Empirical Issues in Syntax and Semantics 8
O. Bonami & P. Cabredo Hofherr (eds.) 2011, pp. 167–194
http://www.cssp.cnrs.fr/eiss8

Reambiguating: on the non-monotonicity of disambiguation

Fritz Hamm and Torgrim Solstad*

1 Introduction

The relation between lexical ambiguity and disambiguation is mostly approached from an intra-sentential perspective. Thus, when analyzing the ambiguity and disambiguation of a lexical item, one tends to study its variance in interpretation when it is modified by or occurs as an argument of other lexical items. Broadening this perspective, this paper shows that there are important insights into the nature of disambiguation to be gained by studying more closely how ambiguous expressions behave in contexts spanning more than one sentence. More specifically, we introduce new data involving anaphora resolution with the following two characteristics: (i) a potentially ambiguous antecedent which is disambiguated in its local context, and (ii) anaphora which refer to one of the possible readings of the antecedent which was not selected in the local antecedent context. We argue that these data call for a revision of how we conceive of and formalize the process of disambiguation, introducing the notion of *reambiguation*, which characterizes the process of reintroducing alternative interpretations which were originally excluded by disambiguation.

The paper is structured as follows. In Section 2, we discuss properties of ambiguity and disambiguation. We also give 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 some consequences of our approach for formal discourse semantics in general. Section 5 concludes the paper.

2 Ambiguity, Disambiguation and Underspecification

Formally, ambiguities are often represented by means of underspecification. In computational linguistics and formal semantics alike, underspecification is 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

*We would like to thank Hans Kamp, Uwe Reyle, Arndt Riester, Antje Roßdeutscher, Henriette Slogsnat, the anonymous reviewers and the CSSP 2009 audience for valuable comments and 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.

place, as is the case when the information present does not allow an informed decision as to which interpretation to choose, the underspecified representation allows interpretational decisions to be deferred to a later point at which such information may become available. We focus on another aspect of underspecification, however, namely its role in the relation between ambiguous and disambiguated expressions. In general, it is assumed that semantic information included in an underspecified representation is discarded and not retrievable when interpretational decisions resulting in disambiguation are made. Contrary to this, we argue that this view is inadequate for the phenomena of anaphora resolution which we analyze in this paper: Still assuming disambiguation to involve the discarding of information in underspecified representations, we allow the result of disambiguation to be reversed or altered in subsequent discourse under certain conditions.

Underspecified representations of lexical (or scopal) ambiguities typically involve some kind of disjunction (Reyle, 1993) or conjunction (Poesio, 1996). In the Underspecified Discourse Representation Theory 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) $$\left\langle \alpha \; \begin{array}{|c|} \hline \\ \hline \alpha = \mathrm{e} \overset{!}{\vee} \alpha = \mathrm{y} \\ \mathrm{e: deliver(x,y)} \\ \textsc{agent}(\mathrm{e})=\mathrm{x} \\ \textsc{theme}(\mathrm{e})=\mathrm{y} \\ \hline \end{array} \right\rangle$$

More specifically, the representation in (1) shows the semantic representation for *delivery* at NP level, stating that the event of delivering involves an agent and a theme argument. Importantly, α in (1) 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 the 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 such underspecified ambiguous expressions, disambiguation is naturally 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 top-most condition in (1).[1]

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

In (2), *damaged* is assumed to combine only with the object reading of *delivery*, whereas *quick* selects only the event reading. As mentioned briefly above, the status of the deleted disjunct(s) or conjunct(s) will be of main interest in this paper. We also assume that the disambiguation of underspecified expressions leads to disjunct or conjunct deletion. However, we argue that the information contained in the deleted conjunct should be retrievable under certain conditions.

Our data mainly involve German deverbal nominalizations. More specifically, we present a study of nouns derived by means of the suffix *-ung* (comparable both to *-tion*

[1] Similar remarks may be made with regard to Poesio's (1996) disambiguation inference mechanism.

and *-ing* nominalizations in English, cf. Ehrich and Rapp, 2000; Rossdeutscher and Kamp, 2010). While all productively derived *-ung* nouns have an event reading, quite a few *-ung* derivations additionally have result state and/or object readings, cf. *Absperrung* (from *absperren* 'cordon off', 'block') in (3), 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.'

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') (3-a) is assumed to select for object interpretations, *aufrecht erhalten* ('sustain') (3-b) for states and *behindern* ('hamper') (3-c) for event interpretations (for details see Hamm and Kamp, 2009). A simplified, underspecified semantic representation covering all three readings is provided in (4):

(4) ⟨α

z
$\alpha = \mathrm{e} \stackrel{!}{\vee} \alpha = \mathrm{s} \stackrel{!}{\vee} \alpha = \mathrm{y}$
e CAUSE s
s: HAVE(y,z)
FUNCTION_AS_BARRIER(y)
AGENT(e)=x

⟩

Briefly stated, *Absperrung* involves an event e causing 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 disambiguation, the reader is referred to Hamm and Kamp (2009).

Taking the above considerations of Reyle (1993) or Poesio (1996) as a starting point, there is nothing special about how the disambiguating contexts in (3) influence the possible referential arguments of the DPs headed by *Absperrung* ('cordoning-off', 'barrier'). However, we present data (the naturalness of which has been confirmed by numerous native speakers) which are highly problematic for a naive disambiguation-as-disjunct-deletion approach as described above. These data involve two-sentence sequences where a potentially ambiguous deverbal nominalization is disambiguated in the first sentence. The second sentence contains a pronoun which is clearly co-referential with the DP headed by the deverbal nominalization. However, due to sortal restrictions in its local context, this pronoun can only pick up a reading which was *not* selected for in the first disambiguating sentence, cf. the sequence in (5):

(5) *Die Absperrung des Rathauses wurde vorgestern von*
the cordoning-off the town hall was the day before yesterday by
*Demonstranten behindert. Wegen anhaltender Unruhen wird **sie** auch heute*
protesters hampered. Due to continuing unrest is it also today
aufrecht erhalten.
sustained.
'The cordoning-off of the town hall was hampered 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', literally: 'she') is co-referential with the noun phrase headed by *Absperrung* in the first sentence. As stated in the discussion of example (3-c), the predicate *behindern* ('hamper') restricts the ambiguity of *Die Absperrung des Rathauses* and fixes an event reading of the noun phrase. However, recall that the matrix predicate in the second sentence, *aufrecht erhalten* ('sustain'), only allows the referential argument of the anaphora *sie* ('it') 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* ('it') to pick up, contrary to intuitions. Given the naturalness of the sequence in (5), we contend that the disambiguation-as-deletion view must be revised. The mechanism of *reambiguation* which we propose accounts adequately for data such as (5) by allowing the restricted recovery of information which has been discarded as a result of disambiguation.

Attempting to pre-empt some of the most obvious arguments against granting examples such as (5) any special status, let us discuss briefly (i) a "lazy" approach, and (ii) the option of coercion, which have both been suggested to us in discussion. What we refer to as a "lazy" approach attempts to avoid the problem by assuming that disambiguation does not involve any deletion whatsoever. We contend that this is no option, as it would predict that every possible discourse referent of a noun is always available in subsequent discourse. The following unacceptable example (indicated by the '#' sign), which will be discussed later, shows that this is not the case. It crucially involves a 'physical object antecedent' and an anaphora of event type:

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

Concerning the second option of *coercion* (or rather *reinterpretation* in the terms of Egg, 2005), this is a more intricate issue, which we can only touch upon in this paper.[2] Obviously, coercion would in principle always be applicable, as there are basically no restrictions to the mechanism of coercion given a sufficient complexity of types, which

[2]The notion of *coercion* as introduced by Moens and Steedman (1988) was originally restricted to aspectual phenomena. Later, it has been widened to include a number of other phenomena such as e.g. sortal shifts (Dölling, 2003) in the nominal domain (cf. e.g. the work of Egg, 2005).

is problematic in itself. As we will see below (see the discussion of examples (38-a)-(38-b) in Section 4), establishing proper restrictions for the acceptability of anaphoric relations such as the one in (5) is beyond the scope of simple type conflict resolution generally considered in formalizations of coercion. A more general argument against such an approach is that coercion, also in broader terms of sortal shifts, is taken to be a locally restricted phenomenon involving predicate-argument or modifier-head relations for which a (sortal) type conflict may be observed. In the case of (5) it is not all that clear what should initiate the process of coercion in the first place as there are no local type conflicts involved. Both the semantics of the DP headed by the nominalization and also obviously that of the pronoun satisfy the sortal restrictions of the arguments selecting for them locally. Of course, there is a type conflict involving the anaphora and its antecedent in (5), but we contend that applying coercion is not an appropriate way to deal with such phenomena. Rather, sequences such as (5) provide counter-examples to generally accepted assumptions in formal-semantic theories of anaphora resolution (cf. e.g. van Eijck and Kamp, 1997), which assume type identity between anaphora and their antecedents. Crucially, the solution we propose for dealing with the type conflict in (5) also has interesting, more general consequences for (formal) discourse semantics.

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* ('it') makes reference to. We show that such a reconstruction is possible even under the assumption that *behindern* ('hamper') 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* ('it') 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* ('it') in (5), one can certainly assume that the discourse referent of the anaphor is allowed to be identified with the referent of the DP *die Absperrung des Rathauses*, also based on the morpho-syntactic constraints on referential identification for the discourse referent introduced by the singular feminine pronoun *sie* ('it'): Gender features exclude the referential argument of the neuter noun *Rathaus* ('town hall') and number features excludes the referential argument of the plural *Demonstranten* ('protesters'). 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* ('cordon off'), involves an object (y), which is incrementally constructed in order to block access to a region (z), i.e. the agent (x) of the event (e) causes a state (s) of inaccessibility of the region (z).

- The referential argument of the relevant 'anaphora theme argument' of the predicate *aufrecht erhalten* ('sustain') is of result state type, while the 'antecedent theme argument' of *behindern* ('hamper') is of event type.

- The properties of the pronoun *sie* ('it') – 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* ('the cordoning-off of the town hall') to the result state of being cordoned off. This state is accessible via the semantics of the predicate *absperren* ('cordon off'). The mapping from the event to the state consists in an abstraction over the times for which the predicate holds (from $absperr(e,t)$ to the reified $absperr[e,\hat{t}]$). This set of times can in principle be both the one 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.

As mentioned above, the proposed formalization allows us to account adequately for cases where the application of coercion would offer no obvious solution. This is for instance the case with the non-monotonic inference which is triggered by *behindern* ('hamper') and blocked by *verhindern* ('prevent'), respectively (cf. Hamm and Kamp, 2009). Making reference to this inferential variance enables us to explain the difference concerning the possibility of anaphora resolution in (5) versus (7).

(7) #*Die Absperrung des Rathauses wurde vorgestern von*
the cordoning-off the town hall was the day before yesterday by
Demonstranten verhindert. Wegen anhaltender Unruhen wird ***sie*** *auch heute*
protesters prevented. Due to continuing unrest is it also today
aufrecht erhalten.
sustained.
'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.'

In (7), anaphora resolution fails because the above-mentioned non-monotonic inference that the activity of cordoning-off leads to a result state of being cordoned off is blocked by *verhindern* ('prevent'). Note that from the perspective of coercion, it is hard to differentiate the two cases, since they both involve antecedents with referential arguments of event type.

Finally, the problematic case in (6) discussed in connection with the "lazy approach" is accounted for under the assumption that physical objects are represented by predicates without temporal parameters. In this case, anaphora resolution is blocked correctly, since the above depicted three-step procedure involving inference, reification and unification is not applicable for predicates without temporal parameters.

Concerning the notion of reambiguation, it should be noted that reambiguation may involve a complete recovery of all readings which were deleted in the preceding context, and not just shifting to a different one, as in (5). Consider (8), where *ignorieren* ('ignore') allows *sie* ('it') 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:

(8)

Die	*Absperrung*	*des Rathauses*	*wurde*	*von*	*Demonstranten*	*behindert.*	*Später*
the	cordoning-off	the town hall	was	by	protesters	hampered.	Later

haben	*sie*	*alle*	*ignoriert.*
have	it	everyone	ignored.

'The cordoning-off of the town hall was hampered by protesters. Later, everyone ignored it.'

In the next section, we present the most important theoretical prerequisites for the formal analysis alluded to in the informal description above. It will involve a coupling of Discourse Representation Theory (DRT; Kamp and Reyle, 1993) with Constraint Logic Programming (CLP; van Lambalgen and Hamm, 2005).

3 Event Calculus

Crucially, our approached is based on Constraint Logic Programming. However, before we start to develop integrity constraints and programs for the examples discussed 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 originated in Artificial Intelligence and was used for high level control of mobile robots (see McCarthy and Hayes, 1969; Kowalski and Sergot, 1986; Shanahan, 1997). In van Lambalgen and Hamm (2005) the event calculus is formalized as a (constraint) logic program with the aim to represent planning. The motivation for logic programming as an adequate tool for planning is as follows: 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 consider a typical clause of a propositional logic program, say

$$p_1, \ldots, p_n \rightarrow q.$$

In this clause, one may think of q as a goal to be achieved if conditions $p_1, \ldots, p_n$ are satisfied. This accounts for the basic intuition concerning planning as well as for the recursive character of planning, since the conditions $p_1, \ldots, p_n$ could be given as subgoals as well:

$$r_{i_1}, \ldots, r_{i_m} \rightarrow p_i.$$

Moreover, logic programing nicely captures the crucial non–monotonicity of planning. 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 and logic programming is such a formalism.

The 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.[3] Let us now consider a specific example.

3.1 Linguistic Motivation

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

In (9), 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:

(10) 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* may be assumed to hold initially. Similarly no terminating event is mentioned. Thus, *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.

Before we proceed to describe the event calculus a bit more formally, we will first outline a kind of roadmap for the formalism as a whole, since this type of formal system is rather unusual in linguistic semantics. We will also indicate which part of the combined system is used for the derivation of anaphora resolution in the examples discussed so far. The combined system consists of the event calculus as a logic program which, however, is confined to provide only universal information. For the introduction of existential information the calculus is therefore combined with a tool from data base theory – integrity constraints, which also allow to give a precise formulation of the above slogan saying that a sentence S is to be considered as a goal (make S true) to be achieved by updating the discourse model. The last component of the formalism is a theory of reification which allows to turn predicates into terms. This is crucial for the second step of the three-step procedure (inference, reification and unification) of computing the anaphoric link in example (5).

[3] Van Lambalgen and Hamm (2005) argue for a close connection between planning and tense. The justification of this claim is however beyond the scope of this paper. The interested reader is therefore referred to van Lambalgen and Hamm (2005).

The combined system

Event calculus	Integrity constraints	Reification
Inference	update of discourse models	maps predicates to terms

We will now 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.[4] 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.

(11) *Initially*(f)

(12) *Happens*(e, t)

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

(14) *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 (a time-dependent property) 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.

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

(16) *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

[4]The term *fluent* was coined by Newton for functions with a temporal parameter.

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 continuously 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.

(17) $Clipped(t_1, f, t_2)$

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

(18) $HoldsAt(f, t)$

'*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 (10).

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.

(19) 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 some t' such that $t < t'$, 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 (10), this axiom was used 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*.[5] 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 (19) is given by the combination of the following two axioms:

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

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 conjunction of (negated and non–negated) formulas. This conjunction is composed of predicates of the event calculus and temporal information such as $t < t'$ which are interpreted in the structure of the reals, i.e. in $(\mathbb{R}, 0, 1, +, \cdot, <)$. These are the constraints of the event calculus considered as a 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).

(20) a. *Prof(confucius)* (Confucius is a professor.)
b. *Prof(socrates)* (Socrates is a professor.)
c. ¬ *Prof(y)* → *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 (20) states about the predicate *professor* that *confucius* belongs to its extension (20-a) and also that *socrates* belongs to its extension (20-b); 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 $x = \mathit{confucius}$ and $x = \mathit{socrates}$, where x is a new variable, which intuitively plays the role of an arbitrary

[5] 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.

member of the extension of *professor*, and making this disjunction into the antecedent of the following implication:

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

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 $socrates \neq confucius \neq plato$.[6] Then (21) implies that *plato* is not a professor. Now consider the third clause of program (20). A similar procedure applied to this clause yields:

(22) $\forall x(Student(x) \leftrightarrow \neg Prof(x))$[7]

Formula (22) implies that Plato is a student. The conjunction of (21) and (22) is the completion of program (20). 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 (20) that bring additional entities into play. Suppose for instance that we add to (20) the fact *beard(plato)*, which states that Plato has a beard. A minimal model for the completion of the extended program will have as a universe $\{ confucius, socrates, plato \}$. In this model Plato is not a professor, but the only student and the only individual 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 given by the following program:

(23) 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 (23). The completion of the program is given by

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

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

[7] 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 (22) are equivalent.

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. Program (23) could easily be enrichted with the 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 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.

(24) 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 (24). For this purpose, assume that a discourse model is given as a collection of facts concerning events and fluents and assume that sentence (24) 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. Furthermore, no terminating event has been mentioned. Therefore we conclude by closed world reasoning that no such event occurred. Consider again axiom (19) repeated here as (25).

(25) 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 some t' such that $t < t'$, then f holds at t'

According to this axiom there is only one fact missing in order to establish the truth of $HoldsAt(flu, now)$. 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 $HoldsAt(flu, now)$.

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 (26), this query reduces to the new query

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

$$\neg Clipped(t, flu, t')$$

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

(26) $Happens(e, t) \land Initiates(e, f, t) \land \; t < t' \; \land \neg Clipped(t, f, t') \rightarrow HoldsAt(f, t')$

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

$$?\, Clipped(t, flu, t').$$

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

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

Since we do not have a matching clause for this query $?HoldsAt(flu, now)$, interpreted as a 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

$$?HoldsAt(flu, now)$$

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.

$$\phi_1 \to q$$

$$\phi_2 \to q$$

$$\vdots$$

$$\phi_n \to q$$

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.

(27) 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 is represented as

$$?Happens(e, now), \texttt{fails}$$

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 (27) can be given as

$$?Happens(e,t), t < \; now, \; \neg Happens(e, now), \texttt{succeeds}$$

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

In this section we will extend Constraint Logic Programming (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 formula of the form $HoldsAt(f, t)$ an entity that can be regarded as the state of the fluent f obtaining.[8] 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).[9]

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*.[10] 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.*[11] *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 (24), here repeated as (28).

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

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

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

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

(28) 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 (28) as well. For simplicity, we will assume that the personal pronoun *I* is represented by the individual constant *i*. Under this assumption, sentence (28) 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.

3.6 Event Calculus and DRT

In this section we will outline the connection between Discourse Representation Theory and the Event Calculus with the simplest example from Hamm et al. (2006). Consider again sentence (29).

(29) Max arrived.

The DRS for this sentence is given in (30):

(30)

$m\ t\ e$
$t < n$ $e \subseteq t$ 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 $HoldsAt(\mathrm{f},\mathrm{t})$. 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 $Happens(\exists \mathrm{s}.arrive[\mathrm{m},\mathrm{s}],t)$; *n* is set to *now* and *t* correspond to the context fluent f. In this way, the DRS for sentence (29) is turned into integrity constraint (31).

(31) $?HoldsAt((\mathrm{f},\mathrm{t}),\mathrm{t}), Happens(\exists \mathrm{s}.arrive[\mathrm{m},\mathrm{s}],\mathrm{t}), \mathrm{t} < now,$
$\neg Happens(\exists \mathrm{s}.arrive[\mathrm{m},\mathrm{s}], now)$, `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 ignore 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 (31) now reads:

(32) $?Happens(\mathrm{e},\mathrm{t}), \mathrm{t} < now$, `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* ('cordon off') and the derived *ung*-nominal *Absperrung* ('cordoning-off', 'barrier') respectively the NP *die Absperrung des Rathauses* ('the cordoning-off of the town hall'). Let us start with the accomplishment verb *absperren* ('cordon off'). 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* ('cordon off').

(33) a. $Initially(barrier(0))$
b. $Initially(accessible(r))$
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. $Terminates(finish, accessible(r), t)$
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(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* ('cordon off') and the lexical content of this verb.

We are primarily interested in the NP *Absperrung des Rathauses* ('cordoning-off of the town hall'). 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 $a = \exists t.absperr[x, r, t]$, in which r is an individual constant representing the town hall. This is a possible denotation for *Absperrung des Rathauses* ('cordoning-off of the town hall'), 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 defined by* hierarchical planning *using f if the following holds:*

$$Happens(start_f, s) \wedge s < w \wedge HoldsAt(f, w) \rightarrow Happens(e, w)$$

In the special case considered here Definition 1 gives:

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

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* ('the cordoning-off of the town hall'). Next, we have to consider the verbal contexts of this NP. The first verb is *behindern* ('hamper') in (34).

(34) *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* ('hamper') is given. Then we arrive at the following integrity constraint:[12]

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

This is certainly too simple. An event type like *behindern* ('hamper') requires its own scenario. We think that for *behindern* ('hamper') to be applied successfully, the activity of cordoning-off must have been initiated and *behindern* ('hamper') 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* ('prevent'). In (36) the result state – the town hall being cordoned off – is clearly not achieved.

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

This is adequately represented by integrity constraint (37). Since according to (37) *finish* is not allowed to happen, we cannot derive $HoldsAt(barrier(m),s)$ and $\neg HoldsAt(accessible(r),s)$ for some time s.

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

4 Anaphora resolution

In this Section, we first show how the above theoretical considerations apply to the crucial example (5) in Section 2 (to be repeated below). Next, we go on to point at some consequences of our approach for formal discourse semantics in general.

4.1 Reconstructing anaphoric relations

In this section, we will show why anaphora resolution is possible in (38-a) and explain why is it blocked in (38-b) in a slightly more formal way.

(38) 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 hampered 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.'

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

Clearly, in (38-a) the pronoun *sie* ('it') in the second sentence refers to the target state of being cordoned-off which may be inferred from 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* ('prevent'), such a target state is not available in (38-b).

We will simplify the formalisation as far as possible, concentrating only on what is essential for anaphora resolution. The first sentence of (38-a) is represented by integrity constraint (35), i.e. by

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

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

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

The whole little discourse in (38) is thus represented by the integrity constraint in (39).

(39) $?-Happens(a,t), Happens(behindern(a),t), HoldsAt(aufrecht\text{-}erhalten(s),t),$
$t < now$, 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* ('cordon off'). Since *aufrecht-erhalten* selects the (result) state reading of the NP *die Absperrung der Botschaft* ('the cordoning-off of the town hall') we first have to introduce a denotation for this NP that represents this reading. Note that we assume that *behindern* ('hamper') allows – perhaps later than planned – *finish* to happen (non-monotonically). From this we can derive via resolution $\neg HoldsAt(accessible(r), w)$ for some time w. Using Ferferman coding we can reify this formula and obtain the fluent object $\neg HoldsAt[accessible(r), \hat{w}]$. We take this object as the denotation of the (result) state reading of the NP *die Absperrung des Rathauses.*[13] Now we can compute the anaphoric relation between the pronoun *sie* ('it') and its antecedent *die Absperrung des Rathauses* ('the cordoning-off of the town hall') by unifying s – representing *sie* ('it') – with $\neg HoldsAt[accessible(r), \hat{w}]$. Writing *inaccessible* for $\neg HoldsAt[accessible(r), \hat{w}]$ we arrive at the following representation for discourse (38-a):

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

(40) $? - Happens(a, t), Happens(behindern(a), t),$
$HoldsAt(aufrecht\text{-}erhalten(inaccessible), t), t < now,$ `succeeds`[14]

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 from the scenarios 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 (38-a).

Consider now the mini-discourse in (38-b), where the only difference from (38-a) is that *behindern* ('hamper') in (38-a) has been replaced by *verhindern* ('prevent'). Combining integrity constraint (37) with the representation of the second sentence of example (38-b), we get integrity constraint (41) for (38-b).

(41) $? - Happens(a, t), Happens(finish, t), t < now,$ `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* ('it') with the NP *die Absperrung des Rathauses* is correctly blocked. As mentioned in Section 2, is it hard to see how applying coercion could account for the difference between (38-a) and (38-b), given that *behindern* ('hamper') and *verhindern* ('prevent') both select for arguments of the same (event) type.

Note that the possibility to reconstruct the anaphoric relation in (38-a) depends 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* – repeated here as (42).

(42) *#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.'

In example (42), the pronoun *sie* ('it') cannot refer back to *Absperrung* ('barrier'). As mentioned in Section 2, this is somewhat surprising for a "lazy" approach, in which disambiguation does not involve conjunct or disjunct deletion of underspecified representations. We will only briefly indicate how we can account for the inacceptability

[14] A more realistic constraint would be:

$$? - Happens(a, t), Happens(behindern(a), t),$$

$$HoldsAt(aufrecht\text{-}erhalten(inaccessible), t'), t < t' < now, \texttt{succeeds}$$

which requires that the state *inaccessible* temporally succeeds the *disturb* event. The derivation of the temporal ordering of eventualities is however beyond the scope of this paper. The interested reader is adviced to consult van Lambalgen and Hamm (2005), in particular chapter 9.

of the sequence in (42).

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 means of the fluent *barrier*(*m*) which is contained in the scenario of the verb *absperren* ('cordon off'). This fluent holds after the *finish* event happened. It corresponds to a completed barrier. The denotation for the object reading of the noun *Absperrung* ('barrier') can now be given by (43).

(43) *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 result state pronoun *sie* ('it') in example (42) cannot refer back to the DP *die Absperrung* ('the barrier').

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.[15] 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 Dynamic Predicate Logic, 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 (44).

(44) 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:

(45) 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 (44). This kind of negation was improved among others by Dekker (1993).

The following examples due to Rainer Bäuerle (1988), however, show that the presence or absence of negation is not the only factor determining anaphora resolution. Rather, the interaction of negation with certain types of verbs is crucial. Consider first the examples in (46), which are coherent with the predictions of the early formal discourse theories.

(46) 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.'

[15] 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.'

A variation of the second sentence, however, shows that this is in general not correct.

(47) 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 (46). Let *e* be the event type representing *Hans writing a letter*. The first sentence of (46-a) is then formalised as

$$?-\mathit{Happens}(e,t), t<\mathit{now}, \texttt{succeeds}$$

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

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

Together they represent the discourse in (46-a).

(48) $?-\mathit{Happens}(e,t), t<\mathit{now}, \mathit{Happens}(\mathit{dauern}(e'),t),$
$t=2\ \mathit{hours},\ \texttt{succeeds}$[16]

In the minimal model computed by integrity constraint (48), e' and e will be unified. Thus, *das* ('it') 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 (46-b) is given as in (49):

(49) $?\mathit{Happens}(e,t), t<\mathit{now}, \texttt{fails}$

The integrity constraint for the second sentence is the same as the one for (46-a). Integrity constraint (49) computes a model in which there is no event type with the required property, i.e. of Hans writing a letter. Therefore, *das* ('it') cannot be unified with such an event type. This explains the grammaticality distribution in (46).

We will now consider the examples in (47-a). First we have to determine the sort of arguments *überraschen* ('surprise') requires. We will assume here that this verb takes only facts as arguments. In case that *überraschen* ('surprise') turns out to be ambiguous between an event and a fact reading, a slightly more involved argument will explain the facts in (47-a) too.

The first parts of the sentences in (47-a) are of course formalised as above. The second part gives rise to the following integrity constraint:

[16]The same proviso as in footnote 14 concerning the derivation of the temporal ordering of eventualities applies here as well.

(50) $? - HoldsAt(surprise(f), t),\ t < now,$ `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 the predicate $Happens(e, t)$ 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:

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

This means that the fact that Hans wrote a letter surprised us. Let us now consider example (47-b). The integrity constraint for the first sentence is:

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

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

$$? - \neg Happens(e, t)\ t < now, \texttt{succeeds}$$

Applying reification to the *Happens*-part of this constraint we can 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:

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

The formula says that the fact that Hans didn't write a letter surprised us. This shows that we get the correct results for the Bäuerle examples in a completely systematic way too.

5 Conclusion and Outlook

We argued that disambiguation may be non-monotonic in nature. 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 reconstruction to the antecedent to resolve the type mismatch. We refer to this process as *reambiguation*.

Future work needs to address the generality and complexity of such reconstruction processes. For instance, we argued that the resolution of the anaphora in example (38) is achieved by a more complex computations than those involved in the analysis of the examples in Section 4.2. However, (38) is certainly not the most complicated case one has to face. Although the reconstruction process for the following example is beyond the scope of this paper, we will nevertheless sketch a possible analysis in an informal way.

(52) *Auf Gemarkung Schönau bei Heidelberg wurde ein toter Fuchs gefunden,*
in district Schönau at Heidelberg was a dead fox found,
der Tollwut hatte. Deswegen wurde der Bereich nördlich des Neckars
which rabies had. For this reason was the area north of the Neckar
östlich der Bundesstraße zum wildtollwutgefährdeten Bezirk erklärt.
east of the federal highway to the wild-rabies-endangered area declared.
'In the district of Schönau a fox was found which had died from canine madness. For this reason, the territory which is north of the Neckar and to the east of the federal highway was declared a wildlife rabies high-risk area.'

The discourse particle *deswegen* (for this reason) introduces a *causal*[17] anaphoric relation between the first and the second sentence. Example (52) is informative about the effect of the cause – namely the declaration of the territory north of the Neckar and east of the federal highway as a wildlife rabies high-risk area – but is rather vage concerning the reason for this effect.

Let us now assume that *deswegen* introduces a causal relation $cause(\phi, \psi)$ where ψ (the effect) is given. Then an appropriate integrity constraint should trigger an abductive reasoning process which reconstructs the cause of the given effect. This is similar to the examples considered in the body of the text. But in the case of sentence (52) an additional complication is involved. Given only (52), the cause and therefore the anaphoric relation to be reconstructed is not unique. Many facts are possible causes for ψ in this case; for instance *that a dead fox was found* or *that a dead fox which had rabies* was found or *that a dead fox which had rabies was found in the district of Schönau.* Of course further context may rule out some of these possibilities but sentence (52) is rather uninformative in this regard. Therefore, a formally precise analysis of such examples requires techniques which are beyond those introduced in this paper.

A further generalization of the approach to anaphora resolution argued for in this paper necessitates maps which correspond to dot objects discussed by Pustejovsky (1995):

(53) *Jonathan Strout hat das Buch geschrieben, es hat 539 Seiten und ist 2004 im*
Jonathan Strout has the book written, it has 539 pages and is 2004 in the
Bertelsmann Verlag erschienen.
Bertelsmann publishing house appeared
'Jonathan Strout wrote the book, it has 539 pages and was published by Bertelsmann.'

In order to resolve the anaphora *es* ('it') in example (53) a function mapping the content denotation of *Buch* ('book') to the physical manifestation reading of this noun is required.

[17] *Deswegen* is composed of the anaphoric element *des-* and the (factively) causal preposition *wegen*. For an extensive investigation of causality expressed by means of prepositional phrases (exemplified by the German preposition *durch*) the reader is referred to Solstad (2007). In Solstad (2010) a DRT analysis of the factively causal *because of* is presented, which is by and large equivalent to its German counterpart *wegen*.

References

Bäuerle, R., 1988. *Ereignisse und Repräsentationen.* Tech. rep., LILOG-Report 43.

Dekker, P., 1993. *Transsentential Meditations.* Ph.D. thesis, University of Amsterdam.

Dölling, J., 2003. Flexibility in adverbial modification: reinterpretation as contextual enrichment. In Lang, Ewald, Claudia Maienborn, and Cathrine Fabricius-Hansen (eds.), *Modifying Adjuncts*, pp. 511–552. Berlin/New York: Mouton de Gruyter.

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

Egg, M., 2005. *Flexible Semantics for Reinterpretation Phenomena.* Stanford: CSLI Publications.

Ehrich, V. and I. Rapp, 2000. Sortale Bedeutung und Argumentstruktur: *ung*-Nominalisierungen im Deutschen. *Zeitschrift für Sprachwissenschaft*, 19:245–303.

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

———, 2008. Axioms for determinateness and truth. *The Review of Symbolic Logic*, 1:204–217.

Groenendijk, J. and M. Stokhof, 1990. Dynamic montague grammar. In Kálmán, L. 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. In Roßdeutscher, A. (ed.), *Disambiguation and Reambiguation*, vol. 6 of *SinSpec – Working Papers of the SFB 732 "Incremental Specification in Context"*, pp. 1–67. Stuttgart: SFB 732.

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

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

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.* München: Fink.

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

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

Moens, M. and M. Steedman, 1988. Temporal ontology and temporal reference. *Computational Linguistics*, 14:15–28.

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

Poesio, M., 1996. Semantics ambiguity and perceived ambiguity. In van Deemter, K. and S. Peters (eds.), *Semantics Ambiguity and Underspecification*, pp. 159–201. Stanford: CSLI.

Pustejovsky, J., 1995. *The Generative Lexicon.* Cambridge: MIT Press.

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

Rossdeutscher, A. and H. Kamp, 2010. Syntactic and semantic constraints in the formation and interpretation of *ung*-nouns. In Alexiadou, Artemis and Monika Rathert (eds.), *The Semantics of Nominalizations 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.* Cambridge, MA.: MIT Press.

Solstad, T., 2007. *Mehrdeutigkeit und Kontexteinfluss.* Oslo: Unipub/Faculty of Humanities, University of Oslo.

———, 2010. Some new observations on *because (of)*. In Aloni, M. and K. Schulz (eds.), *Amsterdam Colloquium 2009*. Springer.

van Eijck, J. and H. Kamp, 1997. Representing discourse in context. In van Benthem, J. and A. ter Meulen (eds.), *Handbook of Logic and Language*. Amsterdam: Elsevier.

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

Fritz Hamm and Torgrim Solstad
Institute for Natural Language Processing,
University of Stuttgart
{fritz,torgrim}@ims.uni-stuttgart.de