufheben* is evidently a major player in this. Here is one proposal for the organisation of the information about *aufheben* that we have already discussed above. It involves a major division of the entry into the case were the theme is a state (or prohibition, regulation or law) on the one hand and the two cases in which the theme is a physical object on the other. A second division is then made between the last two cases - that where *aufheben* means 'lift' and the one where it means 'keep'. The first division is a matter of selection restrictions, the second one of semantic representation.

Characterising the actual semantic content of a word is always a delicate matter, to which we haven't devoted much commentary so far. In one case, *behindern*, we have simply used the verb itself (more precisely: a constant of our DRS language that is assumed to denote the same content and that we have conveniently given the same orthographic shape). In the other cases - *verhindern* and, especially, *unterbrechen* - we have made an effort to analyse at least some of the semantic structure of the denoted concept with the help of independently grounded terms. The entry for *aufheben* that is given below is a mixture in this respect. The 'lift a physical object' reading is not analysed at all; here the English verb 'lift' is used as a mnemonic for the intended concept; but that of course isn't what one could call semantic analysis. The other two readings, however, are so unequivocally inchoative that an analysis in terms of the result states brought about by the events they describe quasi imposes itself. When *aufheben* combines with an abstract theme (such as a prohibition) the result is that the theme is no longer in force. The conceptual analysis of 'being in force' belongs to the province of deontic logic and lies beyond the horizons of this paper. So we allow ourselves the use of the predicate 'in force' without attempts at further elucidation. The third reading, that according to which the decision is made to keep the theme, leads to a state of the agent 'keeping' the theme. We use 'keep' to characterise this state, but here too we are relying on an intuitive understanding – shared by competent speakers of English – of what concept is meant.

We conclude our discussion of *aufheben* with our proposal for its lexical entry. We leave it to whosoever (including ourselves) to use this entry in a detailed description of the semantic representation of (1)e.

(E5) **aufheben**

(i)

aufheben	verb	nom	acc
	e	x	y
	event(e)	agent(x)	prohibiton(y)
		∨ event(x)	∨ state(y)

SEMANT:

$$\begin{array}{|c|}\hline s \\ \hline \text{s: } \neg \text{ in-force(y)} \\ \text{CAUSE(e,s)} \\ \hline\end{array}$$

(ii)

aufheben	verb	nom	acc
	e	x	y
SEL.RES:	event(e)	agent(x)	physical obj(y)

SEMANT: $\boxed{\text{e: lift(x,y)}} \overset{!}{\vee} \begin{array}{|c|}\hline s \\ \hline \text{s: keep(x,y)} \\ \text{CAUSE(e,s)} \\ \hline \end{array}$

3.2.8 ignorieren

Our final verb, *ignorieren* (ignore, not pay attention to), has been included as a reminder that there are also verbs that do nothing towards resolving sortal ambiguities like those of *Absperrung*. As far as we can tell, *ignorieren* imposes no selection restrictions on its theme argument. (And even if there is something we have missed and its possible themes are restricted in some way, it is certain that those restrictions won't exclude any of the three readings of *Absperrung*.) In this regard, *ignorieren* is like other attitude verbs. Many attitude verbs, such as *denken an* (think of), and also 'referring verbs' such as *erwähnen* (mention), exemplify an apparently complete absence of categorical restrictions on their theme arguments. This shouldn't come as a surprise, for pretty much anything can be made into an object of thought, or into an object of mention. *ignorieren* belongs also to this group of verbs. What it means is roughly to the effect that the subject is not paying attention to something or other in the context of a certain train of thought or deliberation. Here too, the something or other that is being ignored can be pretty much anything.

Because of the negative character of the meaning of *ignorieren* – the *not* taking account of the theme in the context of a certain train of thought, deliberation or argumentation - coming up with a satisfactory representation of its meaning is a non-trivial challenge. (What needs to be represented is what it is for someone to not include an object of thought in a given thought or thought process.) The challenge is an interesting and important one - one of many that confront us when we extend the study of attitude verbs beyond the narrow horizon of the small collection - *know*, *believe*, *want* and a few more – to which semanticists have by and large restricted their attention for far too long. But this is not the place to go into such matters. For no matter how diligently we represent the lexical content of *ignorieren*, it won't alter the fact that in a sentence like (1)f *ignorieren* cannot make any contribution to the disambiguation of *Absperrung*.

No lexical entry, therefore, for *ignorieren*.

4 Event–Calculus and DRT

Before we start to develop integrity constraints and programs for the DRSs introduced so far we will give a short informal introduction to the event calculus. For a much more comprehensive introduction the reader is referred to van Lambalgen and Hamm (2005). The event calculus was originally developed on the basis of McCarthy's situation calculus (McCarthy and Hayes, 1969) by Kowalski and Sergot 1986 and Shanahan 1997 and used for high level control of mobile robots. The theoretical aim pursued with this calculus was the solution of the frame problem in Artificial Intelligence.

4.1 Linguistic Motivation

Consider the following short piece of discourse:

(24) It was hot. Jean took off his sweater.

We naturally understand that the eventuality expressed by the second sentence is included in the temporal profile of the eventuality expressed by the first sentence. In order to establish this temporal overlap one could intuitively argue as follows:

(25) World knowledge contains no link to the effect that taking off one's sweater changes the temperature. Since it is hot at some time before *now*, the state *hot* must either hold initially or must have been initiated at some time t. The latter requires an event, which is however not given by the discourse. Therefore *hot* holds initially. Similarly no terminating event is mentioned, so that *hot* extends indefinitely, and it follows that the event described by the second sentence must be positioned inside the temporal profile of *hot*.

The event calculus is meant to formalize this kind of argumentation. Note the following important feature of the above argument. Several steps use a non–monotonic inference scheme. For instance the conclusion that the state *hot* holds initially is derived from the observation that the discourse does not mention an initiating event. From this observation we conclude that there is no initiating event, leaving only the possibility that *hot* holds initially. A second feature of this reasoning involves the principle of *inertia*. If a state – *hot* in our example – is not forced to change under the impact of an event it is assumed to remain unchanged. This is the principle of inertia, which is axiomatized by the axioms of the event calculus.

This specific kind of non–monotonicity is intimately linked to the event calculus as a planning formalism. Planning is defined as setting a goal and devising a sequence of actions that will achieve that goal, taking into account events in the world, and properties of the world and the agents. Now given a goal G and circumstances C under which G can be achieved it does not follow in a strict sense that G can be achieved under C plus some additional circumstances D. In this sense a planning system requires a non-monotonic formalism.

A close connection between planning and linguistic processing is established by assuming that a sentence S is considered as a goal (make S true) to be achieved by updating the discourse model. This means that we can model understanding a sentence in a discourse as such a goal. The goal is to make a sentence – as part of a discourse – true by accommodating those facts necessary for establishing the truth of the sentence. This is one of the leading ideas of DRT. In example (24) the first sentence provides a discourse model which is updated to make the second sentence true unless it is forced to give up essential parts by explicit information incompatible with it.

We will now procceed to describe the event calculus a bit more formally. We start with the language of the event calculus.

4.2 The language of the event calculus

Formally the event calculus is a many-sorted first order logic. The sorts include event types, fluents (time-dependent properties, such as activities), real numbers, and individuals. We also allow terms for fluent-valued and event type-valued functions.

The event calculus was devised to formally model two notions of change, instantanous change – such as two balls colliding – and continuous change – for instance the acceleration of a body in a gravitational field. A first series of primitive predicates is used for modelling instantaneous change.

(26) *Initially*(f)

(27) *Happens*(e, t)

(28) *Initiates*(e, f, t)

(29) *Terminates*(e, f, t)

The intended meaning of these predicates is more or less self-explanatory. The predicate *Initially*(f) takes as its argument a fluent and says that f holds at the beginning of a scenario. *Happens*(e, t) holds if event type e happens at time point or interval t. The event calculus allows to interpret t as a point or as an interval. *Initiates*(e, f, t) says that event type e causes f to be true strictly after t; i.e. f does not hold at t. Finally *Terminates*(e, f, t) expresses that f holds at t and that e causes f not to hold after t.

The next two predicates are used to formalize continuous change.

(30) *Trajectory*(f_1, t, f_2, d)

(31) *Releases*(e, f, t)

The 4–place predicate *Trajectory*(f_1, t, f_2, d) measures the change of f_2 under the force f_1 in the interval from t to $t + d$. Linguistically it is very close to the notion of incremental theme (see for instance Krifka, 1989; Dowty, 1991). One may think of f_1 as an activity which acts on f_2. Dowty uses *mowing a lawn* in order to explicate the notion *incremental theme*. In Dowty's example f_1 is the mowing activity and f_2 the changing state of the lawn under this activity. The fluent f_2 should therefore be considered a parameterized partial object; in Dowty's example the state of the lawn after d time steps of the ongoing activity *mowing*. The axioms of the event calculus then provide the homomorphism between the ongoing activity and the resulting (partial) state – the partially mowed lawn – as required by Dowty.

The *Releases*(e, f, t) predicate is necessary for reconciling the two notions of change formalized by the event calculus. Without this predicate the axioms would immediately produce an inconsistency. Intuitively the $Releases$–predicate says that after event e happened f is no longer subject to the principle of inertia. This allows f to change continuously. Consider a scenario of filling a bucket with water. Event type *tap–on* releases the parametrized fluent $height(x)$ that measures the continously changing level of the water in the bucket from the principle of inertia.

The *Clipped*–predicate of the calculus expresses that an event either terminating fluent f or releasing this fluent from the principle of inertia occurred between times t_1 and t_2.

(32) *Clipped*(t_1, f, t_2)

The last predicate states that fluent f is true at time t.

(33) $HoldsAt(f, t)$

'*HoldsAt*' should be considered a truth predicate although the axioms of the event calculus don't contain the characteristic truth axiom, i.e.

$$HoldsAt(\overline{\phi}, t) \leftrightarrow \phi(t)$$

where $\overline{\phi}$ is a name for formula ϕ. More formal machinery is necessary to transform $HoldsAt$ into a truth predicate satiyfying the cahracteristic truth axiom. We will resume the discussion of this topic in section 4.5.

In the next section we will introduce the axioms of the event calculus in an informal way and motivate their use by way of the above reasoning example (25). Appendix I contains a complete (formal) list of the axioms.

4.3 Axiomatization

In this section we will show how the axioms of the event calculus constrain the meanings of the basic predicates and how they formalize the principle of inertia. Moreover we will illustrate how the concept of the completion of a program helps to implement the intutive idea that events that are not required to happen by a narrative are assumed not to occur. We will demonstrate that this strategy forces the reasoning to be non–monotonic. Let us start with an informal example.

(34) If a fluent f holds initially or has been initiated by some event occurring at time t and no event terminating f has occurred between t and $t' > t$, then f holds at t', (here $>$ indicates the temporal precedence relation).

It is clear that this axiom embodies a law of inertia since if no f-related event occurs then f will be true indefinitely. In the reasoning of example (25) this axiom was used for instance when we concluded from the fact that no terminating event for *hot* is mentioned that this state holds indefinitely with regard to the story told so far. But this was not the only reasoning principle we applied. From the fact that no terminating event was mentioned in the short discourse we conclude that none occurred. The axioms of the calculus per se don't allow such a conclusion. We want a strengthening of the assumptions in which only those events occur which are explicitly mentioned in the discourse. In this sense understanding discourses is closely linked to *closed world reasoning*[37]. There are many techniques for formalizing this kind of reasoning; one is circumscription (for a good overview see Lifschitz, 1994). In this paper, however, we use the notion of the *completion of a logic program*. The advantage of logic programming is that these techniques allow us to *compute* discourse models via fix point constructions.

Let us be slightly more formal. The informal principle (34) is given by the combination of the following two axioms. These are axioms 1 and 3 of the official axiomatization in Appendix I.

[37]A typical example of this kind of *closed world reasoning* is provided by (train) schedules. If the schedule mentiones the departure of a train from Stuttgart to Tübingen at 10.15 and the next at 11.01 one assumes that there will be no train leaving Stuttgart between 10.15 and 11.01.

1. *Initially*$(f) \rightarrow HoldsAt(f, 0)$

2. *Happens*$(e, t) \wedge$ *Initiates*$(e, f, t) \wedge t < t' \wedge \neg$*Clipped*$(t, f, t') \rightarrow$ *HoldsAt*(f, t')

The most important feature to notice here is that the head – the part to the right of the implication sign – consists of a simple atom, and the body – the part to the left of the implication sign – of a combination of formulas from two languages. The first language is the language of the event calculus and the second language is the first order language of the reals, i.e. of the structure $(\mathbb{R}, 0, 1, +, \cdot, <)$. This means that the axioms are clauses of a constraint logic program. The formulas of the second language, such as $t < t'$, are the constraints of the constraint logic program. They are used to compute the time profile of the predicates of the event calculus. All variables in the clauses of logic programs are supposed to be universally quantified.

The completion of a program is a strengthening of it which explicitly expresses that the predicates occurring in the program have extensions that are as small as possible. Before we apply the method of completion to the examples on which we focus in this paper, we indicate how it works at the hand of a very simple program taken from Nienhuys-Cheng and de Wolf (1997).

(35) a. *Prof(confucius)* (Confucius is a professor.)
b. *Prof(socrates)* (Sokrates is a professor.)
c. $\neg Prof(y) \rightarrow Student(y)$ (Every person who is not a professor is a student.)

The program involves two predicates, *professor* and *student*. The programming formalism is set up in such a way that it is only possible to make positive statements about the extensions of predicates. Thus (35) states about the predicate *professor* that *confucius* belongs to its extension (35a) and also that *sokrates* belongs to its extension (35b); and these are all the definite claims the program makes about the extension of this predicate. The completion of the program ought to make this intuition concrete by stating explicitly that the extension of *professor* consists just of these two individuals. We accomplish this by forming the disjunction of the formulas $x = confucius$ and $x = socrates$, where x is a new variable, which intuitively plays the role of an arbitrary member of the extension of *professor*, and making this disjunction into the antecedent of the following implication:

(36) $x = confucius \vee x = socrates \rightarrow Prof(x)$

In the next step we universally quantify over the variable x and strengthen the implication to a bi–implication. The result is:

$$\forall x(x = confucius \vee x = socrates \leftrightarrow Prof(x))$$

This formula now says that the set of professors just consists of Confucius and Socrates. Under the assumption that Confucius and Socrates are the only individuals in the model we get that the set of students is empty. But assume now that the language in which the program is formulated contains an additional individual constant $plato$ interpreted by an element of the universe of discourse. Assume further that $socrates \neq confucius \neq$

$plato$[38]. Then (36) implies that $plato$ is not a professor. Now consider the third clause of program (35). A similar procedure applied to this clause yields:

(37) $\forall x(Student(x) \leftrightarrow \neg Prof(x))$[39]

Formula (37) implies that Plato is a student. The conjunction of (36) and (37) is the completion of program (35). This completion implies that Confucius and Socrates are the only professors and that Plato is a student. The program itself does not support such strong conclusions. A similar observation applies to certain extensions of (35) that bring additional entities into play. For instance suppose that we add to (35) the fact *beard(plato)*, which states that Plato has a beard. A minimal model for the completion of the extended program will have the universe $\{confucius, socrates, plato\}$. In this model Plato is not a professor, but the only student and the only one with a beard.

Let us now give a simple example with events. Consider a description of a situation where the light is switched on at 1 in the night and switched off at 7 in the morning and given by the following program:

(38) a. $Happens(switch\text{-}on, 1)$
b. $Happens(switch\text{-}off, 7)$

The uncompleted program does not yet imply that the light wasn't switched off at 2 in the night and switched on at 3 in the night and so on. However, these events should not occur in the minimal model of program (38). The completion of the program is given by

$$\forall e(Happens(e,t) \leftrightarrow (e = switch\text{-}on \wedge t = 1) \vee (e = switch\text{-}off \wedge t = 7))$$

This formula means the same as:

$$\forall e(Happens(e,t) \leftrightarrow (Happens(switch\text{-}on, 1) \vee (Happens(switch\text{-}off, 7))$$

Any intervening events are thereby excluded.

This illustrates how the concept of the completion of a program helps to implement the intuitive idea that events that are not required to happen by a narrative are assumed not to occur. Note that this strategy forces the reasoning to be non–monotonic. We could easily enrich program (38) with clauses *Happens(switch-off, 2)* and *Happens(switch-on, 3)*. From the modified program the conclusion that there are no events happening between *Happens(switch-on, 1)* and *Happens(switch-off, 7)* are now no longer derivable.

To sum up: Understanding a sentence in a discourse is like computing a minimal model of the discourse in which the sentence is true. This computation is based on the completion of a constraint logic program for the discourse under discussion. In the next section we will see, however, that this aim cannot be achieved by the technical means inroduced so far.

[38]This is an instance of the uniqueness of names assumption.

[39]This is technically not quite correct. The formula produced by the official algorithm for computing the completion of a program is:

$$\forall x(Student(x) \leftrightarrow \exists y(x = y \wedge \neg Prof(y)))$$

But for the simple example discussed above this difference does not matter. The official formula and (37) are equivalent.

4.4 Integrity Constraints

As pointed out above, the variables in the clauses of logic programs are universally quantified. Therefore logic programs are restricted to provide universal information only. This is clearly not sufficient for our purpose. For example tense requires existential information (see the example below) and DRSs in general introduce existential information. We will use here a device from database theory – integrity constraints – to obtain the required additional information. In database theory integrity constraints are means to ensure that a database stays consistent under updates. In this paper we will use integrity constraints in a slightly different way; we employ them as means to update a discourse model. Let us explain this idea with a simple example, of an English sentence in the perfect.

(39) I have caught the flu.

This sentence says that I have the flu now and world knowledge tells us that there was an infection event in the past. Let flu be the fluent corresponding to *having the flu* and let e be the infection event. Our knowledge is thus formalized by the following program clause.

$$Initiates(e, flu, t)$$

As already said we view a sentence S as a goal (make S true) to be achieved by updating the discourse model. In general it isn't possible, however, to simply add this information to the discourse model without further ado. There are two reasons for this. First, we would like the updated discourse model to include explicitly all the events that must have occurred in order for the total information represented by it to be true. And, second, when spelling out what that comes to reveals a conflict, then that should mean that the new sentence cannot make a coherent contribution to the discourse as the starting model represents it. It is therefore important that we do not just add the condition that I have the flu now, but also the event that must have led to this state of affairs. The formalisation of the event calculus given earlier offers a systematic way of doing this. In the present instance what needs to be inferred from $HoldsAt(flu, now)$ is that there was an earlier event e initiating flu, something that is expressed in the present formalism by the clauses $Initiates(e, flu, , t)$, $Happens(e, t)$ and $t < now$.

We will now show how this reasoning applies to example (39). For this purpose assume that a discourse model is given as a collection of facts concerning events and fluents and assume that sentence (39) is formalized as $HoldsAt(flu, now)$. We do not take this formula as a program clause but as an instruction to construct a minimal adaptation of the discourse model in which $HoldsAt(flu, now)$ is true. In order to detect the events that must have occurred for $HoldsAt(flu, now)$ to be true we apply abductive reasoning using the basic program constituted by the axioms of our formulation of the event calculus, as well as, possibly, additional axioms that capture aspects of world knowledge. To this end we use $HoldsAt(flu, now)$ as the trigger that sets this reasoning process in motion. Informally the reasoning is as follows. We know that fluent flu is initiated by some event e. No terminating event has been mentioned. Therefore we conclude by closed world reasoning that no such event occurred. Consider again axiom (34) repeated here as (40).

(40) If a fluent holds initially or has been initiated by some event occurring at time t and no event terminating f has occurred between t and $t' > t$, then f holds at t'.

According to this axiom, to establish the truth of $HoldsAt(f, now)$ there is only one fact missing. We have to add $Happens(e, t), t < now$ and its logical consequences to the discourse model. This is sufficient to guarantee the truth of $HoldsAt(flu, now)$[40].

Let us now be a little bit more formal and see how this update is steered by the proof system of logic programming, which is called *resolution*. Resolution can be regarded as a species of abductive reasoning in which a premise is matched with the heads of all clauses with which it can be matched and the abductive inference is then drawn that the matching instantiation of at least one of the bodies of those clauses must hold. Note the obvious connection between this type of inference and the concept of program completion. We start with the query $?HoldsAt(flu, now)$. Applying Axiom 3 – repeated below as (41) – the query reduces to the new query

$$?\mathit{Initiates}(e, flu, t)$$

$$\neg \mathit{Clipped}(t, flu, t')$$

$$\mathit{Happens}(e, t), t < now$$

(41) $\quad$ ***Happens***$(e, t) \wedge$ ***Initiates***$(e, f, t) \wedge\ t < t' \ \wedge \neg$***Clipped***$(t, f, t') \rightarrow$ ***HoldsAt***(f, t')

The first clause can be resolved, since ***Initiates***(e, flu, t) is given. For the second query we have to use a form of resolution for negated queries. This means that we set up a new derivation with the positive query

$$?\ \textbf{\textit{Clipped}}(t, flu, t').$$

Since we have no matching clauses this query fails and therefore the negated query succeeds (This is the proof–theoretic version of negation as failure.). We are left with the last query

$$?\mathit{Happens}(e, t), t < now.$$

Since we don't have a matching clause for this query $?HoldsAt(flu, now)$, interpreted as query, would fail (finitely). However, $HoldsAt(flu, now)$ interpreted as an integrity constraint leads to an update of the discourse model with the missing clause. In this updated model $HoldsAt(flu, now)$ is clearly satisfied. This integrity constraint is written as

$$?HoldsAt(flu, now),\ succeeds$$

[40]There is a subtle difference between (39) and sentence (i)

(i) $\quad$ I have the flu.

Given general world knowledge, these two statments can be said to convey the same information: Anyone who has the flu must have caught it at some earlier ime. But in (39) the occurrence of such an event e is an inalienable part of the content, whereas (i) entails it only in conjunction with the relevant piece of world knowledge. When our CLP version of the event calculus is combined with DRT, this difference manifests itself in that the DRS constructed from (39) will contain the flu–catching event already. So if the integrity constraints contributed by (39) are derived from the DRS, the abductive reasoning we are discussing isn't needed. More precisely, it will lead to constraints that are already in the given constraint set. This is different for (i). The DRS for (i) only contains a representation of the current state of the speaker. So in this case the abductive process reveals a constraint that isn't present yet, and which therefore has to be added to the discourse context with the condition $HoldsAt(flu, now)$.

A more general description of this procedure is as follows: Given a program P containing the clauses below and an integrity constraint q we want to conclude that q can only be the case because one of the $\phi_i s$ is the case.

$$\begin{array}{c} \phi_1 \rightarrow q \\ \phi_2 \rightarrow q \\ \vdots \\ \phi_n \rightarrow q \end{array}$$

This is a strengthened form of closed world reasoning.

A second type of integrity constraint occurs when the top query must fail. This is important for sentences about the past.

(42) Max arrived.

This sentence tells us that Max's arrival was situated entirely in the past, and thus is not going on any more at the present. The positive query

$$?Happens(e,t), t < now$$

expresses just the first part. The second part can only be expressed by the negative constraint, which can be represented as

$$?Happens(e, now), fails$$

Since the resolution process also accepts queries beginning with a negation we can reduce this negative query to the positive query

$$\neg Happens(e, now)$$

Since both positive and negative constraints are admitted and the latter are identified by the term *fails*, it is natural to introduce a similar term to flag the positive queries. We use *succeeds*. So the constraints contributed by (42) can be given as

$$?Happens(e,t), t < \ now,\ \neg Happens(e, now)\ succeeds$$

We will say that an integrity constraint IC is *satisfiable* if it can be made to succeed in case it is positive, and can be made to fail in case it is negative.

4.5 Reification

DRSs will in general contain not only (discourse referents for) events, but also for states. The version of CLP we have presented so far differs in that it has variables for events but not for states. This gap can be filled by expanding our version of CLP with a *reification* component. This component makes it possible to associate a 'res' with each condition. In particular, it will enable us to associate with each condition of the form $HoldsAt(f, t)$ an entity that can be regarded as the state of the fluent f obtaining[41]. The reification procedure is based on a method due to S. Feferman.

[41] Reification can be put to many other uses as well, but this is the one for which we need it here.

We will explain briefly how this works. For this purpose we will enrich the event calculus with a specialization of the theory of truth and abstraction in Feferman (1984)[42].

Consider the predicate $burn(x, y, t)$ where t is a parameter for time. Feferman's system allows to form terms from this predicate in two different ways. The first possiblity is to existentially bind t and construct the term $\exists t.burn[x, y, t]$. The square brackets are used here as a notational device to indicate that $\exists t.burn[x, y, t]$ is a term and not a predicate any more. The second possibility is to abstract over the temporal parameter and form the term $burn[x, y, \hat{t}]$. Informally $burn[x, y, \hat{t}]$ should be understood as the set of times at which $burn(x, y, t)$ is true. But note that $burn[x, y, \hat{t}]$ is a term and therefore denotes an object. Feferman's system thus provides two different kinds of structured abstract objects. Intuitively we want to think of $\exists t.burn[x, y, t]$ as the event type corresponding to *x's burning of* y and of $burn[x, y, \hat{t}]$ as the fluent or state corresponding to *x's burning* y[43]. However nothing in the formal set up so far tells us that $\exists t.burn[x, y, t]$ is an event type and $burn[x, y, \hat{t}]$ is a fluent. In order to make sure that $burn[x, y, \hat{t}]$ behaves as a fluent $HoldsAt$ has to be turned into a real truth predicate. The following theorem from Feferman (1984) provides the necessary technical result.

Theorem 1 *Any system that is consistent – in the sense that it has a model – can be extended to a system with truth axioms*[44]*. The extension is conservative over the original system.*

For the special theory under discussion here we need just one truth axiom, which reads as follows:

$$HoldsAt(\phi[\hat{t}], s) \leftrightarrow \phi(s)$$

The specialization for $burn[x, y, \hat{t}]$ therefore is:

$$HoldsAt(burn[x, y, \hat{t}], s) \leftrightarrow burn(x, y, s)$$

This shows that $burn[x, y, \hat{t}]$ behaves like a fluent. Moreover, $\exists t.burn[x, y, t]$ cannot be substituted as an argument of the *HoldsAt*–predicate, but it can be substituted as an argument of the *Happens*–predicate. Hence, with regard to the axioms of the event calculus, abstract terms like $\exists t.burn[x, y, t]$ function as event types and terms like $burn[x, y, \hat{t}]$ as fluents.

To see what this process of reification adds to the representations developed so far consider again sentence (39), here repeated as (43).

(43) I have caught the flu.

The structure of this sentence was represented by the simple fluent flu in the derivation of Section 4.4. For the purposes of this section this representation was sufficient. However, we would like to have access to the internal structure of sentence (43) as well. For simplicity we will assume that the personal pronoun *I* is represented by the individual

[42]For the most recent version of this theory see Feferman (2008).

[43]For an analysis of these different types of English gerunds see van Lambalgen and Hamm (2005), chapter 12.

[44]A model for the event calculus was constructed in van Lambalgen and Hamm (2005).

constant *i*. Under this assumption sentence (43) can be formalized as the structured fluent $flu[i,\hat{t}]$. This representation allows us to have access to the subject of the sentence. We will see in a moment that the possibility to structure fluent and event type objects is an indispensible prerequisite for the transformation of DRSs to integrity constraints.

5 Programs and Integrity Constraints for DRSs

5.1 An Integrity Constraint for a Simple DRS

In this section we will outline the connection between DRT and EC with the simplest example from Hamm et al. (2006). Consider again sentence (44).

(44) Max arrived.

The DRS for this sentence is:

$$\boxed{\begin{array}{ccc} m & t & e \\ Max(m) & t < n & e \subseteq t \\ & e : arrive(m) & \end{array}}$$

Since DRSes introduce existential presuppositions which have to be accommodated integrity constraints are the appropriate means to represent their inferential potential. First we assume that predicates $max(x,t)$ and $arrive(x,t)$ are given. These predicates will be used in their reified forms. The first possibility for reification derives the fluent term $max[x,\hat{s}]$ and the second the event type $\exists s.arrive[x,s]$.

It has often been observed that the simple past uttered out of the blue is infelicitous. This tense requires that the context provides additional information such somethin like a ‘reference time’. We will represent the context here with a new fluent constant f and the clause $HoldsAt(\mathrm{f},\mathrm{t})$. This constant can then be unified with further contextually given information.

The discourse referent m and the condition $Max(m)$ corresponds to $HoldsAt(max[\mathrm{x},\hat{\mathrm{s}}],t)$; discourse referent e to $\exists s.arrive[x,s]$, which introduces the clause $Happens(\exists \mathrm{s}.arrive[\mathrm{x},\mathrm{s}],t)$; n is set to now and t correspond to the context fluent f. In this way the DRS for sentence (44) is turned into integrity constraint (45).

(45) $?HoldsAt(\mathrm{f},\mathrm{t}), HoldsAt(max[\mathrm{x},\hat{\mathrm{s}}],\mathrm{t}), Happens(\exists \mathrm{s}.arrive[\mathrm{x},\mathrm{s}],\mathrm{t}),\ \mathrm{t} < now,$ $\neg Happens(\exists \mathrm{s}.arrive[\mathrm{x},\mathrm{s}], now)$, *succeeds*

Since in the rest of this paper we will not be concerned with tense, we will simplify integrity constraints as much as possible. First we will drop the clause for the context fluent and the negative integrity constraint. Moreover we will skip over the internal structure of fluent and events whenever this does not lead to confusion. For instance we will simply write f for $max[\mathrm{x},\hat{\mathrm{s}}]$ and e for $\exists s.arrive[x,s]$. Given these assumptions integrity constraint (45) now reads:

(46) $?HoldsAt(\mathrm{f},\mathrm{t}), Happens(\mathrm{e},\mathrm{t}), \mathrm{t} < now$, *succeeds*

This is certainly not completely adequate but the topics to be discussed in the rest of this paper will not be affected by this simplification.

5.2 Scenarios and Hierarchical Planning

In this section we will start our discussion of more complex examples. The first one is the verb *absperren*. So far nothing was said about Aktionsart. According to van Lambalgen and Hamm (2005) every Aktionsart determines a specific 'scenario'. A scenario should be considered as a local program in contrast to the global program given by the axioms of the event calculus. These local programs provide the additonal information for the Aktionsarten in question, in this case the information specific to accomplishments. In order to formulate this local program we need the following terms in the language of the event calculus.

- $construct$ is an activity fluent.
- $barrier(x)$ is a parameterized fluent indicating the construction state x of the barrier.
- m a real constant indicating the construction stage at which the barrier is considered finished. Thus $barrier(m)$ may be considered the completed object.
- 0 is a real constant indicating the state at which the construction of the barrier starts.
- $start$ is an event initiating constructing.
- $finish$ is the event terminating the constructing activity when the barrier is finished.
- a fluent $accessible(b)$ represententing the state in which the embassy is accessible, where b is a constant denoting the embassy.
- g is a function relating the constructing activity to the construction stage of the barrier. To keep things simple we assume that g is monotone increasing.

These terms allow us to write the following set of clauses as one possible scenario for the accomplishment verb *absperren*.

(47)
a. $Initially(barrier(0))$
b. $Initially(accessible(b))$
c. $HoldsAt(barrier(m), t) \wedge HoldsAt(construct, t) \rightarrow$ $Happens(finish, t)$
d. $Initiates(start, construct, t)$
e. $Initiates(finish, barrier(m), t)$
f. $Terminates(finish, accessible(b), t)$
g. $Terminates(finish, construct, t)$
h. $HoldsAt(barrier(x), t) \rightarrow$ $Trajectory(construct, t, barrier(x + g(d)), d)$
i. $Releases(start, barrier(0), t)$

The scenarios for the Aktionsarten are not determined uniquely, but every scenario is required to include information specific to the Aktionsart of the verb under consideration. For the example above this means that every scenario has to include clauses about the starting and finishing events, about the activity *constructing*, the state *accessible(b)*, and clauses relating this activity to the state of the partial object $barrier(x)$. Together with the axioms of the event calculus these clauses determine inferences triggered by the Aktionsart of *absperren* and the lexical content of this verb.

We are primarily interested in the noun phrase *Absperrung der Botschaft*. We will first concentrate on the event reading; the result state reading will be discussed later.

The first step consists in establishing an event type corresponding to the event reading of *Absperrung der Botschaft*. Using Feferman coding we can transform the predicate $absperren(x, b, t)$ into the abstract event type $a = \exists t.absperr[x, b, t]$. Here b is again an individual constant representing the embassy. This is a possible denotation for *Absperrung der Botschaft*, but so far this event type is not related to the verb from which *Absperrung* is derived.

In order to link the nominal to the semantics of the base verb given by its scenario we introduce an *event definition* by hierarchical planning. The intuitive idea is that hierarchical planning allows to abstract from certain details of the verb's eventuality while maintaining the most important features of the verb's time profile. Formally hierarchical planning is given by program clauses defining an event occurring in the head atom of a clause. The following definition is from van Lambalgen and Hamm (2005):

Definition 1 *Suppose a scenario for the fluent f is given. In the context of this scenario, the event e is interpreted using f by* hierarchical planning *if* $Happens(start_f, s) \wedge s < r < t \wedge HoldsAt(f, r) \wedge Happens(finish_f, t) \rightarrow Happens(e, r)$

For present purposes, however, this definition lacks the needed flexibility. We see this when we try to make use of it in the interpretation of a sentence like (1c) which we repeat here as (48).

(48) Die Absperrung der Botschaft wurde verhindert.
The blocking-off of-the embassy was prevented.

When we use Definition 1 in its interpretation we get a contradiction. The details are as follows. The purport of Definition 1 is that it applies generally to uses of accomplishment verbs in the past tense: If e is the event described by such a verb on a given occasion, then e satisfies that instantiation of the programming clause schema presented in Definition 1 which we obtain when the (schematic) constants of the schema are replaced by the corresponding constants of the scenario that defines the lexical meaning of the verb (in the way in which (47) specifies the lexical meaning of *absperren*). Let us suppose, moreover, that the event reading of an *ung*–noun like *Absperrung*, which is derived from an accomplishment verb, inherits both the semantics given by the scenario for the underlying verb and the relevant instantiation of the clause schema in Definition 1 which reflects that the verb is an accomplishment. Then the trouble that we get into with (48) is this. The integrity constraint to which (48) gives rise concern on the one hand the denotation of the phrase *die Absperrung der Botschaft* and on the other the predication that involves both this phrase and the verb *verhindern* of which it is the theme argument. In order to calculate the minimal model of the underlying program (consisting of the Event Calculus together with the lexical entries for thw words occurring in the sentence) that satisfies the integrity constraints, the constraints are made into queries that trigger the kind of abductive reasoning that will eventually lead to the entities that the minimal model must contain. When we apply this procedure in the given situation to the integrity constraint concerning *die Absperrung der Botschaft*, then only the head of the instantiation of the clause of Definition 1 will match the conjunct of the integrity constraint which says that the event $\exists t.absperr[x, b, t]$ happens at some time s. This will lead to the conclusion that the event was finished, which directly contradicts the contribution that is made by the integrity constraint that reflects the predication involving *verhindern*.

We therefore choose the following weaker definition (49) for the event type representing the *ung*–nominal *Absperrung der Botschaft*. This definition allows for instantiations that do not involve attainment of the goal.

(49) $Happens(start_{construct}, s) \wedge s \leq r \leq t \wedge HoldsAt(construct, r) \rightarrow$
$Happens(\exists t.absperr[x, b, t], r)$

From now on we will simply write a for the defined event type $\exists t.absperr[x, b, t]$. Given this event type we can now write integrity constraints for the DRSs constructed so far showing how the *ung*–nominal *Absperrung der Botschaft* behaves as argument of verbs with differing semantic properties. We will discuss the verbs *behindern*, *verhindern*, *abbrecchen*, *unterbrechen* and finally *aufheben*.

Let us assume that an event type valued function $behindern$ is given. Then we arrive at the following integrity constraint[45]:

$$? - Happens(a, t), Happens(behindern(a), t), t < now\ succeeds$$

This is certainly too simple. An event type like *behindern* requires its own scenario. At the moment we are not able to write a precise scenario for the verb *behindern*. We think that for *behindern* to be applied successfully the cordoning-off activity must have been initiated and *behindern* supplies the additional information that this activity does not proceed in a smooth way. However, we think that although the cordoning-off activity is hampered in more or less serious ways, nevertheless the goal – the sealing off of the embassy – will eventually be achieved (non-monotonically).

This changes when one considers our next verb, *verhindern*. Data from anaphora resolution suggest that the result state is not initiated in this case. Consider the examples in (50).

(50) a. Die Absperrung des Rathauses wurde vorgestern von Demonstranten behindert. Wegen anhaltender Unruhen wird **sie** auch heute aufrecht erhalten.

The cordoning-off of the town hall was disturbed by protesters the day before yesterday. Due to continuing unrest, it is maintained today as well.

b. Die Absperrung des Rathauses wurde vorgestern von Demonstranten verhindert. *Wegen anhaltender Unruhen wird **sie** auch heute aufrecht erhalten.

The cordoning-off of the town hall was prevented by protesters the day before yesterday. Due to continuing unrest, it is maintained today as well.

Clearly in (50a) the pronoun *sie* in the second sentence refers to the target state of being cordoned off introduced by the first sentence. The impossibility of such an interpretation –

[45]This is a simplification: The scenario for *behindern* plus hierarchical planning triggered by past tense introduces an event type e which has to be unified with a.

this is what * is meant to signal – suggests that due to the meaning of the verb *verhindern* such a target state is not available in (50b).

An integrity constraint like

$$? - Happens(a, t), Happens(verhinder(a), t), t < now\ succeeds$$

is therefore inadequate, since it does not imply that the barrier was not completed in the minimal model.

From integrity constraint (51) however, we can deduce that in the case of *verhindern*, in contrast to *behindern*, $barrier(m)$ – the completed barrier – does not come into existence and that the embassy is still accessible, since the constraint requires that the canonical event *finish* from the accomplishment scenario doesn't happen and therefore $barrier(m)$ is not initiated. From the axioms of the event calculus and closed world reasoning we derive $\neg HoldsAt(barrier(m), now)$. Similarily it follows that the embassy is still accessible.

(51)

$$? - Happens(a, t), Happens(finish, t), t < now\ fails$$

5.3 Programs and Integrity Constraints for DRSs: *unterbrechen*

Before we discuss the rather complicated example *unterbrechen* let us first consider the somewhat simpler verb *abbrechen* in example (52).

(52) Die Absperrung der Botschaft wurde abgebrochen.
The blocking-off of-the embassy was stopped.

Since *abbrechen* in contrast to *verhindern* presupposes that the cordoning-off activity has been started and was going on for some time before this activity was stopped we have to write a program clause which captures this additional information. Clause (53) defines these semantic features of the verb *abbrechen*.

(53) $Happens(start, s) \wedge Initiates(start, f, s) \wedge s < r \leq t \wedge$
$\neg Happens(finish, r) \wedge Happens(e', t) \wedge Terminates(e', f, t) \rightarrow$
$Happens(abbrechen(e), t)$.

Integrity constraint (54) now computes a minimal update of the given discourse model in which all the information required by the antecedent of clause (53) is satisfied.

(54)

$$? - Happens(a, t), Happens(abbrechen(a), t), t < now\ succeeds$$

We will now turn to the more complicated verb *unterbrechen*. As in the case of *abbrechen* we have to write a program which specifies the meaning of this verb and then use this clause to write an integrity constraint for (55).

(55) Die Absperrung der Botschaft wurde unterbrochen.
The blocking-off of-the embassy was interrupted.

A first candidate which comes to mind is:

(56) $Happens(start, s) \wedge Initiates(start, f, s) \wedge s < r \leq t \wedge$
$\neg Happens(finish, r) \wedge Happens(e', t) \wedge Terminates(e', f, t)$
$\wedge\, t < t' \wedge Happens(e'', t') \wedge Initiates(e'', f, t')$
$\rightarrow Happens(unterbrechen(e), t')$.

Informally the time profile this clause defines may be paraphrased as follows: The *start*–event initiates at some time an activitiy fluent – in our case the *cordoning–off* activity – which is stopped after a certain amount of time. Again after an indefinite amount of time a further event e'' happens which causes the stopped activity to be resumed again.

With this intuitive interpretation in mind we can now write integrity constraint (57) for (55).

(57)

$$?-Happens(a, t), Happens(unterbrechen(a), t), t < now\ succeeds$$[46]

But now we run into a problem. To illustrate the problem recall sentence (58). As noted in Section 3.2.6 that there is nothing odd with this sentence and it receives a natural interpretation, which says that the blocking-off of the embassy was stopped for good.

(58) Die Absperrung der Botschaft wurde unterbrochen und nie wieder aufgenommen.
The blocking-off of-the embassy was interrupted and never taken up again.

However, the update of the given discourse model integrity constraint (57) computes will be empty for this sentence; i.e. integrity constraint (57) together with sentence (58) produces an inconsistency. The reason for this is simple. The antecedent of clause (56) forces a stopped activity to be resumed again by an event e'', but sentence (58) denies exactly this possibility. A way out is to weaken the clause defining the verb *unterbrechen*. As one possibility consider clause (59)

(59) $Happens(abbrechen(e), t') \vee$
$Happens(start, s) \wedge Initiates(start, f, s) \wedge s < r \leq t \wedge$
$\neg Happens(finish, r) \wedge Happens(e', t) \wedge Terminates(e', f, t)$
$\wedge\, t < t' \wedge Happens(e'', t') \wedge Initiates(e'', f, t')$
$\rightarrow Happens(unterbrechen(e), t')$

In order to facilitate the reading of formula (59) let us write $unterbrechen_{strong}$ for the antecedent of clause (56). Then we get the following simplified clause for *unterbrechen*.

(60) $Happens(abbrechen(e), t) \vee Happens(unterbrechen_{strong}(e), t) \rightarrow$
$Happens(unterbrechen(e), t)$

[46] From this constraint it does neither follow monotonically that the construction of the barrier was completed nor that it wasn't. However, non–monotonically, i.e. in the minimal model, we get the result that the construction process was successfully finished. The proof of this result follows the argumentation on pages 62ff of van Lambalgen and Hamm (2005). We also get the result that the embassy is inaccessible after the construction process has been finished.

Interpreting integrity constraint (57) by updating with clause (60) solves the above mentioned problem. To be slightly more precise: There are now two possibilities to update a discourse model. If we choose the strong disjunct of clause (60) for a situation described by sentence (58) we will still produce an inconsistency, but now we can also use the first disjunct which then derives that in a situation described by sentence (58) the meaning of *unterbrechen* collapses into that of *abbrechen*. This seems intuitively correct. But now we run into a further problem. This time the problem is caused by the updating strategy of the logic programming approach.

Remember that an integrity constraint always computes a *minimal* update of a given discourse model. A model update satisfying the defining clauses for *abbrechen*, i.e. the antecedent of (53), requires fewer clauses to be satisfied than a model update for $unterbrechen_{strong}$, i.e. the antecedent of clause (56). The consequence is that in general the update will only satisfy the clauses for *abbrechen* und not the additional information for $unterbrechen_{strong}$. This in turn means that we predict that in general the meanings of *abbrechen* and *unterbrechen* are the same. This is clearly wrong.

However, in order to remedy this situation we can use information contained in DRS (23), which we repeat here as (61).

(61)
$$\left\langle \begin{array}{|c|} \hline t\ e\ x\ A'\ a\ s \\ \hline \mathrm{t} < \mathrm{n} \quad \mathrm{e} \subseteq \mathrm{t} \\ \text{e: unterbrechen(x, } \tau) \\ \text{PREP(A', } \tau) \\ \text{Agent(e) = x} \\ \text{A'(a)} \quad \text{Agent(a) = x} \\ \mathrm{a} \supset\!\subset \mathrm{e} \quad \mathrm{e} \subseteq \mathrm{s} \\ \text{s: } \neg \begin{array}{|c|} \hline e' \\ \hline \text{A'(e')} \\ \hline \end{array} \\ \hline \end{array}, DEFEAS \begin{array}{|c|} \hline e'' \\ \hline \text{A'(e'')} \\ \mathrm{s} \supset\!\subset \mathrm{e''} \\ \hline \end{array} \right\rangle$$

with $\tau = \lambda\alpha$.K' (See Section 3.2.6).

The DRS contains the feature DEFEAS. What does this feature mean? Let us first say what we don't want this condition to mean. DEFEAS is not a new logical operator in the sense that it gives rise to new inferences that would be absent without it. It is rather a kind of test. The operational meaning of DEFEAS as a marker in the DRS for sentence (55) may be paraphrased as follows: Take the strong disjunct in the antecedent of the clause defining $unterbrechen(e)$ and check whether this leads to an inconsistency in the updated discourse model[47]. If no inconsistency can be derived, use the strong clause as a definition of $unterbrechen(e)$ and add the component marked DEFEAS to the main DRS. The result in this case is:

[47]This can be done by the resolution machinery of logic programming.

(62)

$$\begin{array}{|c|}\hline
t\ e\ x\ A'\ a\ s\ e'' \\ \hline
\mathrm{t} < \mathrm{n}\ \ \mathrm{e} \subseteq \mathrm{t} \\
\mathrm{e{:}\ unterbrechen(x,}\ \tau) \\
\mathrm{PREP(A',}\ \tau) \\
\mathrm{Agent(e) = x} \\
\mathrm{A'(a)\ \ Agent(a) = x} \\
\mathrm{a} \supset\!\subset \mathrm{e}\ \ \mathrm{e} \subseteq \mathrm{s} \\
\mathrm{s{:}}\ \neg \begin{array}{|c|}\hline e' \\ \hline \mathrm{A'(e')} \\ \hline \end{array} \\
\mathrm{A'(e'')} \\
\mathrm{s} \supset\!\subset \mathrm{e''} \\ \hline
\end{array}$$

However, if a contradiction is derived, as in the case of example (58), the weaker disjunct in clause (60) is selected, which then specifies the inferential potential of the following DRS (63).

(63)

$$\begin{array}{|c|}\hline
t\ e\ x\ A'\ a\ s \\ \hline
\mathrm{t} < \mathrm{n}\ \ \mathrm{e} \subseteq \mathrm{t} \\
\mathrm{e{:}\ unterbrechen(x,}\ \tau) \\
\mathrm{PREP(A',}\ \tau) \\
\mathrm{Agent(e) = x} \\
\mathrm{A'(a)\ \ Agent(a) = x} \\
\mathrm{a} \supset\!\subset \mathrm{e}\ \ \mathrm{e} \subseteq \mathrm{s} \\
\mathrm{s{:}}\ \neg \begin{array}{|c|}\hline e' \\ \hline \mathrm{A'(e')} \\ \hline \end{array} \\ \hline
\end{array}$$

Since these two DRSs are the most important ones occurring in this paper we will explain briefly how the conditions they contain relate to the logic programs that are meant to capture their inferential potential[48]. Some of the entries like e $<$ t, e $\subseteq$ t are self-explanatory, but remember our provisos concerning past tense on Page 65. The condition e: unterbrechen(x, τ) corresponds to the event type *unterbrechen(e)* defined by clause (60). The condition PREP(A', τ) relateso two closely related components of the CLP system. The activity type A' relates to the *construct* fluent (type) of the scenario of the verb *absperren* (47). The fact that this is a preparatory activity – as explained in the entry for *absperren* in Section 3.2.6 – is accounted for by the *Trajectory*–predicate in which this fluent occurs as the first argument. The event type τ which is the second argument of PREP is mapped to the event type a introduced by hierarchical planning (49). The next two conditions Agent(e) = x and Agent(a) = x which tell us that the agents of the interrupt–activity and cordoning–off–activity have to be the same, are not represented explicitly in the clauses given so far. In order to account for this information in the CLP approach we have to be more explicit about the internal structure of fluents and events. This can be accomplished by Feferman coding. Consider the scenario for the

[48]A systematic translation of a DRT-fragment into integrity constraints is contained in Appendix 2.

verb *absperren* (47) again. This scenario contains the activity–fluent *construct*. The agent of this activity appears when we write this fluent in the form $construct[x, \hat{t}]$. A similar notation reveals the agent of the event type *unterbrechen(e)* defined by clause (60), for which we write $\exists t.unterbrechen[y, e, t]$. In minimal models the variables x and y will then be automatically unified be a (generalisation) of the unification algorithm. When we want to have different agents for the *interrupt-* and the *cordoning-off*-activity[49] we have to write additional integrity constraints which forbid the unification of the respective variables.

Finally, program clause (60) reflects the DRT–entry for the verb *unterbrechen*, in which the first disjunct corresponds to the non–defeasible part of the entry and the second disjunct to the non–defeasible part plus the defeasible resumption component.

To sum up: We claim that the time profile of the verb *unterbrechen* is roughly as given by the following picture:

$$- - - - - - - - - - - - - \qquad\qquad\qquad - - - - - - - - - - - - \downarrow$$

goal

By the law of inertia embodied in the axioms of the event calculus the goal-directed activity will, after having been interrupted and then resumed, eventually attain its goal in the computed minimal model. As explained, this is a defeasible conclusion. But in contrast to a verb like *behindern* this is not the only defeasibility aspect involved in the semantics of *unterbrechen*. The other aspect was discussed at length in Section 3: *unterbrechen* defeasibly implies that what is interrupted will be resumed. The reader will recall that we have represented this defeasible part of the meaning of *unterbrechen* by using the marker 'DEFEAS'. The 'DEFEAS' part of the meaning of *unterbrechen* can be overwritten, we have seen, by further information in the sentence or discourse, as in (58). In the absence of such defeating information, however, the 'DEFEAS' past of *unterbrechen* is incorporated into the semantics of the sentence just like the part that is not marked 'DEFEAS'. That then constitutes defeasible inference number one; in our reconstruction it takes place at the level of DRS construction. Once this defeasible inference has occurred, minimal model construction in the sense of the event calculus then adds goal completion as a second defeasible inference.

The conclusion of the above discussion is that the combination of DRT and event calculus that we propose in this paper solves the empirical problems we encountered with the semantics of the verb *unterbrechen* and relates the representational and the inferential potential of DRT in an effective way.

5.4 Programs and Integrity Constraints for DRSs: *aufheben*

The last verb we consider in this paper is *aufheben* as in (64).

(64) Die Absperrung der Botschaft wurde aufgehoben.
The blocking-off of the embassy was lifted.

[49] As mentioned in Section 3.2.6, there are such exceptions.

We noted in Section 3.2.7 that the verb *aufheben* can be used in the sense of lifting a physical object – for instance a crane lifting the fence which is used for blocking off the embassy – or in the sense of lifting a state – in our case the state of the embassy being inaccessible. At this point we are primarily interested in the second reading. To model this reading we cannot use our defined event type a which was introduced in Section 5.2 for the event reading of *Die Absperrung der Botschaft*, but we have to find something else. For this purpose let us consider the scenario for the verb *absperren* again.

1. $Initially(barrier(0))$
2. $Initially(accessible(b))$
3. $HoldsAt(barrier(m), t) \wedge HoldsAt(construct, t) \rightarrow$
   $Happens(finish, t)$
4. $Initiates(start, construct, t)$
5. $Initiates(finish, barrier(m), t)$
6. $Terminates(finish, accessible(b), t)$
7. $Terminates(finish, construct, t)$
8. $HoldsAt(barrier(x), t) \rightarrow$
   $Trajectory(construct, t, barrier(x + g(d)), d)$
9. $Releases(start, barrier(0), t)$

Assume that the cordoning-off of the embassy was successfully finished at some time s_0. This is clearly presupposed by the verb *aufheben*, when this presupposition is not fulfilled, the verb cannot be applied. We then derive $HoldsAt(barrier(m), s)$ and $\neg HoldsAt(accessible(b), s)$ for all $s > s_0$ on the assumption that there are no intervening events.

The fluent $barrier(m)$ is a material object which came into existence as a result of the constructing activity. Therefore we can choose this fluent as the denotation for the object reading of the DP *Die Absperrung der Boschaft*. Since – as already pointed out above – we are primarily interested in the state reading we will abstain from formalising this reading precisely here.

Consider now the formula $\neg HoldsAt(accessible(b), s)$. Using the Feferman system we can transform this formula into the fluent $\neg HoldsAt[accessible(b), \hat{s}]$, which we take as the denotation of *Die Absperrung der Botschaft* in its state reading. The structured object $\neg HoldsAt[accessible(b), \hat{s}]$ is an adequate candidate for this denotation since it encodes the period of time in which the embassy is not accessible. Let us therefore abbreviate this fluent in the following way: $\neg HoldsAt[accessible(b), \hat{s}] := inaccessible(b)$. We can now write an integrity constraint for sentence (64) this time not using the $Happens$–predicate but the $HoldsAt$–predicate of the event calculus.

(65) $?HoldsAt(inaccesssible(b), t),$
$HoldsAt(aufheben(inaccessible(b)), t), succeeds$

This integrity constraint does not yet capture the specific inference due to the verb *aufheben*. We want to conclude that after the state characterized by the embassy not being accessible has been cancelled the embassy is accessible again. To achieve this we have to write a meaning postulate for *aufheben*.

(66) $HoldsAt(aufheben(f), t) \wedge t < t' \rightarrow \neg HoldsAt(f, t')$[50]

Assume now that $HoldsAt(aufheben(inaccessible(b)), t)$ is correct for some time t. Clause (66) allows us to derive $\neg HoldsAt(inaccessible(b), t')$ for times $t' > t$. Substituting the definition of $inaccessible(b)$ in this formula we get:

$$\neg HoldsAt(\neg HoldsAt[accessible(b), \hat{s}], t')$$

which reduces to $HoldsAt(accessible(b), t')$. This is exactly the result we want. We therefore are now able to express the inferential potential of the following DRS [51].

[50]Strictly speaking this meaning postulate is not a formally correct program clause, since negated atoms are not allowed in the head of a clause. But instead of $\neg HoldsAt$ we can use $\overline{HoldsAt}$ which intuitively is the antonym of $HoldsAt$ in contrast to its classical negation. We will have to use antonyms anyway because we will iterate the $HoldsAt$–predicate in the following. Iterated $HoldsAt$–predicates necessarily lead to an antonymic interpretation of negation. Readers interested in the formal details are referred to chapter 6 of van Lambalgen and Hamm (2005).

[51]This DRS is obtained from the first reading of the entry (E5) for *aufheben*. The selection restrictions which this reading imposes on the theme argument are compatible only with the result state reading of *die Absperrung der Botschaft*. If the discourse referent s_1 which represents this reading in (7) is substituted for y in the Semantics of (E5,i), we obtain, as part of the resulting representation, the pair of conditions

(c.i) s: $\neg$ in force(s_1)
(c.ii) s_1: inaccessible(b)

We take it to be a Meaning Postulate connected with the predicate 'in force' that this pair of conditions can be simplified to a single condition, which says that s is a state to the effect that 'inaccessible(b)' does not hold. In fact, this is one half of a double-headed Meaning Postulate which covers both the case where the first condition predicates 'in force' and the case where it predicates '$\neg$in force'. The general form of the postulate is as follows:

(i) a.

s s'
s: inforce(s') s': X

⇒ s:X

b.

s s'
s: $\neg$ inforce(s') s': X

⇒ s: $\neg$X

An application of i.(b) and subsequent dropping of the conditions (c.i) and (c.ii) yields the DRS in the text.

(67)

s_0 s_1 e x y b
Absperrung$_s$(y,b)
Botschaft(b)
Agent(x)
e: aufheben(x, Absperrung$_{state}$(y,b))
CAUSE(e, s_1)
s_1:$\neg$ inaccessible(b)

Note, however, that the theory predicts that the fluents corresponding to *accessible(b)* and *inaccessible(b)* are related by the antonym relation, not by classical negation (see Footnote (50)).

Let us conclude this section with a remark about the kind of reasoning formalized by the event calculus. Meaning postulate (66) allows us to derive:

$$\neg HoldsAt(inaccessible(b), t')$$

for time $t' > t$. Since we may assume that at t the fluent $inaccessible(b)$ held for the first time (the completion of) Axiom 2 of the event calculus here repeated as (68)

(68) $\mathit{HoldsAt}(f, r) \wedge r < t \wedge \neg\exists s < r HoldsAt(f, s) \wedge$
$\neg\mathit{Clipped}(r, f, t) \rightarrow \mathit{HoldsAt}(f, t)$

allows us to derive that $\neg\mathit{Clipped}(r, f, t)$ fails for some interval. Therefore $Clipped$ must be true for this interval. The completion of Axiom 5 here repeated as (69)

(69) $\mathit{Happens}(e, s) \ \wedge \ t < s < t' \wedge$
$(\mathit{Terminates}(e, f, s) \ \vee \ \mathit{Releases}(e, f, s)) \ \rightarrow \ \mathit{Clipped}(t, f, t')$

allows us to derive that there is an event happening which either releases the fluent *inaccessible(b)* from the law of inertia – $Relases(e, inaccessible(b), s)$ – or terminates that this fluent holds – $Terminates(e, inaccessible(b), s)$. This seems to be intuitively correct.

6 Conclusion

In this paper we have presented reconstructions of two forms of inferencing that are part of the interpretation of certain natural language sentences. The proposals we have made give rise to two questions. First, there is the question whether our reconstructions are right, and in what sense. Do the proposals generalise in a natural way to other cases of the same sort - other selection-based resolutions of sortal ambiguities and other inferences based on minimal models? And do the proposals capture what actually goes on in the minds of human interpreters, and in what way and to what extent? To the first half of this question the paper says all that we can say to it at this moment, and about the second half we can do no more than speculate; we - and in this we are not alone - simply do not have a framework that allows us to formulate or check interesting answers to it.

The second question is more general. The . inferential phenomena we have discussed are just two of many that play a part in language interpretation. But how many such inference mechanisms are involved in interpretation of natural language input generally? This is a question which can and should be asked not just in relation to German, but also to other human languages. But even in relation to German alone it is one about which we still know comparatively little. Relevant information is becoming available gradually but slowly, for instance in studies of the inferences that may be involved in resolving certain anaphoric expressions or the presuppositions generated by certain presupposition triggers in certain contexts. But as contributions to the question what is the range of different inference mechanisms that interpreters may have to resort to when dealing with linguistic input these studies can only be regarded as fragmentary and anecdotal. A general overview and classification of such inference mechanisms, and of the 'inference' problems they can be used to solve, is urgently needed – as part of an account of a semantics-pragmatics interface that can serve as component of a comprehensive theory of linguistic meaning, as a necessary foundation to the powerful NLP systems we want to be able to build eventually, and, lastly, as a step towards a better understanding of the role that inference plays in human information processing.

With this we declare the paper to be concluded.

7 Appendix I: Axioms of the Event Calculus

Axiom 1 $\textit{Initially}(f) \;\rightarrow\; HoldsAt(f, 0)$

Axiom 2 $\textit{HoldsAt}(f, r) \wedge r < t \wedge \neg\exists s < r HoldsAt(f, s) \;\wedge$ $\neg\textit{Clipped}(r, f, t) \rightarrow \textit{HoldsAt}(f, t)$

Axiom 3 $\textit{Happens}(e, t) \wedge \textit{Initiates}(e, f, t) \wedge\; t < t^{'} \;\wedge \neg\textit{Clipped}(t, f, t^{'}) \rightarrow \textit{HoldsAt}(f, t^{'})$

Axiom 4 $\textit{Happens}(e, t) \wedge \textit{Initiates}(e, f_1, t) \wedge\; t < t^{'} \wedge t^{'} = t+d \wedge \textit{Trajectory}(f_1, t, f_2, d) \wedge$ $\neg\textit{Clipped}(t, f_1, t^{'}) \rightarrow \textit{HoldsAt}(f_2, t^{'})$

Axiom 5 $\textit{Happens}(e, s) \;\wedge\; t < s < t^{'} \;\wedge\; (\textit{Terminates}(e, f, s) \;\vee\; \textit{Releases}(e, f, s)) \;\rightarrow$ $\textit{Clipped}(t, f, t^{'})$

8 Appendix II: The 1981 DRT Fragment

Since in the main part of this paper we didn't develop an algorithm which transform DRSs into integrity constarints we will show in this appendix how to combine the original DRT fragment of Kamp (1981) systematically with integrity constraints and thus with the CLP formalism. We don't know yet whether such algorithms exist for DRT in general. The many existing varieties of DRT render a general answer nearly impossible, but we will hint at an extension of the DRT fragment of 1981 at the end of this appendix.

Definition 2 states which conditions and DRSs are syntactically well formed[52].

[52]For an introduction the reader is advised to consult Gamut (1991).

Definition 2 *[DRT-Syntax]*

1. *If P is a n–place predicate constant and $s_1, \dots, s_n$ are terms, then $P(s_1, \dots, s_n)$ is a condition.*

2. *If s and s' are terms, then $s = s'$ is a condition.*

3. *If Φ is a DRS, then $\neg\Phi$ is a condition.*

4. *If Φ and Ψ are DRSs, then $(\Phi \rightarrow \Psi)$ is a condition.*

5. *If Φ and Ψ are DRSs, then $(\Phi \vee \Psi)$ is a condition.*

6. *If $x_1, \dots, x_n$ are reference markers $(n \geq 0)$ and $\phi_1, \dots, \phi_m$ are conditions $(m \geq 0)$,then*
$$< \{x_1, \dots, x_n\}, \{\phi, \dots, \phi_m\} >$$
is a DRS

Since the predicates of the event calculus all contain argument slots for time variables or constants we first map every n-place predicate mentioned in clause (1) of definition 2 to an $n + 1$-place predicate P^* as follows:

$$P(s_1, \dots, s_n) \mapsto P^*(s_1, \dots, s_n, t)$$

Here t is a parameter for time. We then transform the new predicate via Fefermann coding into a fluent. The result is:
$$P^*[s_1, \dots, s_n, \hat{t}]$$

The equality relation in clause (2) of Definition 2 is translated analogously.

Assume now that an arbitrary temporal constant w is given. Neglecting complex conditions for the moment, i.e. clauses (3), (4), (5) of definition 2, we can translate DRSs containing only atomic conditions into integrity constraints in the following way: Given DRS
$$D =< \{x_1, \dots, x_n\}, \{\phi, \dots, \phi_m\} >$$
we write $\phi_i^*[x_1, \dots, x_n]$ for the reified translation of atomic condition ϕ_i (as explained above). It is understood that the discourse markers occurring in ϕ_i are among $x_1, \dots, x_n$. The integrity constraint correponding to DRS D is then:

$$HoldsAt(\phi_1^*[x_1, \dots, x_n], w), \dots, HoldsAt(\phi_m^*[x_1, \dots, x_n], w), succeeds\ iff$$

$$HoldsAt((\phi_1^*[x_1, \dots, x_n] \wedge \dots \wedge \phi_m^*[x_1, \dots, x_n]), w), succeeds$$

We still have to explain how to deal with the complex conditions (3), (4) and (5). Let us first consider a negated DRS $\neg\Phi$.

Assume that $HoldsAt(\phi^*, w)$ is the clause contained in the integrity constraint corresponding to DRS Φ in $\neg\Phi$. Let us write $\neg(that(\phi^*)$ for $\neg HoldsAt[\phi^*, \hat{t}]$. The integrity constraint for DRS $\neg\Phi$ now is:

(70)

$$?HoldsAt(\neg(that(\phi^*), w), succeeds$$

From this integrity constraint we get: $?HoldsAt(\neg(that(\phi^*), w), succeeds$ iff (by definition) $?HoldsAt(\neg HoldsAt[\phi^*, \hat{t}], w), succeeds$ iff (because $HoldsAt$ is a truth predicate) $?\neg HoldsAt(\phi^*, w), succeeds$ iff $?HoldsAt(\phi^*, w), fails$.

Let us now consider condition (4). We take again clause $HoldsAt(\phi^*, w)$ from the integrity constraint for DRS Φ and further clause $HoldsAt(\psi^*, w)$ from the integrity constraint for DRS Ψ. We will write $HoldsAt[\phi^*, \hat{t}] \rightarrow HoldsAt[\psi^*, \hat{t}]$ for $\neg HoldsAt[\phi^*, \hat{t}] \vee HoldsAt[\psi^*, \hat{t}]$ (see lemma 5 on page 216 in van Lambalgen and Hamm (2005)). Condition (4) of definition 2 then contributes integrity constraint (71).

(71)

$$HoldsAt(HoldsAt[\phi^*, \hat{t}] \rightarrow HoldsAt[\psi^*, \hat{t}], w), succeeds$$

This integrity constraint allows us to derive:

$$?HoldsAt(HoldsAt[\phi^*, \hat{t}] \rightarrow HoldsAt[\psi^*, \hat{t}], w), succeeds\ \mathit{iff}$$

$$?HoldsAt(\phi^*, w) \rightarrow HoldsAt(\psi^*, w),\ succeeds\ \mathit{iff}$$

$$IF\ ?HoldsAt(\phi^*, w),\ succeeds\ THEN\ ?HoldsAt(\psi^*, w),\ succeeds$$

The last line is in accordance with the general definition of integrity constraint from van Lambalgen and Hamm (2005) on page 101.

Condition (5) of definition 2 can be handeled in a similar way. Let again $HoldsAt(\phi^*, w)$ and $HoldsAt(\psi^*, w)$ be the clauses from the integrity constraints for DRSs Φ and Ψ respectivly. This time we form $HoldsAt[\phi^*, \hat{t}] \vee HoldsAt[\psi^*, \hat{t}]$. Clause (5) of definition 2 then contributes the following integrity constraint:

(72)

$$?HoldsAt(HoldsAt[\phi^*, \hat{t}] \vee HoldsAt[\psi^*, \hat{t}], w), succeeds$$

From (72) we derive

$$IF\ ?HoldsAt[\phi^*, w),\ fails\ THEN\ ?HoldsAt(\psi^*, w), succeeds$$

Again this is in accordance with the general definition of integrity constraint from van Lambalgen and Hamm (2005).

It is clear that the computational semantics provided by these integrity constraints does not coincide with the standard embedding semantics for the original DRT-fragment[53]. The models which are computed by the integrity constraints are minimal models of the DRSs in the sense of logic programing. These models are computed as the least fixpoints of the monotone consequence operator[54]. For instance given the following short discourse and its DRS[55] (74)

(73) No author writes a poem. James writes a novel. Sam writes a short story.

[53]For precise statments of the semantics of the DRT fragment see Kamp (1981) or Gamut (1991).

[54]For an introduction to the semantics of logic programs the reader is advised to consult Doets (1994), Nienhuys-Cheng and de Wolf (1997) or Jäger and Stärk (1998) for an overview article.

[55]We omit the analysis of the direct objects since this does not add anything that is relevant here.

(74)
$$\begin{array}{|c|}\hline j\ s \\ \hline \neg \begin{array}{|c|}\hline x \\ \hline \text{author(s)} \\ \text{writes-poems(x)} \\ \hline \end{array} \\ \text{writes-a-novel(j)} \\ \text{writes-a-short-story(j)} \\ \hline \end{array}$$

the computed minimal model $\mathcal{M}$ is:

$$\mathcal{M} = \{\mathbf{writes-a-novel(j,w), writes-a-short-story(s,w)}\}$$

This is certainly not an interesting model. For instance it does not contain the information that *James* and *Sam* are authors.

However, this can be easily changed by adding the program clauses in (75).

(75) a. *HoldsAt(writes-a-novel(x), t) → HoldsAt(author(x), t)*
b. *HoldsAt(writes-a-short-story(x), t) → HoldsAt(author(x), t)*

Providing such additional information is the task of scenarios (local programs) which we used in the body of this paper to describe e.g. the semantics of the verb *absperren*. But we used much more. For example the DRSs contained events and conditions of the form $e : absperren(x, b)$. But again it is not hard to see how to formulate a fixpoint semantics for a DRT-fraagment extended with events. Let us consider just the example $e : absperren(x, b)$. We first introduce an event type $\exists s.absperren[x, b,]$ supplied by Feferman coding and then use clauses of the form $Happens(\exists s.absperren[x, b, s], t)$ to express the respective integrity constraints.

Acknowledgements

We would like to thank Uwe Reyle, Antje Roßdeutscher and Torgrim Solstad. The research reported here was supported by the projects B4 and D1 of the Collaborative Research Centre "Incremental Specification in Context" (SFB 732) at the University of Stuttgart.

References

Bierwisch, M. (1993). Semantische und konzeptuelle repräsentationen lexikalischer einheiten. In Ruzicka, R. and Motsch, W., editors, *Untersuchungen zur Semantik*. Akademie-Verlag, Berlin.

Blackburn, P. and Bos, J. (2009). Representation and inference for natural language: Volume ii. Manuscript.

Condrovadi, C., Crouch, R., and van den Berg, M. (2001). General and specific concepts. In *Poceedings of the 13th Amsterdam Colloquium.*

Doets, K. (1994). *From Logic to Logic Programming*. The MIT Press, Cambridge, Mass.

Dowty, D. (1991). Thematic proto–roles and argument selection. *Language*, 67:547–619.

Eberle, K., Faaß, G., and Heid, U. (2009a). Corpus based identification and disambiguation of reading indicators for german nominalization. In *Proceedings of Corpus Linguistics 2009*, Liverpool.

Eberle, K., Faaß, G., and Heid, U. (2009b). Proposition oder temporalangabe? disambiguierung von ung-nominalisierungen von verba dicendi in nach-pps. In Chiarcos, C., de Castilho, R., and Stede, M., editors, *Von der Form zur Bedeutung: Texte automatisch verarbeiten/From Form to Meaning: Processing Texts Automatically. Proceedings of the Biennial GSCL Conference 2009, Potsdam*, Tübingen. Narr.

Feferman, S. (1984). Toward useful type-free theories i. *The Journal of Symbolic Logic*, 49:75–111.

Feferman, S. (2008). Axioms for determinateness and truth. *The Review of Symbolic Logic*, 1:204–217.

Fellbaum, C. (1998). A semantic network for verbs. In *Wordnet. An Electronic Lexical Database*. MIT-Press.

Gamut, L. (1991). *Logic, Language and Meaning: Volume II*. University of Chicago Press, Chicago.

Genabith, J., Kamp, H., and Reyle, U. (2010). Discourse representation theory. an updated survey. In Gabbay, D., editor, *Handbook of Philosophical Logic*. Kluwer Academic Press. to appear.

Geurts, B. (1999). *Presuppositions and Pronouns*. Elsevier Science.

Guarino, N. and Welty, C. (2000). Identity, unity, and individuality: Towards a formal toolkit for ontological analysis. In Horn, W., editor, *Proceedings of ECAI-2000: The European Conference on Artificial Intelligence*, Berlin. IOS Press.

Hamm, F., Kamp, H., and van Lambalgen, M. (2006). There is no opposition between formal and cognitive semantics. *Theoretical Linguistics*, 32:1–40.

Jäger, G. and Stärk, R. (1998). A proof-theoretic framework for logic programming. In Buss, S., editor, *Handbook of Proof Theory*, pages 639–682. Elsevier.

Kamp, H. (1981). A theory of truth and semantic representation. In Groenendijk, J., Janssen, T., and Stokhof, M., editors, *Formal Methods in the Study of Language*. Amsterdam: Mathematical Centre Tracts 135, Amsterdam.

Kamp, H. (2001). The importance of presupposition. In Rohrer, C., Roßdeutscher, A., and Kamp, H., editors, *Linguistic Form and its Computation*. CSLI-Publications, Stanford.

Kamp, H. (2003). Einstellungszustände und einstellungsbeschreibungen in der diskursrepräsentationstheorie. In Haas-Spohn, U., editor, *Intentionalitvieät zwischen Subjektivität und Weltbezug*. Mentis, Paderborn.

Kamp, H. (2008). Discourse structure and the structure of context. University of Stuttgart.

Kamp, H. and Reyle, U. (1993). *From Discourse to Logic*. Reidel, Dordrecht.

Kamp, H. and Reyle, U. (1996). A calculus for first order discourse representation structures. *Journal of Logic, Language and Information*, 5.

Kamp, H. and Reyle, U. (2010). Discourse representation theory. In von Heusinger, K., Maienborn, C., and Portner, P., editors, *Handbook of Semantics*. Elsevier. to appear.

Kamp, H. and Roßdeutscher, A. (1994). Drs construction and lexically driven inference. *Theoretical Linguistics*.

Kamp, H. and Roßdeutscher, A. (2010). Syntactic and semantic constraints in the formation and interpretation of *-ung*-nouns. In Alexiadou, A. and Rathert, M., editors, *Nominalsation across Languages and Frameworks*. to appear.

Kohlhase, M., Kuschert, S., and Pinkal, M. (1995). A type-theoretic semantics for λ-drt. In *Proceedings of the Tenth Amsterdam Colloquium*.

Kowalski, R. A. and Sergot, M. (1986). A logic-based calculus of events. *New Generation Computing*, 4:65–97.

Krifka, M. (1989). *Nominalreferenz und Zeitkonstitution*. Fink, München.

Kuschert, S. (1996). Higher order dynamics: Relating operational and denotational semantics for λ-drt. Technical report, CLAUS-Report 72, Universität des Saarlandes.

Lenat, D. (2006). Hal's legacy: 2001's computer as dream and reality. from 2001 to 2001: Common sense and the mind of hal. Technical report, ycorp, Inc.. http://www.cyc.com/cyc/technology/halslegacy.html. Retrieved 2006-09-26.

Lifschitz, V. (1994). Circumscription. In Gabbay, D., Hogger, C., and Robinson, J., editors, *Handbook of Logic in Artificial Intelligence and Logic Programming: Vol 3*. Clarendon Press, Oxford.

McCarthy, J. and Hayes, P. (1969). Some philosophical problems from the standpoint of artificial intelligence. In Michie, D. and Meltzer, B., editors, *Machine Intelligence 4*. Edinburgh University Press, Edinburgh.

Nienhuys-Cheng, S.-H. and de Wolf, R. (1997). *Foundations of Inductive Logic Programming*. Springer, New York, Berlin.

Partee, B. (1987). Noun phrase interpretation and type shifting principles. In Groenendijk, J., d.Jongh, D., and Stokhof, M., editors, *Studies in Discourse representation Theory and the Theory of Generalized Quantifiers*. Foris, Dordrecht.

Partee, B. and Rooth, M. (1983). Generalized conjunction and type ambiguity. In Bäuerle, R., Schwarze, C., and von Stechow, A., editors, *Meaning, Use, and Interpretation of Language*. Walter de Gruyter, Berlin.

Pustejovsky, J. (1995). *The Generative Lexicon*. MIT-Press, Cambridge.

Reyle, U., Roßdeutscher, and Kamp, H. (2008). Ups and downs in the theory of temporal reference. *Linguistics and Philosophy*.

Roßdeutscher, A. (2000). *Lexikalisch gestützteformale Textinterpretation*. Arbeitsberichte des Sonderforschungsbereichs 340, Stuttgart/Tübingen, Nr. 157. Habilitationsschrift Universität Stuttgart.

Shanahan, M. (1997). *Solving the Frame Problem – A Mathematical Investigation of the Common Sense Law of Inertia*. MIT Press, Cambridge, MA.

Smith, C. (1991). *The Parameter of Aspect*. Kluwer.

Sternefeld, W. (1995). Voice phrases and their specifiers. Seminar für Sprachwissenschaft, Universität Tübingen, SfS-Report 05-95.

van der Sandt, R. (1992). Presupposition projection as anaphora resolution. *Journal of Semantics*.

van Lambalgen, M. and Hamm, F. (2005). *The Proper Treatment of Events*. Blackwell, Malden.

Anaphora resolution and reambiguation

Fritz Hamm (fritz.hamm@ims.uni-stuttgart.de)
Institute for Natural Langauge Processing
University of Stuttgart

Torgrim Solstad (torgrim.solstad@ims.uni-stuttgart.de)
Institute for Natural Language Processing
University of Stuttgart

Abstract Starting off from common assumptions on the relationship between ambiguity and the process of disambiguation with regard to the technique of underspecification, we argue that disambiugation may be viewed as non-monotonic in nature in certain cases involving anaphora resolution. We then go on to present a formal analysis framed in a coupling of Discourse Representation Theory and Constraint Logic Programming. Concluding the paper, we also discuss some consequences of our proposal for formal discourse semantics in general.

1 Introduction

Lexical ambiguity and its resolution is mostly only approached from an inter-sentential perspective. Thus, when analyzing the ambiguity and disambiguation of a lexical item, one mostly studies (i) the behaviour of other lexical items that modify or select the lexical item in question, as witnessed by the first paper in this collection (Hamm and Kamp this volume). Broadening this perspective in the present paper, we will argue that there are important insights to be gained with regard to disambiguation by studying more closely how disambiguated expressions behave in contexts spanning more than one sentence. More specifically, we will study cases of anaphora resolution involving antecedents which are disambiguated and anaphora which refer to one of the possible readings of the antecedent which was not selected in the disambiugating antecedent context. We will argue that such cases call for what we term a *reambiguation* of the antecedent expression, reintroducing an interpretation which was originally excluded.

The paper is structured as follows. In Section 2, we discuss properties of ambiguity, underspecification and disambiguation. We also provide an informal overview of our approach, including a discussion of the notion of *reambiguation*. In Section 3, the formal basis of our analysis is presented. In Section 4, we present the analysis and discuss the consequences of our approach for formal discourse semantics in general. Section 5 concludes the paper.

2 Disambiguation and Underspecification

Formally, ambiguities are often represented by means of underspecification, which is by many thought to be a more efficient way of handling the interpretational variance of expressions in the case of e.g. both scopal and lexical ambiguities. Thus, if no disambiguation can take place, for instance in cases where insufficient information is present to make an informed decision on which interpretation to choose, the underspecified representation

Antje Roßdeutscher (ed.): Reambiguation and Disambiguation (SinSpec06)
Working Papers of the SFB 732 *Incremental Specification in Context*,
University of Stuttgart, 2009: 69 – 92

allows us to defer interpretational decisions to a later point at which information helping us in our interpretational decision may be available. In this paper, however, we will focus on another aspect of underspecification, namely its role in the relation between ambiguous and disambiguated expressions.

One standard for representing lexical (or scopal) ambiguities by means of underspecification is to formalize the range of interpretations of an expression by means of disjunction (Reyle, 1996) or conjunction (Poesio, 1996). In the Underspecified Discourse Representation Theory (UDRT) approach of Reyle, for instance, underspecification is represented by means of the disjunctive operator $\overset{!}{\vee}$, cf. the simplified representation of the two-way ambiguous deverbal nominalization *delivery* in (1):

(1) $$\left\langle \alpha \begin{array}{|c|} \hline \\ \hline \alpha = \mathrm{e} \overset{!}{\vee} \alpha = \mathrm{y} \\ \text{e: deliver(x,y)} \\ \text{AGENT(e)=x} \\ \text{THEME(e)=y} \\ \hline \end{array} \right\rangle$$

More specifically, the representation in (1) shows the semantic representation for *delivery* at NP level. In (1), α represents the referential argument of the noun phrase which is assumed to be bound at DP level. As indicated in the first line of the condition part of this representation, the referential argument α of *delivery* may either be an event or an object, the latter corresponding to the theme of the verb *deliver*.

Assuming a disjunct or conjunct representation of ambiguous expressions, disambiguation is often viewed as a process of disjunct or conjunct deletion. Thus, the disambiguating contexts for *delivery* in (2) are often thought to lead to a deletion of the first or second disjunct in the DRS condition in (1).[1]

(2) a. the damaged delivery ($\cancel{\alpha = \mathrm{e}} \overset{!}{\vee} \alpha = \mathrm{y}$)
b. the quick delivery ($\alpha = \mathrm{e} \overset{!}{\vee} \cancel{\alpha = \mathrm{y}}$)

In (2), *damaged* is assumed to combine only with the object reading of *delivery*, whereas *quick* selects only the event reading.

Our data will mainly involve German deverbal nominalizations. More specifically, we will present a study of nouns derived by means of the suffix *-ung* (comparable both to *-tion* and *-ing* nominalizations in English, cf. Ehrich andRapp, 2000; Rossdeutscher and Kamp, 2010). While all productively derived *-ung* nouns have an event reading, quite a few *-ung* derivations additionally have state and/or object readings, cf. the examples in (3), involving *Absperrung* (from *absperren* ‘cordon off’), which is three-way ambiguous:

(3) a. *Die Absperrung wird morgen abgebaut.*
the barrier will be tomorrow dismantled
‘The barrier will be dismantled tomorrow.’
b. *Die Absperrung des Gebiets wird noch aufrecht erhalten.*
the cordoning-off the area is still sustained
‘The cordoning-off of the area is still sustained.’

[1] Similar remarks may be made for Poesio’s 1996 disambiguation inference mechanism.

c. *Die Absperrung des Gebiets wurde von den Demonstranten behindert.*
the cordoning-off the area was by the protesters hampered
'The cordoning-off of the area was hampered by the protesters.'

All noun phrases headed by *Absperrung* in (3) are disambiguated in context: the predicate *abbauen* 'dismantle' (3a) is assumed to select for object interpretations, *aufrecht erhalten* 'sustain' (3b) for states and *behindern* 'hamper' (3c) for event interpretations (for details see Hamm & Kamp this volume). A simplified, underspecified semantic representation covering all three readings is provided in (4):

(4)
$$\left\langle \alpha \;\; \begin{array}{|c|}\hline z \\ \hline \alpha = \mathrm{e} \stackrel{!}{\vee} \alpha = \mathrm{s} \stackrel{!}{\vee} \alpha = \mathrm{y} \\ \text{e CAUSE s} \\ \text{s: HAVE(y,z)} \\ \text{FUNCTION\_AS\_BARRIER(y)} \\ \text{AGENT(e)=x} \\ \hline \end{array} \right\rangle$$

Briefly stated, *Absperrung* involves an event e leading to a state s in which the (incremental) theme y blocks access to some region z. Again, the topmost condition of the representation provides information on the possible referential arguments of the noun: it may be an event (e), a state (s) or an object (y). For details on the logic and ontology of disambiguiation, the reader is referred to Hamm & Kamp (this volume).

Taking the above disambiguating contexts of *Absperrung* as a starting point, one can show that for anaphora resolution, a naive deletion approach to disambiguation makes the wrong predicitions, cf. the sequence in (5):

(5) *Die Absperrung des Rathauses wurde vorgestern von Demonstranten behindert. Wegen anhaltender Unruhen wird* ***sie*** *auch heute aufrecht erhalten.*
'The cordoning-off of the town hall was disturbed by protesters the day before yesterday. Due to continuing unrest, it [the state of being cordoned off] is sustained today as well.'

In (5), the anaphora *sie* 'it' is clearly coreferential with the noun phrase headed by *Absperrung* in the first sentence. As just stated, the predicate *behindern* 'hamper' restricts the ambiguity of *Die Absperrung des Rathauses* and fixes an event reading of the noun phrase. However, the matrix predicate in the second sentence, *aufrecht erhalten* 'sustain', only allows the referential argument of the anaphora *sie* to be a state. But if the fixation of the event reading, i.e. the disambiguation of *Absperrung*, involves the irreversible deletion of its other possible referential arguments, there should be no appropriate discourse referent for *sie* to pick up, contrary to intuitions.

Attempting to preempt some of the most obvious arguments against granting examples such as (5) any special status, let us briefly remark that a sloppy approach within centering theory (cf. e.g. Hardt, 2003) does not seem to offer a straightforward solution, since center shifting is not available if the relevant discourse referent has been elided in the preceding context. Attempting to avoid the problem by assuming that disambiguation does not involve any deletion whatsoever is no option (this is the "lazy" option discussed also by Hardt), as this would predict that every possible discourse referent of a noun is always available in subsequent sentences. The following example, which we will discuss below, shows that this is not the case:

(6) *#Die Absperrung wurde heute verstärkt. Sie war am Vortag massiv behindert worden.*
Intended: 'The barrier was fortified today. It [the cordoning-off] had been massively hampered the day before.'

Before turning to the formal details of our analysis, we would like to give its main characteristics in informal terms. To account for the acceptability of examples such as (5), we reconstruct the required result state which the anaphora *sie* makes reference to. We show that such a reconstruction is possible even under the assumption that *behindern* erases the result state reading of the first sentence in (5). This is achieved in a process of *reambiguation*, which involves a three-step procedure of inference, reification (turning a predicate into a term) and unification. This reconstructed result state then serves as a suitable antecedent for the anaphoric pronoun *sie* of the second sentence in (5). More specifically, the procedure may be described as follows: Although there is no semantically suitable antecedent – in terms of semantic types – for the pronominal anaphora *sie* in (5), one can certainly assume that the discourse referent of the anaphor may be identified with the referent of the DP *die Absperrung des Rathauses*, based on the constraints on referential identification for the discourse referent introduced by *sie*, taking gender (which excludes the referent of *Rathaus*) and number (which excludes the referent of *Demonstranten*)) into consideration. These constraints trigger a mapping from the event denotation of *die Absperrung des Rathauses* to the result state, involving a non-monotonic inferential process. The following pieces of information are of relevance for this process:

- The semantics of *Absperrung*, which derives from the verb *absperren*, involves an object (y), which incrementally constructed in order to block access to a region (z), i.e. the agent of the event causes a state (s) of inaccessibility of the region (z).

- The referential argument of the predicate *aufrecht erhalten* is of result state type, while the one of *behindern* is of event type.

- The properties of the pronoun *sie* – its referent needs to be identified with one which is introduced by a DP – requires a mapping from the event referent of the DP *die Absperrung des Rathauses* to the result state of being cordoned off. This state is accessible via the semantics of the predicate *absperren*. The mapping from the event to the state consists in an abstraction over the times for which the predicate holds (from $\mathtt{absperr(e,t)}$ to the reified $\mathtt{absperr[e,}\hat{t}\mathtt{]}$). This set of times can in principle both be that for which the process of cordoning-off holds as well as the one for which the result state holds. In our analysis, we only exploit the latter possibility, since we assume that the predicate *aufrecht erhalten* 'sustain' only applies to *result* states.

- Consequently, a non-monotonic inferential process is initiated, in which the coming about of the result state of being cordoned off is inferred from the occurrence of the process of cordoning-off.

Moreover, the inference which is triggered by *behindern* and blocked by *verhindern* respectively (cf. Hamm & Kamp this volume) allows an explanation of the difference concerning the possibility of anaphora resolution in (5) versus (7).

(7) #*Die Absperrung des Rathauses wurde vorgestern von Demonstraten verhindert. Wegen anhaltender Unruhen wird* **sie** *auch heute aufrecht erhalten.*
'The cordoning-off of the town hall was prevented by protesters the day before yesterday. Due to continuing unrest, it [the state of being cordoned off] is sustained today as well.'

The problematic case in (8) is accounted for under the assumption that objects are represented by predicates without temporal parameter. In this case, anaphora resolution is correctly blocked, since the above depicted three-step procedure involving inference, reification and unification is not applicable for predicates without temporal parameters.

(8) #*Die Absperrung wurde heute verstärkt. Sie war am Vortag massiv behindert worden.*
Intended: 'The barrier was fortified today. It [the cordoning-off] had been massively hampered the day before.'

Concerning the notion of reambiguation, it should be noted that the process of reambiguating may involve a complete recovery of all readings which were deleted in previous context, cf. (9), where *ignorieren* allows *sie* to have a referential argument of all three possible types (object, event and result state), whereas the *Absperrung*-DP in the first sentence clearly only has an event reading:

(9) Die Absperrung des Rathauses wurde von Demonstranten behindert. Später haben sie alle ignoriert.

We now turn to a formalisation of the above description.

3 Event Calculus

Before we start to develop integrity constraints and programs for DRSs, we will give a short informal introduction to the event calculus. For a much more comprehensive introduction the reader is referred to van Lambalgen and Hamm (2005). The event calculus was originally developed on the basis of McCarthy's situation calculus (McCarthy and Hayes, 1969) by Kowalski and Sergot (1986) and Shanahan (1997) and used for high level control of mobile robots. The theoretical aim pursued with this calculus was the solution of the frame problem in Artificial Intelligence.

3.1 Linguistic Motivation

Consider the following short piece of discourse:

(10) It was hot. Jean took off his sweater.

We naturally understand that the eventuality expressed by the second sentence is included in the temporal profile of the eventuality expressed by the first sentence. In order to establish this temporal overlap one could intuitively argue as follows:

(11) World knowledge contains no link to the effect that taking off one's sweater changes the temperature. Since it is hot at some time before *now*, the state *hot* must either hold initially or must have been initiated at some time t. The latter requires an event, which is however not given by the discourse. Therefore *hot* holds initially. Similarly no terminating event is mentioned, so that *hot* extends indefinitely, and it follows that the event described by the second sentence must be positioned inside the temporal profile of *hot*.

The event calculus is meant to formalize this kind of argumentation. Note the following important feature of the above argument. Several steps use a non–monotonic inference scheme. For instance, the conclusion that the state *hot* holds initially is derived from the observation that the discourse does not mention an initiating event. From this observation we conclude that there is no initiating event, leaving only the possibility that *hot* holds initially. A second feature of this reasoning involves the principle of *inertia*. This principle, which is axiomatized by the axioms of the event calculus, states that if a state – *hot* in our example – is not forced to change under the impact of an event, it is assumed to remain unchanged.

This specific kind of non–monotonicity is intimately linked to the event calculus as a planning formalism. Planning is defined as setting a goal and devising a sequence of actions that will achieve that goal, taking into account events in the world, and properties of the world and the agents. Now given a goal G and circumstances C under which G can be achieved, it does not follow in a strict sense that G can be achieved under C plus some additional circumstances D. In this sense a planning system requires a non-monotonic formalism.

A close connection between planning and linguistic processing is established by assuming that a sentence S is considered as a goal (make S true) to be achieved by updating the discourse model. This means that we can model the understanding of a sentence in discourse as such a goal. The goal is to make a sentence – as part of a discourse – true by accommodating those facts necessary for establishing the truth of the sentence. This is one of the leading ideas of DRT. In example (10), the first sentence provides a discourse model which is updated to make the second sentence true unless it is forced to give up essential parts due to explicit information incompatible with it.

We will now proceed to describe the event calculus a bit more formally. We start with the language of the event calculus.

3.2 The language of the event calculus

Formally, the event calculus is a many-sorted first order logic. The sorts include event types, fluents (time-dependent properties, such as activities), real numbers, and individuals. We also allow terms for fluent-valued and event type-valued functions.

The event calculus was devised to model formally two notions of change, instantaneous change – such as two balls colliding – and continuous change – for instance the acceleration of a body in a gravitational field. A first series of primitive predicates is used for modelling instantaneous change.

(12) *Initially*(f)

(13) *Happens*(e, t)

(14) *Initiates*(e, f, t)

(15) *Terminates*(e, f, t)

The intended meaning of these predicates is more or less self-explanatory. The predicate *Initially*(f) takes as its argument a fluent and says that f holds at the beginning of a scenario. *Happens*(e, t) holds if event type e happens at time point or interval t. The event calculus allows to interpret t as a point or as an interval. *Initiates*(e, f, t) says that event type e causes f to be true strictly after t; i.e. f does not hold at t. Finally, *Terminates*(e, f, t) expresses that f holds at t and that e causes f not to hold after t.

The next two predicates are used to formalize continuous change.

(16) *Trajectory*(f_1, t, f_2, d)

(17) *Releases*(e, f, t)

The 4–place predicate *Trajectory*(f_1, t, f_2, d) measures the change of f_2 under the force f_1 in the interval from t to $t + d$. Linguistically, it is very close to the notion of incremental theme (see for instance Krifka, 1989; Dowty, 1991). One may think of f_1 as an activity which acts on f_2. Dowty uses *mowing a lawn* in order to explicate the notion *incremental theme*. In Dowty's example f_1 is the mowing activity and f_2 the changing state of the lawn under this activity. The fluent f_2 should therefore be considered a parameterized partial object; in Dowty's example the state of the lawn after d time steps of the ongoing activity of *mowing*. The axioms of the event calculus then provide the homomorphism between the ongoing activity and the resulting (partial) state – the partially mowed lawn – as required by Dowty.

The *Releases*(e, f, t) predicate is necessary for reconciling the two notions of change formalized by the event calculus. Without this predicate the axioms would immediately produce an inconsistency. Intuitively, the *Releases* predicate says that after event e happened, f is no longer subject to the principle of inertia. This allows f to change continuously. Consider a scenario of filling a bucket with water. Event type *tap–on* releases the parametrized fluent *height*(x) that measures the continously changing level of the water in the bucket from the principle of inertia.

The *Clipped*–predicate of the calculus expresses that an event either terminating fluent f or releasing this fluent from the principle of inertia occurred between times t_1 and t_2.

(18) *Clipped*(t_1, f, t_2)

The last predicate states that fluent f is true at time t.

(19) $HoldsAt(f, t)$

'*HoldsAt*' should be considered a truth predicate although the axioms of the event calculus do not contain the characteristic truth axiom, i.e.

$$HoldsAt(\overline{\phi}, t) \leftrightarrow \phi(t)$$

where $\overline{\phi}$ is a name for formula ϕ. More formal machinery is necessary to transform *HoldsAt* into a truth predicate satisfying the characteristic truth axiom. We will resume the discussion of this topic in section 3.5.

In the next section we will introduce the axioms of the event calculus in an informal way and motivate their use by way of the above reasoning example (11).

3.3 Axiomatization

In this section we will show how the axioms of the event calculus constrain the meanings of the basic predicates and how they formalize the principle of inertia. Moreover we will illustrate how the concept of the completion of a program helps to implement the intutive idea that events that are not required to happen by a narrative are assumed not to occur. We will demonstrate that this strategy forces the reasoning to be non–monotonic. Let us start with an informal example.

(20) If a fluent f holds initially or has been initiated by some event occurring at time t and no event terminating f has occurred between t and $t' > t$, then f holds at t', (here $>$ indicates the temporal precedence relation).

It is clear that this axiom embodies a law of inertia since if no f-related event occurs then f will be true indefinitely. In the reasoning of example (11), this axiom was used for instance when we concluded from the fact that no terminating event for *hot* is mentioned that this state holds indefinitely with regard to the story told so far. But this was not the only reasoning principle we applied. From the fact that no terminating event was mentioned in the short discourse we concluded that none occurred. The axioms of the calculus per se do not allow such a conclusion. We want a strengthening of the assumptions in which only those events occur which are explicitly mentioned in the discourse. In this sense understanding discourses is closely linked to *closed world reasoning*.[2] There are many techniques for formalizing this kind of reasoning; one is circumscription (for a good overview see Lifschitz, 1994). In this paper, however, we use the notion of the *completion of a logic program*. The advantage of logic programming is that these techniques allow us to *compute* discourse models via fix point constructions.

Let us be slightly more formal. The informal principle (20) is given by the combination of the following two axioms:

1. $\textit{Initially}(f) \;\rightarrow\; HoldsAt(f,0)$
2. $\textit{Happens}(e,t) \wedge \textit{Initiates}(e,f,t) \wedge\; t < t^{'} \;\wedge \neg \textit{Clipped}(t,f,t^{'}) \rightarrow \textit{HoldsAt}(f,t^{'})$

The most important feature to notice here is that the head – the part to the right of the implication sign – consists of a simple atom, and the body – the part to the left of the implication sign – consists of a combination of formulas from two languages. The first language is the language of the event calculus and the second language is the first order language of the reals, i.e. of the structure $(\mathbb{R}, 0, 1, +, \cdot, <)$. This means that the axioms

[2]A typical example of this kind of *closed world reasoning* is provided by (train) schedules. If the schedule mentiones the departure of a train from Stuttgart to Tbingen at 10.15 and the next at 11.01 one assumes that there will be no train leaving Stuttgart between 10.15 and 11.01.

are clauses of a constraint logic program. The formulas of the second language, such as $t < t'$, are the constraints of the constraint logic program. They are used to compute the time profile of the predicates of the event calculus. All variables in the clauses of logic programs are supposed to be universally quantified.

The completion of a program is a strengthening of it which explicitly expresses that the predicates occurring in the program have extensions that are as small as possible. Before we apply the method of completion to the examples on which we focus in this paper, we indicate how it works at the hand of a very simple program taken from Nienhuys-Cheng and de Wolf (1997).

(21) a. *Prof(confucius)* (Confucius is a professor.)
b. *Prof(socrates)* (Socrates is a professor.)
c. $\neg Prof(y) \rightarrow Student(y)$ (Every person who is not a professor is a student.)

The program involves two predicates, *professor* and *student*. The programming formalism is set up in such a way that it is only possible to make positive statements about the extensions of predicates. Thus (21) states about the predicate *professor* that *confucius* belongs to its extension (21a) and also that *socrates* belongs to its extension (21b); and these are all the definite claims the program makes about the extension of this predicate. The completion of the program ought to make this intuition concrete by stating explicitly that the extension of *professor* consists just of these two individuals. We accomplish this by forming the disjunction of the formulas $x = confucius$ and $x = socrates$, where x is a new variable, which intuitively plays the role of an arbitrary member of the extension of *professor*, and making this disjunction into the antecedent of the following implication:

(22) $x = confucius \vee x = socrates \rightarrow Prof(x)$

In the next step we universally quantify over the variable x and strengthen the implication to a bi–implication. The result is:

$$\forall x(x = confucius \vee x = socrates \leftrightarrow Prof(x))$$

This formula now says that the set of professors just consists of Confucius and Socrates. Under the assumption that Confucius and Socrates are the only individuals in the model we get that the set of students is empty. But assume now that the language in which the program is formulated contains an additional individual constant $plato$ which is interpreted as an element of the universe of discourse. Assume further that $socrates \neq confucius \neq plato$.[3] Then (22) implies that $plato$ is not a professor. Now consider the third clause of program (21). A similar procedure applied to this clause yields:

(23) $\forall x(Student(x) \leftrightarrow \neg Prof(x))$[4]

[3] This is an instance of the 'uniqueness of names' assumption.

[4] This is technically not quite correct. The formula produced by the official algorithm for computing the completion of a program is:

$$\forall x(Student(x) \leftrightarrow \exists y(x = y \wedge \neg Prof(y)))$$

Formula (23) implies that Plato is a student. The conjunction of (22) and (23) is the completion of program (21). This completion implies that Confucius and Socrates are the only professors and that Plato is a student. The program itself does not support such strong conclusions. A similar observation applies to certain extensions of (21) that bring additional entities into play. Suppose for instance that we add to (21) the fact *beard(plato)*, which states that Plato has a beard. A minimal model for the completion of the extended program will have the universe $\{confucius, socrates, plato\}$. In this model Plato is not a professor, but the only student and the only one with a beard.

Let us now give a simple example with events. Consider a description of a situation where the light is switched on at 1 in the night and switched off at 7 in the morning and given by the following program:

(24) a. $Happens(switch\text{-}on, 1)$
b. $Happens(switch\text{-}off, 7)$

The uncompleted program does not yet imply that the light wasn't switched off at 2 in the night and switched on at 3 in the night and so on. However, these events should not occur in the minimal model of program (24). The completion of the program is given by

$$\forall e(Happens(e, t) \leftrightarrow (e = switch\text{-}on \wedge t = 1) \vee (e = switch\text{-}off \wedge t = 7))$$

This formula means the same as:

$$\forall e(Happens(e, t) \leftrightarrow (Happens(switch\text{-}on, 1) \vee (Happens(switch\text{-}off, 7))$$

Any intervening events are thereby excluded.

This illustrates how the concept of the completion of a program helps to implement the intuitive idea that events that are not required to happen by a narrative are assumed not to occur. Note that this strategy forces the reasoning to be non–monotonic. We could easily enrich program (24) with clauses *Happens(switch-off, 2)* and *Happens(switch-on, 3)*. From the modified program the conclusion that there are no events happening between *Happens(switch-on, 1)* and *Happens(switch-off, 7)* is now no longer derivable.

To sum up: Understanding a sentence in a discourse is like computing a minimal model of the discourse in which the sentence is true. This computation is based on the completion of a constraint logic program for the discourse under discussion. In the next section we will see, however, that this aim cannot be achieved by the technical means introduced so far.

3.4 Integrity Constraints

As pointed out above, the variables in the clauses of logic programs are universally quantified. Therefore logic programs are restricted to provide universal information only. This is clearly not sufficient for our purpose. For example, tense requires existential information (see the example below) and DRSs in general introduce existential information. We will use here a device from database theory – integrity constraints – to obtain the required additional information. In database theory integrity constraints are means to ensure that a

---

But for the simple example discussed above this difference does not matter. The official formula and (23) are equivalent.

database stays consistent under updates. In this paper we will use integrity constraints in a slightly different way; we employ them as means to update a discourse model. Let us explain this idea with a simple example, involving an English sentence in the perfect.

(25) I have caught the flu.

This sentence says that I have the flu now and world knowledge tells us that there was an infection event in the past. Let flu be the fluent corresponding to *having the flu* and let e be the infection event. Our knowledge is thus formalized by the following program clause.

$$Initiates(e, flu, t)$$

As already said we view a sentence S as a goal (make S true) to be achieved by updating the discourse model. In general it is not possible, however, to simply add this information to the discourse model without further ado. There are two reasons for this. First, we would like the updated discourse model to include explicitly all the events that must have occurred in order for the total information represented by it to be true. And, second, when the spelling out of what that comes to reveals a conflict, it should mean that the new sentence cannot make a coherent contribution to the discourse as the initial model represents it. It is therefore important that we do not just add the condition that I have the flu now, but also the event that must have led to this state of affairs. The formalisation of the event calculus given earlier offers a systematic way of doing this. In the present instance what needs to be inferred from $HoldsAt(flu, now)$ is that there was an earlier event e initiating flu, something that is expressed in the present formalism by the clauses $Initiates(e, flu, , t)$, $Happens(e, t)$ and $t < now$.

We will now show how this reasoning applies to example (25). For this purpose, assume that a discourse model is given as a collection of facts concerning events and fluents and assume that sentence (25) is formalized as $HoldsAt(flu, now)$. We do not take this formula as a program clause but as an instruction to construct a minimal adaptation of the discourse model in which $HoldsAt(flu, now)$ is true. In order to detect the events that must have occurred for $HoldsAt(\mathit{flu}, now)$ to be true, we apply abductive reasoning using the basic program constituted by the axioms of our formulation of the event calculus, as well as, possibly, additional axioms that capture aspects of world knowledge. To this end, we use $HoldsAt(\mathit{flu}, now)$ as the trigger that sets this reasoning process in motion. Informally, the reasoning is as follows. We know that fluent *flu* is initiated by some event e. No terminating event has been mentioned. Therefore we conclude by closed world reasoning that no such event occurred. Consider again axiom (20) repeated here as (26).

(26) If a fluent holds initially or has been initiated by some event occurring at time t and no event terminating f has occurred between t and $t' > t$, then f holds at t'.

According to this axiom there is only one fact missing to establish the truth of $HoldsAt(f, now)$. We have to add $Happens(e, t), t < now$ and its logical consequences to the discourse model. This is sufficient to guarantee the truth of *HoldsAt(flu, now)*.[5]

[5]There is a subtle difference between (25) and sentence (i)

Let us now be a little bit more formal and see how this update is steered by the proof system of logic programming, which is called *resolution*. Resolution can be regarded as a species of abductive reasoning in which a premise is matched with the heads of all clauses with which it can be matched and the abductive inference is then drawn that the matching instantiation of at least one of the bodies of those clauses must hold. Note the obvious connection between this type of inference and the concept of program completion. We start with the query $?HoldsAt(flu, now)$. Applying the axiom in (27), the query reduces to the new query

$$?\mathit{Initiates}(e, flu, t)$$

$$\neg \mathit{Clipped}(t, flu, t')$$

$$\mathit{Happens}(e, t), t < now$$

(27) $\;\;$ ***Happens***$(e, t) \wedge$ ***Initiates***$(e, f, t) \wedge\ t < t' \ \wedge \neg$***Clipped***$(t, f, t') \rightarrow$ ***HoldsAt***(f, t')

The first clause can be resolved, since *Initiates*(e, flu, t) is given. For the second query we have to use a form of resolution for negated queries. This means that we set up a new derivation with the positive query

$$?\ \mathit{Clipped}(t, flu, t').$$

Since we have no matching clauses this query fails and therefore the negated query succeeds (This is the proof–theoretic version of negation as failure.). We are left with the last query

$$?\mathit{Happens}(e, t), t < now.$$

Since we do not have a matching clause for this query $?HoldsAt(flu, now)$, interpreted as query, would fail (finitely). However, $HoldsAt(flu, now)$ interpreted as an integrity constraint leads to an update of the discourse model with the missing clause. In this updated model *HoldsAt(flu, now)* is clearly satisfied. This integrity constraint is written as

$$?HoldsAt(flu, now),\ succeeds$$

A more general description of this procedure is as follows: Given a program P containing the clauses below and an integrity constraint q we want to conclude that q can only be the case because one of the ϕ_i's is the case.

(i) $\;\;$ I have the flu.

Given general world knowledge, these two statments can be said to convey the same information: Anyone who has the flu must have caught it at some earlier time. But in (25) the occurrence of such an event e is an inalienable part of the content, whereas (i) entails it only in conjunction with the relevant piece of world knowledge. When our CLP version of the event calculus is combined with DRT, this difference manifests itself in that the DRS constructed from (25) will contain the flu–catching event already. So if the integrity constraints contributed by (25) are derived from the DRS, the abductive reasoning we are discussing is not needed. More precisely, it will lead to constraints that are already in the given constraint set. This is different for (i). The DRS for (i) only contains a representation of the current state of the speaker. So in this case the abductive process reveals a constraint that is not present yet, and which therefore has to be added to the discourse context with the condition $HoldsAt(flu, now)$.

$$\begin{array}{c}\phi_1 \rightarrow q\\ \phi_2 \rightarrow q\\ \vdots\\ \phi_n \rightarrow q\end{array}$$

This is a strengthened form of closed world reasoning.

A second type of integrity constraint occurs when the top query must fail. This is important for sentences about the past.

(28) Max arrived.

This sentence tells us that Max's arrival was situated entirely in the past, and thus is not going on any more at the present. The positive query

$$?Happens(e,t), t < now$$

expresses just the first part. The second part can only be expressed by the negative constraint, which can be represented as

$$?Happens(e,now), fails$$

Since the resolution process also accepts queries beginning with a negation we can reduce this negative query to the positive query

$$\neg Happens(e,now)$$

Since both positive and negative constraints are admitted and the latter are identified by the term *fails*, it is natural to introduce a similar term to flag the positive queries. We use *succeeds*. So the constraints contributed by (28) can be given as

$$?Happens(e,t), t < \ now, \ \neg Happens(e,now) \ succeeds$$

We will say that an integrity constraint IC is *satisfiable* if it can be made to succeed in case it is positive, and can be made to fail in case it is negative.

3.5 Reification

DRSs will in general contain not only (discourse referents for) events, but also for states. The version of CLP we have presented so far differs in that it has variables for events but not for states. This gap can be filled by expanding our version of CLP with a *reification* component. This component makes it possible to associate a 'res' with each condition. In particular, it will enable us to associate with each condition of the form $HoldsAt(f,t)$ an entity that can be regarded as the state of the fluent f obtaining.[6] The reification procedure is based on a method due to S. Feferman.

We will explain briefly how this works. For this purpose we will enrich the event calculus with a specialization of the theory of truth and abstraction in Feferman (1984).[7]

[6]Reification can be put to many other uses as well, but this is the one for which we need it here.

[7]For the most recent version of this theory see Feferman (2008).

Consider the predicate $burn(x, y, t)$ where t is a parameter for time. Feferman's system allows to form terms from this predicate in two different ways. The first possiblity is to existentially bind t and construct the term $\exists t.burn[x, y, t]$. The square brackets are used here as a notational device to indicate that $\exists t.burn[x, y, t]$ is a term and not a predicate any more. The second possibility is to abstract over the temporal parameter and form the term $burn[x, y, \hat{t}]$. Informally $burn[x, y, \hat{t}]$ should be understood as the set of times at which $burn(x, y, t)$ is true. But note that $burn[x, y, \hat{t}]$ is a term and therefore denotes an object. Feferman's system thus provides two different kinds of structured abstract objects. Intuitively we want to think of $\exists t.burn[x, y, t]$ as the event type corresponding to *x's burning of* y and of $burn[x, y, \hat{t}]$ as the fluent or state corresponding to *x's burning y*.[8] However nothing in the formal set up so far tells us that $\exists t.burn[x, y, t]$ is an event type and $burn[x, y, \hat{t}]$ is a fluent. In order to make sure that $burn[x, y, \hat{t}]$ behaves as a fluent $HoldsAt$ has to be turned into a real truth predicate. The following theorem from Feferman (1984) provides the necessary technical result.

Theorem 1 *Any system that is consistent – in the sense that it has a model – can be extended to a system with truth axioms.*[9] *The extension is conservative over the original system.*

For the special theory under discussion here we need just one truth axiom, which reads as follows:

$$HoldsAt(\phi[\hat{t}], s) \leftrightarrow \phi(s)$$

The specialization for $burn[x, y, \hat{t}]$ therefore is:

$$HoldsAt(burn[x, y, \hat{t}], s) \leftrightarrow burn(x, y, s)$$

This shows that $burn[x, y, \hat{t}]$ behaves like a fluent. Moreover, $\exists t.burn[x, y, t]$ cannot be substituted as an argument of the *HoldsAt*–predicate, but it can be substituted as an argument of the *Happens*–predicate. Hence, with regard to the axioms of the event calculus, abstract terms like $\exists t.burn[x, y, t]$ function as event types and terms like $burn[x, y, \hat{t}]$ as fluents.

To see what this process of reification adds to the representations developed so far, consider again sentence (25), here repeated as (29).

(29) I have caught the flu.

The structure of this sentence was represented by the simple fluent flu in the derivation of Section 3.4. For the purposes of this section this representation was sufficient. However, we would like to have access to the internal structure of sentence (29) as well. For simplicity, we will assume that the personal pronoun *I* is represented by the individual constant *i*. Under this assumption, sentence (29) can be formalized as the structured fluent $flu[i, \hat{t}]$. This representation allows us to have access to the subject of the sentence. We will see in a moment that the possibility to structure fluent and event type objects is an indispensible prerequisite for the transformation of DRSs to integrity constraints.

[8]For an analysis of these different types of English gerunds see van Lambalgen and Hamm (2005), chapter 12.

[9]A model for the event calculus was constructed in (van Lambalgen and Hamm, 2005).

3.6 Event Calculus and DRS

In this section we will outline the connection between DRT and EC with the simplest example from Hamm et al. (2006). Consider again sentence (30).

(30) Max arrived.

The DRS for this sentence is given in (31):

(31)

$m\ t\ e$
$t \prec n$
$e \subseteq t$
e: arrive(m)

Since DRSs introduce existential presuppositions which have to be accommodated, integrity constraints are the appropriate means to represent their inferential potential. First we assume that the constant m and the predicate $arrive(x, t)$ are given. This predicate will be used in its reified form. We use the first possibility for reification and derive the event type $\exists s.arrive[x, s]$.

It has often been observed that the simple past uttered out of the blue is infelicitous. This tense requires that the context provides additional information something like a 'reference time'. We will represent the context here with a new fluent constant f and the clause $\mathit{HoldsAt}(\mathrm{f}, \mathrm{t})$. This constant can then be unified with further contextually given information.

The discourse referent e corresponds to $\exists s.arrive[x, s]$ and the condition *e: arrive(m)* to the clause $\mathit{Happens}(\exists \mathrm{s}.arrive[\mathrm{m}, \mathrm{s}], t)$; n is set to now and t correspond to the context fluent f. In this way, the DRS for sentence (30) is turned into integrity constraint (32).

(32) $?\mathit{HoldsAt}(\mathrm{f}, \mathrm{t}), \mathrm{t}), \mathit{Happens}(\exists \mathrm{s}.arrive[\mathrm{m}, \mathrm{s}], \mathrm{t}), \mathrm{t} < now,$
$\neg \mathit{Happens}(\exists \mathrm{s}.arrive[\mathrm{m}, \mathrm{s}], now),$ *succeeds*

Since in the rest of this paper we will not be concerned with tense, we will simplify integrity constraints as much as possible. First we will drop the clause for the context fluent and the negative integrity constraint. Moreover, we will skip over the internal structure of fluent and events whenever this does not lead to confusion. For instance we will simply write e for $\exists s.arrive[m, s]$. Given these assumptions integrity constraint (32) now reads:

(33) $?\mathrm{t}), \mathit{Happens}(\mathrm{e}, \mathrm{t}), \mathrm{t} < now,$ *succeeds*

This is certainly not completely adequate, but the topics to be discussed in the rest of this paper will not be affected by this simplification.

3.7 Scenarios and Hierarchical Planning

In this section we will start our discussion of more complex examples. The first one is the verb *absperren* and the derived *ung*-nominal *Absperrung* respectively the NP *die Absperrung des Rathauses*. Let us start with the accomplishment verb *absprerren*. According to van Lambalgen and Hamm (2005), every Aktionsart determines a specific 'scenario'.

A scenario should be considered as a local program in contrast to the global program given by the axioms of the event calculus. These local programs provide the additonal information for the Aktionsarten in question, in this case the information specific to accomplishments. In order to formulate this local program we need the following terms in the language of the event calculus.

- $construct$ is an activity fluent.
- $barrier(x)$ is a parameterized fluent indicating the construction state x of the barrier.
- m a real constant indicating the construction stage at which the barrier is considered finished. Thus $barrier(m)$ may be considered the completed object.
- 0 is a real constant indicating the state at which the construction of the barrier starts.
- $start$ is an event initiating constructing.
- finish is the event terminating the constructing activity when the barrier is finished.
- a fluent $accessible(r)$ represententing the state in which the town hall is accessible, where r is a constant denoting the town hall.
- g is a function relating the constructing activity to the construction stage of the barrier. To keep things simple we assume that g is monotone increasing.

These terms allow us to write the following set of clauses as one possible scenario for the accomplishment verb *absperren*.

(34) a. $\mathit{Initially}(\mathit{barrier}(0))$
b. $\mathit{Initially}(\mathit{accessible}(r))$
c. $\mathit{HoldsAt}(\mathit{barrier}(m), t) \wedge \mathit{HoldsAt}(\mathit{construct}, t) \rightarrow$
$\mathit{Happens}(\mathit{finish}, t)$
d. $\mathit{Initiates}(\mathit{start}, \mathit{construct}, t)$
e. $\mathit{Initiates}(\mathit{finish}, \mathit{barrier}(m), t)$
f. $\mathit{Terminates}(\mathit{finish}, \mathit{accessible}(r), t)$
g. $\mathit{Terminates}(\mathit{finish}, \mathit{construct}, t)$
h. $\mathit{HoldsAt}(\mathit{barrier}(x), t) \rightarrow$
$\mathit{Trajectory}(\mathit{construct}, t, \mathit{barrier}(x + g(d)), d)$
i. $\mathit{Releases}(\mathit{start}, \mathit{barrier}(0), t)$

The scenarios for the Aktionsarten are not determined uniquely, but every scenario is required to include information specific to the Aktionsart of the verb under consideration. For the example above, this means that every scenario has to include clauses about the starting and finishing events, about the activity *constructing*, the state *accessible(r)*, and clauses relating this activity to the state of the partial object $barrier(x)$. Together with the axioms of the event calculus these clauses determine inferences triggered by the Aktionsart of *absperren* and the lexical content of this verb.

We are primarily interested in the noun phrase *Absperrung des Rathauses*. We will first concentrate on the event reading; the result state reading will be discussed later.

The first step consists in establishing an event type corresponding to the event reading of *Absperrung des Rathauses*. Using Feferman coding we can transform the predicate $absperren(x, r, t)$ into the abstract event type $a = \exists t.absperr[x, r, t]$. Here r is an individual constant representing the town hall. This is a possible denotation for *Absperrung des Rathauses*, but so far this event type is not related to the verb from which *Absperrung* is derived.

In order to link the nominal to the semantics of the base verb given by its scenario, we introduce an *event definition* by hierarchical planning. The intuitive idea is that hierarchical planning allows to abstract from certain details of the verb's eventuality while maintaining the most important features of the verb's time profile. Formally hierarchical planning is given by program clauses defining an event occurring in the head atom of a clause. We will use the following definition.

Definition 1 *Suppose a scenario for the fluent f is given. In the context of this scenario, the event e is interpreted using f by* hierarchical planning *if* $Happens(start_f, s) \ \wedge \ s < r \ \wedge \ HoldsAt(f, r) \ \ \rightarrow \ Happens(e, r)$

In the special case considered here Definition 1 gives:

$$Happens(start_{construct}, s) \wedge s < r \wedge HoldsAt(construct, r)$$
$$\rightarrow Happens(\exists t.absperr[x, r, t], r)$$

We will simply write a for the event type $\exists t.absperr[x, r, t]$ defined in this way. We thus have a denotation for the event reading of the NP *die Absperrung des Rathauses*. Next, we have to consider the verbal contexts of this NP. The first verb is *behindern* in (35).

(35) Die Absperrung des Rathauses wurde behindert.
The cordoning-off of-the town hall was hampered.

Let us assume that an event type valued function $behindern$ is given. Then we arrive at the following integrity constraint:[10]

(36) $? - Happens(a, t), Happens(behindern(a), t), t < now \ succeeds$

This is certainly too simple. An event type like *behindern* requires its own scenario. We think that for *behindern* to be applied successfully, the activity of cordoning-off must have been initiated and *behindern* supplies the additional information that this activity does not proceed in a smooth way. However, we think that although the activity of cordoning-off is hampered in more or less serious ways, nevertheless the goal – the sealing off of the town hall – will eventually be achieved (non-monotonically).

This changes when one considers our next verb, *verhindern*. In (37) the result state – the town hall being sealed off – is clearly not achieved.

(37) *Die Absperrung des Rathauses wurde verhindert.*
'The cordoning-off of-the town hall was prevented.'

[10]This is a simplification: The scenario for *behindern* plus hierarchical planning triggered by past tense introduces an event type e which has to be unified with a.

This is adequately represented by integrity constraint (38). Since according to (38) *finish* is not allowed to happen, we cannot derive $HoldsAt(barrier(m), s)$ and $\neg HoldsAt(accessible(r), s)$ for some time s.

(38)
$$? - Happens(a, t), Happens(finish, t), t < now, \ fails$$

4 Anaphora resolution

4.1 Reconstructing anaphoric relations

In this section, we will show why anaphora resolution is possible in (39a) and explain why is it blocked in (39b) in a slightly more formal way.

(39) a. *Die Absperrung des Rathauses wurde vorgestern von Demonstranten behindert. Wegen anhaltender Unruhen wird* ***sie*** *auch heute aufrecht erhalten.*
'The cordoning-off of the town hall was disturbed by protesters the day before yesterday. Due to continuing unrest, it is maintained today as well.'

b. *Die Absperrung des Rathauses wurde vorgestern von Demonstranten verhindert. *Wegen anhaltender Unruhen wird* ***sie*** *auch heute aufrecht erhalten.*
'The cordoning-off of the town hall was prevented by protesters the day before yesterday. Due to continuing unrest, it is maintained today as well.'

Clearly, in (39a) the pronoun *sie* in the second sentence refers to the target state of being cordoned off introduced by the first sentence. The impossibility of such an interpretation – this is what "*" is meant to signal – suggests that due to the meaning of the verb *verhindern*, such a target state is not available in (39b).

We will simplify the formalisation as far as possible, concentrating only on what is essential for anaphora resolution. The first sentence of (39a) is represented by integrity constraint (36), i.e. by

$$? - Happens(a, t), Happens(behindern(a), t), t < now \ succeeds$$

The important part of the second sentence is the one containing the verb *aufrecht erhalten* (sustain) and the pronoun *sie*. Choosing a fluent variable s – s is mnemonic for state – and a fluent valued function $aufrecht - erhalten$ we formalise this part as:

$$? - HoldsAt(aufrecht\text{-}erhalten(s), s), s < now, \ succeeds$$

The whole little discourse in (39) is thus represented by the integrity constraint in (40).

(40) $? - Happens(a, t), Happens(behindern(a), t),$
$HoldsAt(aufrecht\text{-}erhalten(s), t), t < now, \ succeeds$

Since *aufrecht-erhalten* requires a state – a special type of fluent – as an argument, s cannot be unified with event type a. This is the formal version of the already explained type mismatch. Therefore it seems that anaphora resolution is blocked in this case.

We will now show that it is nevertheless possible to reconstruct an anaphoric relation by using information contained in the scenario for the verb *absperren*. Since *aufrecht-erhalten* selects the (result) state reading of the NP *die Absperrung der Botschaft* we first have to introduce a denotation for this NP representing this reading. Note that we assume that *behindern* allows – perhaps later as planned – *finish* to happen (non–monotonically). From this we can derive via resolution $\neg \mathit{HoldsAt}(\mathit{accessible}(r), w)$ for some time w. Using Ferferman coding we can reify this formula and obtain the fluent object $\neg \mathit{HoldsAt}[\mathit{accessible}(r), \hat{w}]$. We take this object as denotation of the (result) state reading of the NP *die Absperrung des Rathauses*.[11] Now we can compute the anaphoric relation between the pronoun *sie* and its antecedent *die Absperrung des Rathauses* by unifying s – representing *sie* – with $\neg \mathit{HoldsAt}[\mathit{accessible}(r), \hat{w}]$. Writing *inaccessible* for $\neg \mathit{HoldsAt}[\mathit{accessible}(r), \hat{w}]$ we arrive at the following representation for discourse (39a):

(41) $? - \mathit{Happens}(a, t), \mathit{Happens}(\mathit{behindern}(a), t),$
$\mathit{HoldsAt}(\mathit{aufrecht}\text{-}\mathit{erhalten}(\mathit{inaccessible}), t), t < \mathit{now},\ \mathit{succeeds}$

Summing up, we reconstructed the anaphoric relationship between the pronoun *sie* and and the antecedent NP *die Absperrung des Rathauses* in three steps. First, we derived the formula $\neg HoldsAt(accessible(r), w)$ by resolution using information of the scenario of the verbs *absperren* and *behindern*. Second, we transformed this formula into the term $\neg HoldsAt[accessible(r), \hat{w}] = \ inaccessible$ and third, we unified s with this term. In the minimal model this is the only possibility because there are no other result states, but in richer models there may very well be more than just one result state. In this case, s could be freely unified with these other states, but this would result in a deictic reading for the second sentence of example (39a).

Consider now the mini-discourse in (39b). Combinig integrity constraint (38) with the representation of the second sentence of example (39b) we get integrity constraint (42) for (39b).

(42) $? - Happens(a, t), Happens(finish, t), t < now,\ fails,$
$HoldsAt(aufrecht\text{-}erhalten(s), t), t < now,\ succeeds$

Since this integrity constraint forbids $finish$ to happen for any time t we are no longer in a position to derive $\neg HoldsAt(accessible(r), t)$. But then we cannot unify s with the reified version of $\neg HoldsAt(accessible(r), t)$ and thus the resolution of the pronoun *sie* with the NP *die Absperrung des Rathauses* is correctly blocked.

Note that the possibility to reconstruct the anaphoric relation in (39a) depended on the fact that $\neg HoldsAt(accessible(r), t)$ contains a temporal parameter. This is crucial for our next example involving the object reading of *die Absperrung des Rathauses* – here repeated as (43).

(43) **Die Absperrung wurde heute verstärkt. Sie war am Vortag massiv behindert worden.*
'The barrier was fortified today. It [the cordoning-off] had been massively hampered the day before.'

[11] This is justified in Hamm and Kamp (2009)

In example (43), the pronoun *sie* cannot refer back to *Absperrung*. As mentioned above, this is somewhat surprising for a "lazy" approach. We need only briefly indicate, how we can account for the inacceptability of the sequence in (43).

To fortify a barrier presupposes that a barrier already existed. Let us represent this state of the material object which is established by the cordoning-off activity by the fluent $barrier(m)$ which is contained in the scenario of the verb *absprerren*. This fluent holds after the *finish* event happened. It corresponds to a completed barrier. The denotation for the object reading of the noun *Absperrung* can now be given by (44).

(44) $Absperrung(barrier(m))$

Note that this formula does not contain a temporal parameter. Therefore the three step procedure for reconstructing anaphoric relations introduced above cannot be applied in such cases. This explains why the pronoun *sie* in example (43) cannot refer back to the NP *Die Absperrung*.

4.2 Formal Discourse Semantics

In all classical theories of formal discourse semantics, it was assumed that certain logical operators like negation, disjunction and universal quantification – in contrast to existential quantification and conjunction – block anaphora resolution.[12] These operators were considered as static. For instance, in early DRT the accessibility relation – a geometrical relation on the DRS level – caused discourse referents contained in a negated DRS to be inaccessible. In DPL the semantics of negation as a test did not allow scope extension of the existential quantifier as it did in non–negated sentences. This accounted for the grammaticality distribution in (45).

(45) a. A man walked in the park. He whistled.
b. No man walked in the park. *He whistled.

However, there are cases for which this prediction is too strong:

(46) It is not the case that John does not own a car. It is red and it is parked in front of the house.

For this reason, Groenendijk and Stokhof (1990) introduce a dynamic negation which restores the binding potential of the double negated sentence (45). This kind of negation was improved among other by Dekker (1993).

The following examples due to Bäuerle (1988), however, show that the presence or absence of negation is not the determining factor of anaphora resolution alone. Rather, the interaction of negation with certain types of verbs is crucial. Consider first the examples in (47), which are coherent with the predictions of the early formal discourse theories.

(47) a. *Hans schrieb einen Brief. Das dauerte zwei Stunden.*
Hans wrote a letter. It lasted two hours.
'Hans wrote a letter. This took him two hours.'

[12] In this section we will only consider negation.

b. *Hans schrieb keinen Brief. *Das dauerte zwei Stunden*
Hans wrote no letter. It lasted two hours.
'Hans did not write a letter. This took him two hours.'

Introducing a variation in the second sentence, the following sequences are not in accordance with formal discourse theories.

(48) a. *Hans schrieb einen Brief. Das überraschte uns alle.*
Hans wrote a letter. It surprised us all.
'Hans wrote a letter. We were all surprised by that.'
b. *Hans schrieb keinen Brief. Das überraschte uns alle.*
Hans wrote no letter. It surprised us all.
'Hans did not write a letter. We were all surprised by that.'

We will now show that the proposed formalism allows us to account for this grammaticality distribution as well. Again, we will only give those formal details which are essential for anaphora resolution. Let us first consider the examples in (47). Let e be the event type representing *Hans writing a letter*. The first sentence of (47a) is then formalised as

$$? - Happens(e, t), t < now, succeeds$$

and the second as (with e' as a variable representing the pronoun *das*).

$$?Happens(dauern(e'), t), t < now, t = 2\ hours\ succeeds$$

Together they represent the discourse in (47a).

(49) $? - Happens(e, t), t < now, Happens(dauern(e'), t),$
$t = 2\ hours,\ succeeds$

In the minimal model computed by integrity constraint (49), e' and e will be unified. Thus, *das* refers to the event of *Hans writing a letter*. In non–minimal models, e' may be unified with other event types. This will give the deictic reading again.

The integrity constraint for the first sentence of example (47b) is as in (50):

(50) $?Happens(e, t), t < now, fails$

The integrity constraint for the second sentence is the same as the one for (47a). Integrity constraint (50) computes a model in which there is no event type with the required property, i.e. of Hans writing a letter. Therefore *das* cannot be unified with such an event type. This explains the grammaticality distribution in (47).

We will now consider the examples in (48a). First we have to determine the sort of arguments *überraschen* requires. We will assume here that this verb takes only facts as arguments. In case that *überraschen* turns out to be ambiguous between an event and a fact reading a slightly more involved argument will explain the facts in (48a) too.

The first parts of the sentences in (48a) are of course formalised as above. The second part gives rise to the following integrity constraint:

(51) $? - HoldsAt(surprise(f), t),\ t < now, succeeds$

Here, we are facing a type mismatch again. The variable f cannot be unified with event e provided by the first sentence since e and f belong to different sorts.

However, we can reify $Happens(e, t)$ occurring in the integrity constraint for the first sentence and thereby get: $Happens[e, \hat{t}]$. Intuitively one can consider this term as denoting the fact that event e occurred. Unifying f with this term results in:

(52) $? - HoldsAt(surprise(Happens[e, \hat{t}]), t),\ t < now, succeeds$

This means that the fact that Hans wrote a letter surprised us. Let us now consider example (48b). The integrity constraint for the first sentence is:

$$? - Happens(e, t),\ t < now, fails$$

An integrity constraint fails if and only if its negation succeeds. Therefore, we get the following equivalent constraint

$$? - \neg Happens(e, t)\ t < now, succeeds$$

Applying reification to the $Happen$–part of this constraint we derive the term $\neg Happens[e, \hat{t}]$. Since this is a term of the same sort as f, it is possible to unify f with $\neg Happens[e, \hat{t}]$. The result is:

$$?HoldsAt(surprise(\neg Happens[e, \hat{t}]), t),\ t < now, succeeds$$

The formula says that the fact that Hans didn't write a letter surprised us. This shows that we get the correct results in this case as well.

5 Conclusion and Outlook

In the paper, we argued that disambiguation may be non-monotonic. We discussed examples of anaphora resolution involving a type conflict between anaphora and disambiguated antecedents. Since the anaphora picks up a reading which was discarded for the antecedent, we apply a process of reambiguation to the antecedent to resolve the type mismatch.

Future work needs to address the generality of such maps as the above, both with regard to deverbal nominalisations (for which a mapping from e.g. states to events seems rather awkward) and to other kinds of systematically ambiguous nouns. One case at hand involves the dot objects discussed by Pustejovsky (1997):

(53) *Jonathan Strout hat das Buch [content] geschrieben, es [manifestation] hat 539 Seiten und ist 2004 im Bertelsmann Verlag erschienen.*
'Jonathan Strout wrote the book, it has 539 pages and was published by Bertelsmann.'

Acknowledgements

We would like to thank Hans Kamp, Uwe Reyle, Arndt Riester, Antje Roßdeutscher and the audience at CSSP 2009 in Paris for valuable discussion. The research reported here was supported by the projects B4 and D1 of the Collaborative Research Centre "Incremental Specification in Context" (SFB 732) at the University of Stuttgart.

References

Bäuerle, R. (1988): *Ereignisse und Repräsentationen*, LILOG-Report 43, IBM Deutschland GmbH.

Dekker, P. (1993): Transsentential Meditations. PhD thesis, University of Amsterdam.

Dowty, D. (1991): 'Thematic Proto–Roles and Argument Selection', *Language* **67**, 547–619.

Ehrich, V. and Rapp, I. (2000): 'Sortale Bedeutung und Argumentstruktur: *-ung*-Nominalisierungen im Deutschen', in: *Zeitschrift fr Sprachwissenschaft* **19**: 245–303.

Feferman, S. (1984): 'Toward Useful Type-Free Theories I', *The Journal of Symbolic Logic* **49**, 75–111.

Feferman, S. (2008): 'Axioms for Determinateness and Truth', *The Review of Symbolic Logic* **1**, 204–217.

Groenendijk, J. and M. Stokhof (1990): Dynamic Montague Grammar. *In:* L. Kálmán and L. Pólós, eds, *Papers from the second Symposium on Logic and Language*. Budapest.

Hamm, F. and H. Kamp (2009): Ontology and Inference: The Case of German *ung*–Nominals. Technical report, SFB 732, University of Stuttgart.

Hamm, F., H. Kamp and M. van Lambalgen (2006): 'There is no opposition between formal and cognitive semantics', *Theoretical Linguistics* **32**(1), 1–40.

Hardt, D. (2003): 'Sloppy Identity, Binding, and Centering', in Young and Zhou (eds.), *Proceedings of SALT 13*, Ithaca: CLC.

Kowalski, R. A. and M. Sergot (1986): 'A logic-based calculus of events', *New Generation Computing* **4**, 65–97.

Krifka, M. (1989): *Nominalreferenz und Zeitkonstitution*. Fink, München.

Lifschitz, V. (1994): Circumscription. *In:* D. Gabbay, C. Hogger and J. Robinson, eds, *Handbook of Logic in Artificial Intelligence and Logic Programming: Vol 3*. Clarendon Press, Oxford.

McCarthy, J. and P. Hayes (1969): Some Philosophical Problems from the Standpoint of Artificial Intelligence. *In:* D. Michie and B. Meltzer, eds, *Machine Intelligence 4*. Edinburgh University Press, Edinburgh.

Nienhuys-Cheng, S-H. and R. de Wolf (1997): *Foundations of Inductive Logic Programming*. Springer, New York, Berlin.

Poesio, M. (1996): ‘Semantic Ambiguity and Perceived Ambiguity’, in v. Deemter & Peters (eds.): *Semantic Ambiguity and Underspecification*. Stanford: CSLI, 159201

Pustejovsky, J. (1995): *The Generative Lexicon*. Cambridge, Massachusetts: MIT Press.

Reyle, U. (1993): ‘Dealing with Ambiguities by Underspecification: Construction, Representation and Deduction’, *Journal of Semantics* **10**(2): 123–179.

Rodeutscher, A. & Kamp, H. (2010): ‘Syntactic and Semantic Constraints in the Formation and Interpretation of ung-Nouns’, in: Alexiadou & Rathert (eds.), *Nominalisations across Languages and Frameworks*, Berlin: Mouton de Gruyter.

Shanahan, M. (1997): *Solving the Frame Problem – A Mathematical Investigation of the Common Sense Law of Inertia*. MIT Press, Cambridge, MA.

van Lambalgen, M. and F. Hamm (2005): *The Proper Treatment of Events*. Blackwell, Malden.