

The proper treatment of events

Michiel van Lambalgen

*Department of Philosophy
University of Amsterdam*

Fritz Hamm

*Department of Linguistics
University of Tübingen*

Contents

Preface	5
Part 1. Time, events and cognition	7
Chapter 1. Time	9
1. Psychology of time	10
2. Why do we have the experience of time at all?	16
Chapter 2. Events and time	21
1. The analogy between events and objects	21
2. The Russell-Kamp construction of time from events	23
3. Walker's construction	26
4. Richer languages for events	27
5. Some linguistic applications	29
6. **Continuous time from events	31
7. Conclusion	32
Chapter 3. Language, time and planning	33
Part 2. The formal apparatus	39
Chapter 4. Events formalized	41
1. A calculus of events	41
2. The axiom system EC	45
3. Scenarios	48
4. Minimal models	52
Chapter 5. Computing with time and events	55
1. Logic programming with constraints	58
2. Minimal models revisited	62
3. How to get to the other side of a street	67
4. **When do causes take effect?	69
5. Exercises for chapters 4 and 5	70
6. Da capo, with feeling	71
Chapter 6. Finishing touches	75
1. Coding VPs as fluents and events	75
2. Consistency, truth and partiality	77
Part 3. A marriage made in heaven – linguistics and robotics	83

Chapter 7. Aktionsart	85
1. Eventualities	86
2. Formal definition of <i>Aktionsarten</i>	87
3. Perfective and imperfective eventualities	95
Chapter 8. Tense	99
1. Reichenbach's reference time R	100
2. Event time and the sentence	106
3. Present tense	107
4. Past tense	111
5. Future tense	116
6. Exercises	130
Chapter 9. Tense in French: Passé Simple and Imparfait	131
1. Introduction	131
2. Data	132
3. Formalizing the Passé Simple and Imparfait	137
4. Coda	148
5. Exercises	149
Chapter 10. Grammatical aspect	151
1. The perfect	151
2. The progressive	156
3. **A computational proof	159
4. Comments on the literature	161
5. Exercises	167
Chapter 11. Coercion	169
1. Additive coercion	171
2. Subtractive coercion	172
3. Cross-coercion	173
4. Temporal adverbials: 'in' and 'for'	177
5. Coercion and intensionality	180
6. Exercises	182
Chapter 12. Nominalization	185
1. Two types of English Gerunds	185
2. History of the English gerundive system	195
3. Nominalizations formalized I: Denotation types	205
4. Nominalizations formalized II: Lexical meaning	222
Chapter 13. Appendix: the basics of logic programming	227
1. Logic programming for propositional logic	228
2. Logic programming for predicate logic	233
Bibliography	237

Preface

This book studies the semantics of tense and aspect from the vantage point of cognitive science. We start from the hypothesis that one may learn something about the coding of temporal notions in natural language by looking at the way human beings construct time. This formulation may surprise some readers: surely we perceive rather than construct time? In the first Part of the book, ‘Time, events and cognition’, it is argued that, on the contrary, time is a construction, and that it is far from obvious why humans should experience time at all. The provisional answer to the latter query is that the experience of time is intimately related to the need for planning. This suggests that the linguistic encoding of temporal notions is best explained in terms of planning and its next of kin, causality.

Part 2, ‘The formal apparatus’ introduces a fully computational theory of planning, a version of the so-called event calculus as used in AI, reformulated in constraint logic programming. The formalism introduced here is somewhat more technical than is customary in semantics books. The formal setup we have chosen reflects our belief that semantics, in order to be cognitively relevant, should start from a computational notion of meaning. Using the traditional terminology of sense and denotation, our point of view can be described succinctly thus: ‘the sense of a linguistic expression is the algorithm which allows one to compute the denotation of that expression’. An added bonus of such a computational approach is that the step from theory to implementation need not be very large. This part requires a rudimentary acquaintance with logic programming, but the necessary background is provided in the appendix, Chapter 13. Exercises are provided which help the reader to master the material. In Part 3, ‘A marriage made in heaven – linguistics and robotics’, we apply the formalism of part II to a variety of data. For instance, in a chapter devoted to French past tenses, the peculiar interaction between sentence order and verb tense which yields event order is studied in great detail, as an example of the computational power of the theory. This chapter has a pedagogical purpose as well, since the required derivations in logic programming are exhibited in full. The other chapters, for instance those on grammatical aspect and on coercion, require the same computational machinery, although fewer details are given; but in the exercises which follow the chapters, the reader is invited to construct the derivations herself.

Some indications on how to use this book. It is not necessary to read the book from cover to cover. If the reader is not overly interested in the

cognitive background provided in Part 1, she needs to read only chapter 3, with perhaps the occasional glance back to the earlier chapters. Readers not interested in earlier formal approaches to events may skip Chapter 2 except for Section 1. In Part 2, Chapter 6 can (and perhaps should) be skipped at first reading, in the sense that the chapters on tense and aspect in Part 3 can by and large be understood on the basis of Chapters 4 and 5 only. Chapter 12 on nominalization however, requires Chapter 6 essentially. Throughout the book, paragraphs between ‘**’ indicate elucidations of a technical nature. These can be skipped without loss, and the same holds for most footnotes. There is a website accompanying the book (see <http://staff.science.uva.nl/~michiell>), which contains proofs not provided in the text, suggestions for additional reading, slides for instructors, and, more likely than not, errata.

It is a pleasure to record our intellectual debts here. Keith Stenning introduced us to cognitive science, and made us see how our thinking on time and tense tied in with current debates on memory architecture. Mark Steedman read the entire manuscript and provided valuable and encouraging feedback. Darrin Hindsill gave us expert advice on historical linguistics and he and Orrin Percus came to our aid when we had questions about present usage of English. Fabrice Nauze co-authored chapter 9 on French past tenses. Katja Jasinskaja and Peter Kühnlein at Bielefeld carefully read part of the manuscript and suggested many corrections. Kai-Uwe Kühnberger and Graham Katz taught a class at Osnabrück based on our previous work, which again led to many suggestions for improvement. Uwe Mönich contributed excellent advice on most topics discussed in this book. Many thanks to all of them.

Régine Pernoud wrote in her *Héloïse et Abélard*: ‘Men know little about love, but logicians least of all’. Quite so. We dedicate this book to our children:

Stijn
Jacob and Max

Amsterdam and Tübingen, February 2004

Part 1

Time, events and cognition

There is some sense – easier to feel than to state – in which time is a superficial and unimportant characteristic of reality. Past and future must be acknowledged to be as real as the present, and a certain emancipation from the slavery of time is essential to philosophic thought.

Bertrand Russell

CHAPTER 1

Time

The motto above, with which we wholeheartedly agree, is more or less the last nod to philosophy in this book. The reality or unreality of time has of course engaged philosophers since the dawn of philosophy; and since the seventeenth century, physicists have joined in the discussion as well. Whitrow's *The natural philosophy of time* [131] (reissued as [132]) is still one of the best introductions to this area. Analytic philosophers have been particularly active here, focussing on the question 'what must the world be like in order for tensed talk to make sense?'. A good source for this circle of ideas is the collection *Questions of time and tense*, edited by Robin Le Poidevin [71]. On the face of it, that question is precisely what our book should be concerned with, since it involves the semantics of tense, and the adequacy of this semantics as a description of the real world. On closer inspection, however, the correspondence turns out to be less close, since the 'tensed talk' at issue is a rather abstract version of actual temporal discourse, with 'past' and 'future' defined in terms of a single relation 'earlier than' only. We will see abundant evidence that this picture of the tenses is far too simple. By contrast, the point of departure of this book could be encapsulated by the question: 'what must our minds be like for tensed talk to make sense?', where 'tensed talk' is now meant in its full linguistic complexity.

It is a truism that we express temporal relations in language because we are able to experience these relations. It is less clear why human beings, unlike other animals, consciously experience time at all. The purpose of this part is to show that answers to this question may actually have some relevance for the semantics of tense and aspect. That is, we claim that the particular way in which temporal relations are coded in language reflects the human cognitive representation of time; moreover, we also claim that the intricate patterns of the linguistic encoding of time can be fully understood only if the mental construction of time is taken into account.

In a nutshell, the proposal is this. It will be argued that there is an intimate connection between time and planning, in the sense that the mental integration of past, present and future occurs through planning. If this is so, then the linguistic representation of past, present and future may also involve planning. The second and third part of the book are then devoted to showing that planning, suitably formalized, leads to a fully explicit computational theory integrating tense and aspect. In this first part, especially Chapter 2, the reader can moreover find a psychological and mathematical

discussion of the crucial notion of ‘event’. Verbs and verb phrases are traditionally said to denote events – but what kind of entities are these? Or, to reformulate the question more in agreement with our general approach: how are events mentally constructed? We discuss some psychological work which points to the role of goals and plans in the memory encoding of events, and which provides evidence for the hierarchical organization of events as coded in memory. This will turn out to be of some importance for our treatment of tense in Part 3.

We then continue the discussion with an axiomatic treatment of events. This is for three reasons: the reader may be (half-)familiar with Kamp’s axiomatization of event–structure, which plays some role in Discourse Representation Theory; it provides some insight into the vexed relation between events and time as a one-dimensional continuum, and it also provides a precise way of pinpointing what is wrong with the traditional concept of ‘event’.

1. Psychology of time

The reader may be under the impression that, while there are of course many deep questions surrounding the physical concept of time (mainly in relativistic situations), time as consciously experienced is a much more simple affair. After all, ‘time flows’, and we only have to perceive this flow, that is, the moving ‘now’. In reality, it is not at all like this. Our conscious notion of time is a construction, as much dependent on the operation of the memory systems as on perception. We will now set out to explain this, as a prelude to answering the main question: why do we experience time at all?

It is convenient to discuss the cognitive basis of time by distinguishing three aspects:

- (1) time as duration
- (2) temporal perspective: past, present and future
- (3) time as succession

We will discuss these in turn (gratefully using material summarized in Block’s collection [8]), and we will highlight connections to our main theme.

1.1. Time as duration. The durative aspect of time is least important from the perspective of the grammaticalization of time, since duration (‘5 minutes’, ‘two days’) is only lexicalized, not grammaticalized¹. It is nevertheless not entirely irrelevant to our concerns.

Let us consider how we may estimate duration. It is useful to distinguish here between duration in passing or experienced duration, and duration in retrospect or remembered duration. The distinction makes sense of such cases as the following. A period of time filled with exciting events will seem short in passing, but long in retrospect. However, a period of time in

¹One could argue however that it can become grammaticalized when combined with temporal perspective, for example in those languages which have separate past tenses for ‘less than one day ago’, ‘one day ago’, ‘a long time ago’ (for which see Comrie [18, p.].

which little new happens (e.g. a protracted period of illness, or waiting for connection to a helpdesk) will seem long in passing, but short in retrospect. This seems to show that attention is necessary to encode temporal information about duration: if attention is focussed on events rather than their duration, the latter type of information is not, or not correctly, encoded. Even when we do pay attention to the passage of time, our estimates may be fairly unreliable. There is no internal watch from which we can read off how much time has passed. It is not yet quite clear how we manage to give reliable estimates, in those cases where we do so. One possibility, of interest to us because it relates to our technical proposals in Part 2, is that memory contains a number of scenarios with (realistic) default values of durations of activities. By ‘default value’ we mean the duration of an activity which proceeds normally, with a minimum number of interruptions. Experiences of time running slow or fast while performing an activity are then due to a comparison with the scenario for that activity. Jones and Boltz [57] demonstrated this by comparing judgments of durations of ‘natural’ melodies with those of malformed melodies. The latter lead subjects to wrong estimations (either too long or too short) about their durations. It will be seen below that scenarios such as envisaged in [57], containing temporal information about default values, play a very important role in our semantics for tense and aspect.

1.2. Temporal perspective. The various attitudes that people adopt in relation to past, present and future are what characterize temporal perspective. It includes span of awareness of past, present and future, as well as the relative weight given to these. This aspect of time is of course highly relevant to natural language. Temporal perspective is different from the apparently simpler notion of time as succession, because it introduces a vantage point, the deictic *now*, to which all events not co-temporaneous with ‘now’ are related as either ‘past’ or ‘future’. Lest the reader thinks that it is somehow trivial to have a temporal perspective, we provide here an example of a dialogue with a schizophrenic patient who apparently lacks the stable deictic ‘I–here–now’:

Interviewer Are your uncles alive?

Patient One died in France.

Interviewer And which one are still alive?

Patient After the father from the mother’s family, only Jasiiek from France is still alive; he died already, he was killed in some kind of accident [133, p. 32].

Although time as succession is conceptually simpler than its perspectival aspect, the ordering of events or instants is most likely computed from temporal perspective. William James pointed this out forcefully, in his classic discussion of the meaning of ‘now’, or what he calls the ‘specious present’:

[T]he practically cognized present [i.e. the specious present] is no knife-edge, but a saddle-back, with a certain breadth of its own on which we sit perched, and from which we look in two directions of time. The unit of composition of our perception of time is a *duration*, with a bow and a stern, as it were—a rearward- and a forward-looking end. It is only as parts of this *duration-block* that the relation of *succession* of one end to the other is perceived. We do not first feel one end and then feel the other after it, and from the perception of the succession infer an interval of time between, but we seem to feel the interval of time as a whole, with its two ends embedded in it. The experience is from the outset a synthetic datum, not a simple one; and to sensible perception its elements are inseparable, although attention looking back may easily decompose the experience, and distinguish its beginning from its end. [55, p. 574-5]

Before James, Saint Augustine had already pointed out in Book XI of the *Confessiones*, that it is only possible to measure the duration of an event if in a sense its beginning and end are simultaneously present in awareness. Thus the present cannot be truly instantaneous. James bites the bullet, and posits that the present is in fact a bi-directional duration, one end looking toward the past, the other end looking toward the future. Memories of what is just past, and anticipation of what lies in the immediate future co-occur with current perceptions. Modern neuroscience has given some support to this picture (Pöppel [88]). There is evidence for the existence of a window of 3 seconds in which all incoming perceptions are bound together. After 3s, the brain asks: ‘what’s new?’, and updates the complex of bound sensations with new impressions.

Turning to the past, we find the obvious association of the past with memory. But what is colloquially known as ‘memory’ is actually an ensemble of several different systems. We may first distinguish between implicit and explicit memory. The former is also known as procedural or skill memory: it is the repository of our ‘knowledge how’, the various routines and skills. Explicit memory is usefully subdivided in long-term memory (comprising semantic memory and episodic memory), and short-term memory (itself comprising working memory and short-term storage facilities). Working memory is a computational arena of limited capacity, which is essentially involved in action selection. As any computing engine, it needs some registers in which to hold transient information (analogous to RAM on a computer). Semantic memory contains conceptual and encyclopedic knowledge, that is, knowledge about word meanings and about regularities in the world. This knowledge is general in the sense that individual experience has been abstracted from. The true repository of individual experiences is episodic memory, which contains our ‘memories’ in the colloquial sense. Thus, in a sense, our past resides in episodic memory; but of course the retrieval mechanisms must function properly for us to be able to access

the past. In fact, our temporal perspective (remembering the past, anticipating and planning the future) seems so natural that one may well wonder how it could be otherwise. Nevertheless, there exist clinical data which show that temporal perspective can be very different. Melges [77] discusses the case of patients with frontal lobe lesions; it is believed that such lesions may interfere with the action-selection capacity of working memory. Patients with frontal lobe lesions may become a slave to the demand characteristics of the present. Melges cites two examples: one patient who was shown a bed with the sheet turned down, immediately undressed and got into bed; another patient, who was shown a tongue depressor, took it and proceeded to examine the doctor's mouth. What is striking about these examples is that patients become dependent on the Gibsonian 'affordances' of their environment, which then act almost like stimulus-response bonds. Affordances (as defined by Gibson [43]) are the functional roles that objects may 'wear on their sleeves': in this sense a door 'affords' to go through it, and a bed with the sheet turned down 'affords' to go to sleep in it. But healthy humans use an affordance as a possibility only, to be used in the selection of actions toward a goal, and not as a necessity, i.e. as a condition-action rule.

This brings us to the cognitive representation of the future. The sense of the future seems to be bound up inextricably with the fact that humans are goal-oriented beings, as opposed to beings who are governed by large sets of condition-action rules. The realization of a goal is necessarily in the future, even though a representation of the desired state must be held in memory.

The connection between temporal perspective and being goal-oriented was investigated experimentally by Trabasso and Stein [121] in a paper whose title sums up the program: 'Using goal-plan knowledge to merge the past with the present and the future in narrating events on line'. The paper defends the thesis that 'the plan unites the past (a desired state) with the present (an attempt) and the future (the attainment of that state) [121, p. 322]' and '[c]ausality and planning provide the medium through which the past is glued to the present and future [121, p. 347]'. Trabasso and Stein present the results of an experiment in which children and adults were asked to narrate a sequence of 24 scenes in a picture storybook called *Frog, where are you?*, in which a boy attempts to find his pet frog which has escaped from its jar². The drawings depict various failed attempts, until the boy finds his frog by accident. The purpose of the experiment is to investigate what linguistic devices, in particular temporal expressions, children use to narrate the story, as a function of age. They provide some protocols which show a child of age 3 narrating the story in tenseless fashion, describing a sequence of objects and actions without relating them to other objects or actions; none of the encoded actions is relevant to the boy's ultimate goal.

²This is a classic experimental paradigm for investigating the acquisition of temporal notions in children. See Berman and Slobin [7] for methods, results, and last but not least, the frog pictures themselves.

Temporal sequencing comes at age 4, and now some of the encoded actions are relevant to the goal. Explicit awareness that a particular action is instrumental toward a goal shows up at age 5. At age 9, action–goal relationships are marked increasingly, and (normal) adults structure the narrative completely as a series of failed or successful attempts to reach the goal.

We can see from this that there is a connection between children’s developing sense of time, and their ability to structure the narrative as the execution of a plan toward the goal of finding the frog. The child of age 3 is glued to the present³. The child of 4 includes causal relations between events, states of mind and actions; these causal relations implicitly drive the narrative forward. The child of 5 can move from narrating a current action to mentioning a goal state to be attained in the future, and back again. The reason that there must be a *gradual* development of these capabilities is outlined in the following quote

Inferring goals and plans involves considerable social and personal knowledge and places heavy demands on a narrator’s working memory. The child who narrates events needs to attend to and maintain the current event in working memory; to activate and retrieve prior knowledge relevant to events, either in general or from earlier parts of the story, in order to interpret and explain the current event; and to integrate these interpretations into a context within a plan, all within the limitations of knowledge and working memory. In effect, over time the child is engaged in dynamic thinking, actively constructing and evaluating models and hypotheses about what is occurring. In so doing, the child creates a changing mental model that results in a long-term memory representation of what has occurred [121, p. 327].

This quote again emphasizes the important role that construction plays in our sense of temporal perspective. If Trabasso and Stein are correct, the construction comes about because plans integrate past, present and future. Working memory is essentially involved in this process of integration, and failures in its operation may show up in the resulting temporal perspective.

1.3. Time as succession. The idea that time is simple finds its roots in the mistaken notion that we only have to attend to the succession of events in the outside world. As James [55, p.591] clearly saw, this confuses a succession of judgments with a judgment of succession. As we have seen above, the latter type of judgment is only possible when the events are in a sense simultaneously present in awareness. But even this is clearly not yet

³In fact, reading the protocol one gets the impression that the child’s conception of time is what William James called the ‘block universe’: ‘the world is like a filmstrip: the photographs are already there, and are merely being exhibited to us’ [131, p. 274]. One might object that this is precisely the experimental situation, but the important point is that older children are able to see the picture book as a *flow*.

sufficient. As might be expected, there are limits to the human ability to discriminate order. Here are a few more or less random examples to illustrate the point. If there are two visual stimuli with stimulus onset asynchrony (SOA) less than around $44ms$ which are projected on the same retinal area, the stimuli are perceived as simultaneous. With slightly larger SOA's, subjects experience flicker, i.e. they experience successiveness without necessarily being able to encode the order of the events. If in this case the stimuli are projected on different retinal areas, subjects experience apparent movement of the stimulus. Interestingly, temporal order judgements can be wide off the mark in such a case. The paradigm example is the case where the two asynchronous stimuli have different colors, say red and green: the subject then sees one moving stimulus which changes color from red to green in mid-trajectory! What is paradoxical about this phenomenon is that apparently the subject perceives the color green before the green stimulus has occurred. Only at larger SOA's, in the order of $100ms$, can successiveness proper be perceived, in the sense that the subject can make true judgements about the order of succession⁴.

Turning now to judgments of succession of events occurring over larger time spans, in all probability the encoding of succession of such events makes use of temporal perspective. Block [8, p. 6] puts this succinctly:

Perhaps in interaction with human cognitive processes, information relating to the ordering of events from earlier to later gives rise to the common idea that the progression of time may be represented as a line or an arrow. The continuously integrated functioning of perceiving, remembering and anticipating processes apparently produces a relatively automatic awareness of the successive ordering of events. This is a fundamental aspect of all temporal experiences beyond those that merely produce an experience of successiveness without the ability to discriminate temporal order. The primary psychological basis for the encoding of order relationships between events relates to the dynamic characteristics of information processing: in the process of encoding an event, a person remembers related events which preceded it, anticipates future events, or both.

That is, while encoding an event, one simultaneously recalls related preceding events, and anticipates related future events. The relation '*e* precedes now' may then be defined operationally as: 'if I *recall* event *e*, it must have taken place before now', and analogously for the relation 'now precedes *d*': if *d* is anticipated, it must lie in the future. The composition of these two relations then gives the relation '*e* precedes *d*'. The temporal ordering is thus

⁴Data such as briefly described in this paragraph are obtained using extremely simple stimuli, such as tones of short duration and color flashes. In actual perception, where many stimuli jostle for attention, the processes leading up to a judgement of successiveness are more complicated, mainly because it then is no longer possible to treat the stimuli as pointlike.

overlayed with recollections and anticipations, and hence generally with contextual material. Interestingly, it thus seems that temporal perspective has cognitive primacy over temporal succession⁵.

It also appears from the above quotation that the ‘precedes’ relation is not automatically encoded in memory, but requires conscious attention for explicit encoding, in the sense that one should remember related past events or anticipate future events, both of which are conscious activities. Without such attention, encoding apparently does not take place, just as we saw for the case of duration. Nevertheless, it also seems to be the case that ‘precedes’ has a privileged position among temporal relations in the sense that conscious attention to temporal structure focusses on succession, and not on other temporal predicates such as ‘begins before’, ‘ends before’ or ‘overlap’. We will discuss the logical relations between these temporal predicates below. It will turn out that, provided succession satisfies a few axioms including most importantly linearity, the other relations can be defined in terms of succession. Hence also in the logical sense succession is a fundamental relation.

2. Why do we have the experience of time at all?

Summarizing the preceding considerations, the following attempt at a cognitive definition of time emphasizes its constructive nature:

Time is the conscious experiential product of the processes that allow the (human) organism to adaptively organize itself so that its behaviour remains tuned to the sequential (i.e. order) relations in its environment. (Michon [79, p. 40])

We do indeed have a conscious experience of time, and this is of course the *sine qua non* of grammatical representation. But one might ask why this is so: couldn’t we have functioned equally well without conscious experience of time? It will be seen that we can learn something about the linguistic representation of time from a consideration of this question.

First a few explanatory remarks about the kind of answer to the ‘why?’ question that can be expected. There are two main types of answers here, functional and evolutionary. In a functional explanation, one looks at the function a particular capacity has in the whole of cognition. An evolutionary explanation tends to have a somewhat different focus: either the capacity is considered in isolation, and explained as an adaptation to particular environmental pressures (e.g. the melanic moth on the blackened trees of Manchester), or it is considered as an exaptation, explaining the capacity as a fortuitous new use of a capacity evolved for other purposes (feathers evolved for temperature regulation, but were exapted for flight). We will discuss functional explanations first, and then move on to evolutionary considerations.

⁵One is reminded here of the discussions surrounding McTaggart’s ‘A-series’ and ‘B-series’, for which see for example Le Poidevin [71].

The above quotation from Michon rightly emphasizes that it is fundamental for an animal that its motor action must be ‘in tune’, ‘in sync’ with the outside world. One might think that here awareness of time *must* play a role, thus providing a functional explanation for temporal experience; but that is not so. In principle there are two ways to achieve synchronization, with and without internal clock.

Assume for the moment that activity is governed by a motor-program, stored in procedural memory. This motor-program may be seen as an abstract plan that is implemented in ever greater detail as information seeps down the hierarchy. As an example, one may think of a motor-program for writing the letter ‘A’. This case is of some interest, because it has been shown that whether a subject writes an ‘A’ with his (favoured) right hand, his left hand, or even his right and left foot, the ‘A’s produced share many structural similarities, thus leading to the hypothesis that the same abstract motor-program is in charge in all cases. For our purposes, the important point is whether time occurs as a control parameter in the motor-program, as an *internal clock*, to determine the relative timing of the actions to be performed. Sometimes such a parameter is indeed necessary, e.g. in bi-manual coordination when tapping nonharmonically related rhythms. But sometimes the time parameter appears otiose, as in our handwriting example, where the motor-program apparently specifies only the general order of actions, not their time-course, which depends on interaction with the environment (e.g. type of pen, paper, clothing; constraints of musculature). This interaction is achieved not by synchronizing external and internal clocks, but by manipulating parameters (e.g. by simulation or learning) in which time occurs at most implicitly, such as force or momentum. Hence, time itself is not a control parameter in this case. For another example, take the case of a motor skill such as the catching of a ball by a goalkeeper. Here it appears to be unnecessary to compute the trajectory and the estimated time of arrival at the goal; if the ball is shot straight at the keeper, time-to-contact is inversely proportional to dilation of the ball’s image on the retina; and this information is directly available. In principle it is possible to act, i.e. catch the ball, on the basis of this information only, without any estimate of arrival-time. It is of course also possible for the goalkeeper to be fooled by this mechanism, if a banana-shot comes his way.

The upshot of these considerations is that many motor skills do not involve explicit time (e.g. in the form of a clock), which is there to become aware of. Hence, if our conscious experience of time has a function, it is most likely not that of facilitating synchronization. So why then do we need the experience of time?

Michon [79, p. 42] advances the hypothesis that an impasse in the process of action-tuning may lead us to explicitly lay out the order of actions to be performed, and their duration. This ties in with an intriguing hypothesis advanced by Suddendorf and Corballis [118], who claim that what they call ‘mental time travel’, i.e. the possibility to revisit the past and to imagine

the future, is really a corollary of the human ability to dissociate from the present, and in particular to imagine possible worlds and possible sequences of events. The argument leading up to this hypothesis is intricate, and, as is often the case in cognitive science (including this book), built upon a certain interpretation of the literature reviewed. Nevertheless, the synthesis presented by Suddendorf and Corballis is very suggestive.

They first contrast humans with the great apes and other animals, and note that whereas humans can travel backwards and forwards in time, and can sense their self in the not-now, other animals do not seem to have this capability. They propose what they call the ‘Bischof–Köhler hypothesis’ as a description of the difference:

Animals other than humans cannot anticipate future needs or drive states and are therefore bound to a present that is defined by their current motivational state.

In other words, animals would not be able to entertain ‘conflicting’ states of mind such as ‘not hungry now’, ‘will be hungry tomorrow’. And, for that matter, neither can children age 3 or below. In one celebrated experiment from the family of ‘false belief tasks’, the Smarties test, a child is shown a Smarties box, and asked what it expects its contents to be. (For American readers: Smarties are M&M’s.) The child happily answers ‘Smarties!’, but upon opening the box finds that it contains pencils instead. If one now asks the child somewhat later what it expected the content of the Smarties box to be, it answers ‘pencils!’. This is in line with the child’s answer to the question, what another person not present in the room will expect to find in the box; again the answer here is ‘pencils’. The child thus appears to have no representation of its former self. In this sense also the child, unlike the adult, cannot experience its self in the not–now.

Suddendorf and Corballis then consider what cognitive systems subserve the apparently uniquely human capability to dissociate from the present. Implicit, or procedural, memory is not concerned with individual events, but with abstracting patterns from events, including their spatio-temporal characteristics, so implicit memory is no use in explaining time travel. Turning to declarative memory, we may note that semantic memory is able to represent singular facts, unlike procedural memory, but these have the character of learned not experienced facts. Therefore it must be episodic memory which holds the key to mental time travel. The originality of Suddendorf and Corballis [118] lies in the suggestion that episodic memory, seemingly concerned with the past, may actually be instrumental for our conception of the future. The first step in the argument is that episodic memory is not a static datastructure, a repository of experienced facts. It rather involves a constructive activity, which draws upon other cognitive abilities such as semantic memory. The second step is the observation that episodic memory actually offers a rather poor and unreliable representation of the past. This suggests that the primary function of episodic memory is therefore not veridical representation of the past, but rather a generalized capacity for

imagining or constructing possible worlds, possible courses of action, of which the coding of autobiographical memories is only a corollary. If this is correct, our sense of time derives from being goal-oriented agents, as was indeed suggested by Michon [79, p. 42]. It is then but one step to hypothesize that the *linguistic* coding of time is also driven by the future-oriented nature of our cognitive make-up. The main purpose of this book is to work out this suggestion in full technical detail.

CHAPTER 2

Events and time

The basic building block of the human construction of time is the event (cf. [134]). This is however a fairly ill-defined notion, and the present chapter looks at some attempts to add precision, as a prelude to the formal treatment in the second and third parts of the book. We will mostly be concerned with attempts to axiomatize the notion of event. These axiomatizations were often proposed in order to derive the structure of time, as a one-dimensional continuum, from properties of an *event* structure, deemed to be somehow more immediate. The motivation for studying this question has often been of a metaphysical nature, aimed at reducing time to something more basic. For instance, one might doubt that ‘instants’ or time-points are given in experience, unlike events, or one might think that modelling the time-line by the real numbers introduces unpalatable ontological commitments. In such cases a sensible strategy is to define structures which do not lead to such doubts, and to prove a representation theorem showing that the doubtful structure can be constructed from the simpler structure. We shall review several such constructions, and while we do not share the motivation which originally led to these constructions, we nonetheless believe they shed much light on the notions of time and event, considered from a cognitive point of view. Just to mention one example, we will see in the following that James’ enigmatic notion of the ‘specious present’ lends itself quite well to a formalization in some of these frameworks.

1. The analogy between events and objects

Before we embark on a discussion of the various constructions of time out of events, we must say something about the notion of event, considered psychologically¹. So far we have been talking about ‘event’ rather loosely, roughly as referents of certain natural language expressions. Intuitively, all of the following seem to fall in the category ‘event’: hearing a tone, a wink, stopping a penalty kick, singing an aria, a marriage ceremony, a solar eclipse, a stock market crash, World War II, a tank battle, an individual act of bravery, a memorial service, writing a book about World War II, . . .

Zacks and Tversky offer the following as an archetype of ‘event’: ‘a segment of time at a given location that is conceived by an observer to have a beginning and an end’. It is clear that few of the above examples

¹Here we are indebted to the interesting review paper ‘Event structure in perception and cognition’ by Zacks and Tversky [134].

fit this characterization, mainly because of imprecise spatial and temporal boundaries (cf. World War I) or because there may be discontinuities (e.g. writing the book about World War II may have been interrupted for several years). Thus, the archetype can at most provide a guideline. Nevertheless, the question remains how people individuate such events. Zacks and Tversky [134] introduce the very interesting hypothesis that mental representations of events are governed by the ‘equation’

object :: space = event :: time

and apply what is known about object recognition to the temporal domain.

Thus, one might say that in both cases perception involves establishing boundaries, in the sense of separating figure from ground. Object recognition provides a clue as to how this may be achieved. An important ingredient in object recognition is the sudden change in the intensity gradient which marks the boundary between environment and object. Similarly, one may look for sudden changes to individuate events. The following changes have been proposed

- (1) a change in the ‘sphere’ of the behaviour between verbal, social and intellectual
- (2) a change in the predominant part of the body
- (3) a change in the physical direction of the behaviour
- (4) a change in the object of the behaviour
- (5) a change in the behaviour setting
- (6) a change in the tempo of the activity

It is important to note here that these changes may operate at different levels of granularity. Because the notion of granularity is so important, it is worth quoting Zacks and Tversky in full:

The smallest psychologically reified events, on the order of a few seconds, may be defined primarily in terms of *simple physical changes*. For example, think of a person grasping another’s hand, the hands going up, going down, releasing. Longer events, lasting from about 10s to 30s, may be defined in relation to some straightforward intentional act: the events described above, on the time scale indicated, form a handshake. From a few minutes to a few hours, events seem to be characterized by *plots* (i.e. the goals and plans of their participants) or by socially conventional form of activity. Perhaps the handshake was part of signing a treaty. On time scales that are long enough, it may be that events are characterized *thematically*. In this example, perhaps the treaty signing was part of an event called a ‘peace process’. *In general, it seems that as the time scale increases, events become less physically characterized and more defined by the goals, plans, intentions and traits of their participants.* [134, p. 7; emphasis added]

Again we see that goals and plans play an important role in individuating events, and hence indirectly also in constructing time.

Moreover, events are naturally organized in part-whole hierarchies. In fact, humans have the ability to parse events at different levels of granularity to facilitate processing [134, p. 8]: people tend to divide the stream of behaviour into smaller units when it is unpredictable; but providing information about the behaviour's goal leads to larger units. This may reflect the attempt to maintain a coherent understanding of the environment while keeping processing effort low. When a coarse temporal grain is insufficient to achieve this understanding, people shift to a finer grain of encoding. It will be noticed that this picture is completely in line with Marr's view of object recognition and of intelligence generally (cf. chapters 5 and 7 of [75]; compare [123]). In Part 3 these hierarchies of events will be seen to play an important role in defining tense. There, the analogy between events and objects will be extended to an equation first proposed by Emmon Bach [6]

events:processes :: things:stuff

which is invoked to explain the semantic parallel between the pair mass/count nouns on the one hand, and the pair processes/events on the other.

We now return to the main themes of this chapter: axiomatizations of the notion of event, and the construction of time from events. We will adopt a logical point of view, and investigate the relation between a given formal language for events and its expressive power. The conclusion will be that while some languages are adequate to model the construction of time from events, none of these languages captures the rich notion of event necessary for verb semantics.

2. The Russell-Kamp construction of time from events

This construction arose out of Russell's metaphysical considerations. As is well-known, Newton believed that time is a physical entity in itself: 'Absolute, true and mathematical time, in and of itself, in its own nature flows equably and without relation to anything external ...'. Leibniz believed that time is relative in the sense that it is dependent on the events that occur: no events, no time, and moreover, the structure of time depends on the structure of events (see [131] and [132] for discussion). Russell, in 'Our knowledge of the external world' ([95], cf. also [96]), was concerned with formalizing the latter point of view, as part of a program to reduce all knowledge of the world to sense data. His construction was later taken up and somewhat modified by Kamp in [58], and it is this version that we shall discuss.

Here, the setup is very simple. The language for talking about event structures contains the binary predicates P for 'precedes' and O for 'overlap', and nothing else; variables range over events only. This choice of

language entails that we can express only temporal relationships between events²

The following seven axioms then characterize event structures (all variables are assumed to be universally quantified)

- (1) $P(x, y) \rightarrow \neg P(y, x)$
- (2) $P(x, y) \wedge P(y, z) \rightarrow P(x, z)$
- (3) $O(x, x)$
- (4) $O(x, y) \rightarrow O(y, x)$
- (5) $P(x, y) \rightarrow \neg O(x, y)$
- (6) $P(x, y) \wedge O(y, z) \wedge P(z, v) \rightarrow P(x, v)$
- (7) $P(x, y) \vee O(x, y) \vee P(y, x)$

The last axiom blatantly forces linearity of time, which is somewhat disappointing, since it seems hard to motivate it independently of linearity. We could simplify the axioms by defining $O(x, y)$ as $\neg P(x, y) \wedge \neg P(y, x)$, but this definition has linearity built in. The possibility to define $O(x, y)$ emphasizes, however, that in this setup only the ‘precedes’ relation is truly primitive.

We will now see how a version of the time line can be derived from event structures. One way (neither Russell’s nor Kamp’s) of looking at this construction is viewing an instant as the ‘specious present’ in the sense of James. This notion might seem contradictory: the ‘specious present’ has duration, and isn’t an instant supposed to have no duration? Yes and no. An instant will be defined as a maximal set of overlapping events. If one intuitively represents an event by a nontrivial interval, and if there are only finitely many events in all, then, still speaking intuitively, an instant will have duration. On the other hand, since we have taken the set of overlapping events to be maximal, instants are distinguishable only in so far as they can be separated by events, so that they cannot be further subdivided. These intuitive ideas are formalized in the following

DEFINITION 1. *Let $\langle E, P, O \rangle$ be a structure which satisfies axioms 1–7.*

- (1) *An instant of $\langle E, P, O \rangle$ is a maximal subset of pairwise overlapping events, that is, i is an instant of $\langle E, P, O \rangle$ if*
 - (a) $i \subseteq E$
 - (b) $d, e \in i$ implies $O(d, e)$
 - (c) if $e \in E$ but $e \notin i$ then there exists $d \in i$ such that $\neg O(d, e)$.

²Given the informal definition of event given above (‘a segment of time at a given location that is conceived by an observer to have a beginning and an end’) it would be more natural to derive both time and space from an event structure; this has been done in the context of special relativity theory, but it leads to formidable technical complications. Also, in the language chosen we cannot express the difference between events being goals, or actions possibly leading up to a goal.

(2) Let I be the set of instants so constructed³. For $i, j \in I$, put $i < j$ if there are $d \in i, e \in j$ such that $P(d, e)$.

THEOREM 1. (1) The structure $\langle I, \langle \rangle$ thus constructed is a strict linear ordering.

(2) For each $e \in E$, the set $\{i \in I \mid e \in i\}$ is a non-empty interval of $\langle I, \langle \rangle$.

We will not prove this theorem here, but we refer the reader to the website which accompanies this book.

The intuition behind the Russell–Kamp construction is that one conceives of time as composed of instants; hence events must be modelled as particular sets of instants, and conversely, events must be composed of instants. This set theoretic view of time is not the only one possible, as we will see below. Furthermore, if the reader now compares this reconstruction of the ‘specious present’ with James’ explication of this notion, she may notice an unsatisfactory feature. James wrote that ‘the unit of composition of our perception of time is a *duration* . . . with a rearward- and forward-looking end’. The latter feature is absent from the construction of instant thus given. The reader may well wonder whether James’ description is at all consistent, but in fact a different construction of instants from events has been proposed by Walker [129], a construction in which instants become directed even though they are indivisible, thus to some extent vindicating James.

Walker was a physicist interested in the foundations of special relativity theory, in particular in the question where the real numbers come from that physics uses to represent time. Many people have thought that the real line is problematic because it introduces a nondenumerable infinity to account for continuity – denumerable infinities like the integers or the rationals would be less problematic in this respect.

From the standpoint of mathematical logic there is much to be said against this view. One may argue for instance that the intuitionistic theory of the continuum (cf. for example Troelstra and van Dalen [122]) gives quite a good picture of this structure without invoking higher infinities. Even if one stays with classical logic one may question whether the cardinality of the standard model is the only indicator of ontological complexity. A very different indicator would be the degree to which we can mentally grasp these structures, as evidenced by the possibility of a complete axiomatization. Here the roles are reversed: there is no complete axiomatization of the integers with $+$ and \times as a consequence of Gödel’s incompleteness theorems, but there is a complete axiomatization of the structure $\langle \mathbb{R}; <, +, \times; 0, 1 \rangle$. Adding functions like \sin or \exp complicates the picture somewhat, but the resulting structures are still vastly simpler than that of the integers.

³**For infinite event structure the axiom of choice is needed to guarantee the existence of such instants.**

Although the preceding excursion gave some reasons to doubt the traditional way of viewing the ‘problem of higher infinities’, we shall stick here with Walker’s own motivation, not least because part of the linguistic literature on instants and events has also been concerned with this question. That is, occasionally linguists have raised doubts about the representations for time used in semantics, claiming for example that integers or rationals are ontologically more parsimonious than the reals.

3. Walker’s construction

Walker’s intention was thus to construct time as a continuum solely on the basis of (a denumerable infinity of) events and their structure⁴. We will highlight a few concepts and theorems from Walker [129] that give particular insight into the notion of time.

DEFINITION 2. *An instant of an event structure $\langle E, P, O \rangle$ is a triple (P, C, F) such that*

- (1) $P \cup C \cup F = E$
- (2) P, F are non-empty
- (3) $a \in P, b \in F$ implies $P(a, b)$ ⁵
- (4) if $c \in C$, there exist $a \in P, b \in F$ such that $O(a, c), O(b, c)$.

The virtue of this definition is that instants become directed: they have a ‘rearward-looking end’ (the P -part) and a ‘forward-looking end’ (the F -part); Walker instants thus formally represent the directedness inherent in James’ ‘specious present’. Some other features are represented as well. If C is nonempty, the last condition of the definition gives a kind of continuity: there is no gap between the present and the past, and likewise for the present and the future. In a sense this notion of instant also captures the peculiar type of duration involved in the specious present, since the events in C will be seen to correspond to open intervals.

If we now compare Walker’s instants to Russell’s, we see that a Walker instant always occurs in the empty gap between two events; if there are no such gaps, i.e. if all events overlap, then the event structure has no Walker instant. Clearly in this case there is exactly one Russell instant. Here are two more examples:

- (1) $E = \{a, b, c\}$ and these events satisfy the relations $P(a, c), O(a, b), O(b, c)$. In this case there are two Russell instants, and only one Walker instant.
- (2) $E = \{a, b, c, d, e\}$, and these events satisfy the relations $P(a, c), P(c, e), P(a, d), P(b, e); O(a, b), O(b, c), O(b, d), O(d, c), O(d, e)$. In this case there are three Russell instants, and two Walker instants, i_1 and i_2 , which can be seen as lying in the gaps between a and c , and c and e , respectively.

⁴In a sense, Russell had the same motivation, but in his case it is much harder to see which axioms on events would force the corresponding instant structure to be continuous.

⁵ P thus stands for ‘past’, C for ‘current’ and F for ‘future’.

The set of Walker instants can be linearly ordered by means of the following appealing stipulation:

DEFINITION 3. *Put $(P, C, F) < (P', C', F')$ if P is properly contained in P' .*

We then have

LEMMA 1. *The set of Walker instants with the above ordering relation is a linear order.*

Walker's construction gives vastly more insight than Russell's into the relations between past, present and future, and in the continuity of time. In fact, one striking advantage, pointed out by Thomason [119], was not even noticed by Walker himself: namely that in his setup events need not correspond to sets of points – rather, instants serve to separate events, because instants mark change. If nothing happens inside a given event, there will not be an instant inside that event. Walker's construction is thus much closer to intuitionistic conceptions of the continuum [122].

Here is a continuation of the second example above which illustrates these remarks. The instant i_1 is defined as $(\{a\}, \{b\}, \{c, d, e\})$, and i_2 as $(\{a, b, c\}, \{d\}, \{e\})$. In Walker's construction, the intervals corresponding to the events $\{a, \dots, e\}$ will be

- a corresponds to $(-\infty, i_1)$
- b corresponds to $(-\infty, i_2)$
- c corresponds to (i_1, i_2)
- d corresponds to (i_1, ∞)
- e corresponds to (i_2, ∞)

Note that the first three intervals are empty in the sense of not containing instants. This is only a contradiction in a traditional set theoretic framework, where there is only one empty set, so that $(-\infty, i_1)$, (i_1, i_2) and (i_2, ∞) would all coincide. If the intervals are considered to be *formal open intervals*, each defined by a suitable ordered pair, there is no contradiction.

In the Russell representation, the picture is very different: the Russell instants are $1 = \{a, b\}$, $2 = \{b, c, d\}$, $3 = \{d, e\}$, and events correspond to intervals as follows: $a = [1]$, $c = [2]$, $e = [3]$, $b = [1, 2]$, $d = [2, 3]$. This representation, unlike Walker's, in a sense obliterates the distinction between events and instants, as can be seen from the fact that $a = [1]$ in the above example.

4. Richer languages for events

Logic investigates the expressive power of formal languages. A logician would thus ask: is the language containing only the predicates P and O sufficiently rich to express all possible temporal relationships between events? In this Section we briefly consider other temporal relations that are relevant to semantics.

DEFINITION 4. *The predicate $B(c, d)$ (' c begins before d ') is introduced by putting $B(c, d) \leftrightarrow \exists b(P(b, d) \wedge \neg P(b, c))$. Similarly, $E(c, d)$ (' c ends before d ') is introduced by $E(c, d) \leftrightarrow \exists b(P(c, b) \wedge \neg P(d, b))$.*

The above definition embodies the claim that the relations 'begins before' and 'ends before' can only be asserted on the basis of witnesses. In other words, the definition claims that we have no cognitive primitives for the temporal relations 'begins (ends) before'; the only cognitive primitive is the relation 'precedes'.

We next consider the relation of two events being contiguous. This relation is helpful when describing the event structure corresponding to a sentence such as

- (1) John pushed the button. Immediately, light flooded the room.

Here, as indicated by 'immediately', the representation should be such that the events are contiguous.

We extend the language with the predicate $A(x, y)$ meaning ' x abuts y from the left'. If one does not consider this predicate to be a cognitive primitive, it must ultimately be definable using P only. This can be done for example as follows (note that B and E can be defined in terms of P):

DEFINITION 5. *Suppose we are working in a finite event structure. Put $A(a, b)$ iff a is an E -maximal element of $\{c \mid P(c, b)\}$ and b is a B -minimal element of $\{d \mid P(a, d)\}$. Formally, we have $A(a, b)$ iff*

$$P(a, b) \wedge \neg \exists c(P(c, b) \wedge E(a, c)) \wedge \neg \exists d(P(a, d) \wedge B(d, b)).$$

4.1. **Thomason's axioms. So far we have taken the predicates A, B, E, O to be definable in terms of P , reflecting a belief that the true cognitive primitive is 'precedes', and that there is no automatic encoding of the other predicates. It is not quite clear whether this is really so, cognitively speaking. When all of A, B, E, O are taken to be primitive instead of definable from P , one needs axioms describing their mutual relations. One such set, as given by Thomason [120], is

- (1) $P(x, y) \rightarrow \neg P(y, x)$
- (2) $P(x, y) \wedge P(y, z) \rightarrow P(x, z)$
- (3) $O(x, x)$
- (4) $O(x, y) \rightarrow O(y, x)$
- (5) $P(x, y) \rightarrow \neg O(x, y)$
- (6) $P(x, y) \wedge O(y, z) \wedge P(z, v) \rightarrow P(x, v)$
- (7) $P(x, y) \vee O(x, y) \vee P(y, x)$
- (8) $B(x, y) \rightarrow \neg B(y, x)$
- (9) $E(x, y) \rightarrow \neg E(y, x)$
- (10) $B(x, y) \rightarrow B(z, y) \vee B(x, z)$
- (11) $E(x, y) \rightarrow E(z, y) \vee E(x, z)$
- (12) $P(z, y) \wedge \neg P(z, a) \rightarrow B(x, y)$
- (13) $P(x, z) \wedge \neg P(y, z) \rightarrow E(x, y)$

- (14) $A(x, y) \rightarrow P(x, y)$
 (15) $A(x, y) \wedge P(x, z) \rightarrow \neg B(z, y)$
 (16) $A(x, y) \wedge P(u, y) \rightarrow \neg E(x, u)$
 (17) $\neg E(x, z) \wedge \neg E(z, x) \wedge \neg B(y, u) \wedge \neg B(u, y) \rightarrow (A(x, y) \leftrightarrow A(u, z))$

Thus, for example, a sufficient condition for $B(d, e)$ is the existence of a witness, but this is no longer a necessary condition; we allow that $B(d, e)$ is a primitive judgement.**

We will next discuss the respective merits of the Russell and Walker representations by means of a linguistic example.

5. Some linguistic applications

In this Section we will look at a famous example of reference to events, the relation between Imparfait (Imp) and Passé Simple (PS) in French. (See Kamp [58, 60], de Swart [25] and also chapter 9 below.)

Consider the following miniature narrative:

- (2) Il faisait chaud. Jean ôta sa veste et alluma la lampe.

The first sentence has an Imparfait, indicating that we are concerned here with a background which does not have explicit bounds. The second sentence twice features the PS, which indicates that (1) the two events mentioned should be placed inside the background provided by the first sentence, (2) in the context of the discourse, these events are punctual in the sense that they cannot be subdivided further, and (3) the events occur in the order indicated (at least as a default).

The PS has additional uses, which should also be taken account of. In the discourse

- (3) Au moment de mon arrivée a Paris je ne souhaitais pas voir ma tante, ayant été élevé par mon père dans l'horreur du monde. Quand je la connus, elle ne me déplut pas. (A. Maurois, *Climats*)

the PS ('connus') has an inchoative use, and refers to an event marking the onset of the state or activity denoted by the untensed verb (in this case the protagonist's state of knowing his aunt). It is even possible for the PS to refer to the end of an event: the sentence

- (4) Pierre dîna vite.

may have the meaning *Pierre acheva vite de dîner*. Thus the PS may involve reference to an event taken as a whole, and to its left and right endpoints.

We will now try to model this simple example in both the Russell-Kamp and the Walker framework, to see which (if any) fits best, in the sense of supplying discourse referents of the appropriate structure.

We simplify the first example to

(5) Il faisait chaud. Jean ôta sa veste.

In the Russell-Kamp framework, a first analysis would introduce events a (corresponding to *Il faisait chaud*), and e (corresponding to *Jean ôta sa veste*), such that

(6) $O(a, e)$ (first formalization of (5)).

This is clearly not satisfactory, as the representation theorem would yield a linear order consisting of one instant i only, so that both a and e are mapped onto the closed interval $[i, i]$.

One may add more detail, and express that e falls inside a by using the defined predicates B and E

(7) $B(a, e) \wedge E(e, a)$ (second formalization of (5))

Russell's representation theorem now gives the linear order $i < j < k$, with a mapped onto $[i, k]$ and e mapped onto $[j, j]$. This is already much better: a and e have different representations, and e is pointlike. Still, this result seems to interfere with the possibility to have the PS refer to left and right endpoints (assumed to be different). For this, e would have to contain at least two points, which can only happen when there are further events dividing e ; and the discourse does not supply these.

In Walker's framework the outcome is different. The proper formalization is again (7), but Walker's representation theorem makes a correspond to $(-\infty, \infty)$, and e to the formal open interval (i_e, j_e) , where $i_e = (\{b\}, \{a\}, \{e, d\})$, $j_e = (\{b, e\}, \{a\}, \{d\})$. Here, b is a witness for $B(a, e)$, and d is a witness for $E(e, a)$, given by the definitions of these predicates. The perfective aspect of e (forced by the PS) is then expressed by the fact that the formal open interval (i_e, j_e) does not contain any instants, hence does not have any internal structure. Note that this does not imply that e must be a point; perfectivity is not the same as punctuality. At the same time the representation yields left and right endpoints, thus accounting for the other referential possibilities for the PS.

The expressive possibilities and limits of this particular axiomatic approach can also be illustrated nicely by means of the French *Passé Antérieur*⁶.

In the literature, the *Passé Antérieur* (PA) is usually considered as a 'past in the past', that denotes an event directly preceding another one. In [111, p. 214, footnote 2], Sten cites a definition from M. Dauzat (*Phonétique et grammaire historiques de la langue française*) which could be translated as follows. The PA denotes an "immediate or precise anteriority". What is meant by 'precise anteriority' is that the interpretation of the sentence does not allow any *relevant* event in between the two clauses. It is also noted

⁶The next few paragraphs are based on joint work with Fabrice Nauze. See also Chapter 9 below.

that the PA makes reference to the end of the event it denotes. A typical use of the PA is in a subordinate clause introduced by *quand*, *lorsque*, *dès que* among others, where the main clause is in the PS.

(8) Après qu'il eut mangé, Jean partit pour la gare.⁷ (PA, PS)

It is often said in cases such as (8) that the subordinate PA structure denotes "the initial phase of the consequent state" of the subordinate⁸, that is, it has an inchoative meaning. To use Kamp's words in [59, p. 113], it denotes "the onset of state which results from his having eaten: It is the time of the beginning of the result state that serves as an anchor point for the location of the main clause event".

We note here that according to these characterizations the PA is 'future-oriented' in a way that is obscured by the description 'past in the past'. The emphasis is not on the event of eating itself, but on the state that results from that event, which therefore must be later than that event. This ties in nicely with the considerations of the previous chapter; and in this respect the PA is similar to the English perfect discussed in the next chapter.

In the context of the present chapter, which discusses languages and axioms for events, it is of some interest to observe that 'immediate anteriority' can be formally represented by means of the 'abut' predicate $A(b, c)$. We thus seem to need some of the expressive power of the axiom set 2 of 4.1. But A is a purely temporal predicate, and the meaning of the PA goes beyond the purely temporal. This becomes clear when we consider the following example from Michelet

(9) La république romaine ne tomba que 500 ans après qu'elle eut été fondée par Brutus. (PS, PA) (Sten [111, p. 214])

In this example one does not speak of 'immediate anteriority', but rather of 'precise anteriority' between the two events. The emphasis is now on the relation between the two clauses, independently of what could happen in between, even if that is a period as long as 500 years. Very many events have occurred in this period, but these are not relevant for the 'abuts' relation as it is used in the PA.

6. **Continuous time from events

Walker's aim was to show how a sufficiently rich set of events could lead to time as a classical continuum, ideally the real numbers. By 'classical' continuum we mean that the continuum is viewed set theoretically, so that intervals can be characterized as sets of points. Technically, a classical continuum is characterized by two properties: (1) every set of points with an upper bound has a least upper bound (completeness), and (2) the existence

⁷See [59, p. 113].

⁸See Gosselin [45, p. 213].

of a countable dense subset, where ‘dense’ means that between any two points of the set there lies a third.

The first property holds automatically of $\langle \mathcal{I}(E), \langle \rangle \rangle$; we need not go into the actual proof. The second property does not hold without imposing a further condition on the event structure.

DEFINITION 6. *Let $\langle E, P, O \rangle$ be an event structure, and $a, c, d \in E$. One says that the pair (c, d) splits a if $P(c, d)$ and both $O(c, a)$ and $O(d, a)$. $\langle E, P, O \rangle$ is dense if for all $a, b \in E$ with $O(a, b)$ there are $c, d \in E$ such that (c, d) splits a and b . A subset $S \subseteq E$ is dense if for all $a, b \in E$ there are $c, d \in S$ such that (c, d) splits a and b .*

The interest of this condition lies in the fact that it is a qualitative version of the infinite divisibility of time. We then have

THEOREM 2. *If $\langle E, P, O \rangle$ is an event structure, then $\langle \mathcal{I}(E), \langle \rangle \rangle$ is order-isomorphic to the reals if $\langle E, P, O \rangle$ is non-empty and dense, and has a countable dense subset.⁹*

For those who have ontological qualms about the real numbers as a representation of time because the standard model of the reals is uncountable, this result provides some comfort. Technically, it is rather interesting that the continuity property in the definition of Walker instants is used essentially. There is no way one can get a similar result using Russell instants.**

7. Conclusion

This chapter has mostly been concerned with the formal investigation of temporal relations between events, and the derivation of time from events. These relations (‘precedes’, ‘begins before’, ...) typically figure in discourse-oriented approaches to semantics such as DRT. An important question is whether the language chosen is expressive enough for the purposes of semantics. We briefly discussed this question, following Thomason, in the temporal domain. The question can however also be understood as: do purely temporal relations suffice to capture the properties of events that are relevant to semantics? In this chapter and the previous one we have seen several times that adult cognitive representations of events may also involve other, nontemporal, relations, such as that of a goal and an action instrumental in achieving that goal, or the relation of cause and effect. There is some evidence that the semantics for tense and aspect must refer to such nontemporal relations as well, evidence that will be briefly reviewed in the next chapter. The upshot of that discussion will be that, while events are indeed fundamental for verb semantics, the formal language describing them cannot be purely temporal.

⁹Of course the resulting structure $\langle \mathcal{I}(E), \langle \rangle \rangle$ is only *order*-isomorphic to the reals. We have not introduced operations such as $+$ and \times on the set of Walker instants, which would be necessary for the measurement of time.

CHAPTER 3

Language, time and planning

In the literature there have been various attempts to link the language capacity with the planning capacity. The setting is usually a discussion of the evolutionary origin of language. Even if it is granted that some non-human primates have learned a primitive form of language, there is still a striking difference in language proficiency between chimpanzees and ourselves. It is still a matter of ongoing debate to determine exactly what the difference consists in.

Some would say that the difference is in syntax: human syntax is recursive, chimpanzees' syntax (if that is the word) is not. One may then point to an analogy between language and planning. Language production can be characterized as transforming a semantic structure, to which the notion of linearity may not be applicable, into linear form (the linguistic utterance). Similarly, planning may be characterized as setting a goal and devising a linear *sequence* of actions that will achieve that goal, taking into account events in, and properties of the world; hence planning also essentially involves linearization. In the next step of the argument, the recursive structure of syntax is then linked to the recursive structure, i.e. the hierarchical organization, of planning (see for example Greenfield [46] and Steedman [110]). That is, planning is used to explain both the continuity with nonhuman primates, and the divergence. Both humans and nonhuman primates engage in planning. Primates are adept at planning, as has been known since Köhler's 1925 observations [63]. It has even been attested in monkeys. In recent experiments with squirrel monkeys by McGonigle, Chalmers and Dickinson [76], a monkey has to touch all shapes appearing on a computer screen, where the shapes are reshuffled randomly after each trial. The shapes come in different colours, and the interesting fact is that, after extensive training, the monkey comes up with the plan of touching all shapes of a particular colour, and doing this for each colour. This example clearly shows the hierarchical nature of planning: a goal is to be achieved by means of actions which are themselves composed of actions. Assuming that the planning system has been co-opted for language syntax would then ensure evolutionary continuity between ourselves and our ancestors. At the same time, the differences between ourselves and our ancestors can be explained, because human planning allows much greater nesting.

Another view has been expressed by Stenning [114], who claims that one important difference between the language use of humans, even human infants, and a relatively proficient symbol-using ape such as Kanzi

(see Savage-Rumbaugh [97]), is the possibility of *discourse*, the ability to string utterances together into a coherent narrative, with all the anaphoric and temporal bindings that this entails. We have seen while discussing the work of Trabasso and Stein in Section 1.2 that planning plays an important role in establishing such coherence. We now want to amplify this point by discussing some well-known linguistic data, which seem to indicate that planning, and the causal theories which necessarily subserve planning, can also organize temporal features of discourse.

The first data concern the relation between sentence order and event order in French. In particular we are interested in the role played by two French past tenses, *Imparfait* (Imp) and *Passé Simple* (PS).

- (1) Il faisait chaud. Jean ôta sa veste. (Imp, PS)

As we have seen, in the canonical interpretation of this discourse the first sentence defines the background, an ongoing state, and the second sentence foregrounds an event which takes place inside that state. The next examples show that different relations between background and foreground are possible.

- (2) Jean attrapa une contravention. Il roulait trop vite. (PS, Imp)

At least in pre-radar times, Jean's being fined (foreground) would put a halt to his driving too fast (background).

- (3) Jean appuya sur l'interrupteur. La lumière l'éblouissait. (PS, Imp)

In this case, the state described by the *Imparfait* immediately follows the event described by the *Passé Simple*. We see from these examples that there is no relation between on the one hand the order in which the tenses occur, and on the other hand the order of the events. The latter order is determined by world knowledge. For instance, in example (1) the order arises from the well-established fact that taking off one's sweater has no influence on the ambient temperature. In (2) the order is determined by a causal theory, which says that driving too fast may lead to a fine, but not conversely. One commonality in these examples is that one event functions as background to a foregrounded event. This distinction cannot be made in the event ontologies considered in the previous chapter.

Here is another set of examples, this time involving the *Passé Simple* only.

- (4) Pierre ferma la porte. Il monta dans sa chambre. (PS, PS)
 (5) Pierre ferma la porte. Jean monta dans sa chambre. (PS, PS)
 (6) *Pierre monta dans sa chambre. Il se leva. (PS, PS)
 (7) Pierre brisa le vase. Il le laissa tomber. (PS, PS)

In (4) the order of events is that of the sentences. In (5) the order is not determined; because the second sentence has a different subject, the events might as well be simultaneous. Sentence (6) is ungrammatical because mounting the stairs is only possible in upright position¹. In (7) the order of events is the reverse of the order of the sentences, because a vase generally breaks when it is dropped. The picture that emerges from these examples is that the *Passé Simple* introduces events into the discourse with undetermined order relations. The order relations only become fixed upon applying causal knowledge.

The very least thing that one can learn from such examples is that the true nature of tense becomes clear only in discourse, and not in single sentences, as linguists in the generativist tradition would hold. Furthermore, we advance the hypothesis that the planning system (which includes a representation of causal relationships), applied to events described by VPs or sentences, makes humans able to determine order of events automatically, independent of the order of the sentences.

In the following class of data on the English future tense, we find further corroboration for the idea that planning plays a role in semantics. English has several ways to express future tense, either using the syntactic present tense or present progressive, or various auxiliaries such as *will* or *be going to*. The pattern of grammatical and ungrammatical (or at least infelicitous) uses of these expressions at first seems rather bewildering. Here is a sample:

- (8) a. The sun rises at 5.50 a.m.
b. ?The sun will rise at 5.50 a.m.
- (9) a. *It rains tomorrow.
b. It will rain tomorrow.
- (10) a. John flies to Chicago tomorrow.
b. John is flying to Chicago tomorrow.
c. John will fly to Chicago tomorrow.
- (11) a. *I go to Chicago unless my boss forbids me.
b. (Google) I will go unless there is severe or dangerous weather.
- (12) a. Bill will throw himself off the cliff.
b. Bill is going to throw himself off the cliff.
- (13) a. (Google) A decade ago, Boston University played a seminal role in securing initial funds from the U.S. government for the Superconducting Super Collider – a particle accelerator of unprecedented scale that was going to be built in Texas [but never was].
b. (Google) Tony Blair in 1997: 'I am going to be a lot more radical in government than people think'.

¹Although one can imagine a scenario where Pierre is mounting the stairs on his knees as a form of penance. *'s are seldom absolute.

- c. (Anonymous reviewer of this book) This book will undoubtedly sell well in the Netherlands.
- (14) If/When you go out/*will go out in the rain, you will get wet.
- (15) a. *If it'll rain, you should take your umbrella.
b. If it's going to rain, you should take your umbrella.

The English future tense seems to be characterized by *two* dimensions: (a) whether one views an event as a spatio-temporal chunk only, or also views it as a goal to be achieved, and (b) whether or not the event is sure to happen, and whether or not the goal is certain to be achieved. Typically, if one considers goals one uses auxiliaries. The particular auxiliary used depends on the answer to be. *Will* expresses the existence of a plan to achieve a goal, and it implies that the goal is certain to be achieved if the plan is executed properly (see (13-c) for a particularly striking example). There may however be hedges, as in (11-b). *Be going to* also expresses the existence of a plan, but it is not committed to the achievement of the goal. Thus, (12-a) is false when Bill does not in the end throw himself off the cliff, but (12-b) may still be true. Similarly, if Tony Blair had said in 1997 'I *will* be a lot more radical in government than people think', he would have uttered a falsehood, but for the truth of (13-b) it suffices that he had once the intention of being radical. The other examples will be discussed in great detail in Part 3. For now, it suffices to note that 'the' future tense of English is actually only marginally concerned with locating events, conceived of as chunks of space-time, at some time after the speech point. It is much more concerned with subtle nuances in the relation between goals, plans and actions. To provide a faithful semantics for the English future tense one needs a formal theory of goals and plans; it does not suffice to use formalisms such as branching time logic which are concerned with temporal notions only.

The last piece of linguistic data relevant to the relation between language and temporal encoding comes from the English perfect. In Chapter 1 we encountered the suggestion that human beings are actually very much future-oriented, the explanation for this probably being that humans are first and foremost goal-oriented. Suddendorf and Corballis incorporate this feature of human cognition in their interesting definition of *episode*: 'who did what to whom, where, when and why; *and what happened next*' (emphasis added). Now it seems that the perfect is the future-oriented aspect (or tense) par excellence, because the reference time (in the sense of Reichenbach) always lies in the future of the event time:

Past Perfect	Present Perfect	Future Perfect
<i>I had seen John</i>	<i>I have seen John</i>	<i>I will have seen John</i>
$\overline{\text{E R S}} \rightarrow$	$\overline{\text{E R,S}} \rightarrow$	$\overline{\text{S E R}} \rightarrow$

In conclusion, what have we learnt from the data discussed in this chapter? Here is a partial list

- (1) a formal semantics for tense should take discourse, not single sentences, as a starting point
- (2) events may have to be treated differently, depending on whether they are considered to be 'foreground' or 'background'
- (3) in a discourse, the order of events is constructed from, e.g., causal information, and not just from the order of the sentences; in a sense the temporal relations are derived from such information
- (4) the (English) future tense shows evidence of the linguistic encoding of planning
- (5) the perfect shows evidence of human cognition's orientation toward the future

All in all, this list shows that the customary approaches to the semantics of tense and aspect must fail. In its stead one must have a formal theory whose central notions are planning and causality. In Part 2 we present such a theory, which will be applied to linguistic data in Part 3.

Part 2

The formal apparatus

*Altro diletto, che 'mparar, non provo*²
Petrarca

²'My greatest happiness lies in learning'.

CHAPTER 4

Events formalized

The reader who shares Petrarca's zeal for learning has a definite advantage in this part, since there is quite a lot to be learned. Part 1, on the mental representation of time, gave us some reasons to believe that planning might be important for the domain of natural language semantics. The construction of time was seen to depend on the integrating functions of remembering and anticipation. When the goal structure is destroyed by a lesion, the characteristic sense of the future disappears. We found evidence that estimation of the duration of an activity is facilitated by comparison with a canonical scenario (also called 'script' or 'frame') for that activity – such scenarios involve goal structures and plans. In fact, the conscious experience of time, and the possibility for 'mental time travel' may be due to the necessity to solve planning problems in hitherto unfamiliar domains. Since planning is thus important for the construction of time, it is not unreasonable to explore the possibility that the linguistic encoding of time also bears some traces of planning. We thus need a formalism for planning, and here we have chosen the *event calculus*, developed for the purpose of path planning in robotics.

Having thus chosen a representational format, we then have to determine a logical and computational mechanism. Planning is a form of non-monotonic (or default) reasoning. Nonmonotonic logics abound, but none has the computational resources of the formalism adopted here, logic programming with negation as failure. Interestingly, logic programming with negation as failure is also cognitively relevant, since it embodies a kind of computation that working memory can do well. Since planning essentially refers to time, the logic programming mechanism must be able to deal with (real) numbers and their relations. The most elegant formalism for doing so is *constraint logic programming*, which will be introduced in a separate chapter.

1. A calculus of events

By definition, planning means setting a goal and computing a sequence of actions which provably suffice to attain that goal. It involves reasoning about events, both actions of the agent and events in the environment, and about properties of the agent and the environment, which may undergo change as a consequence of those events. A simple example is that of an agent who wants a light L to burn from time t_0 until t_1 , and knows that there is a switch S serving L . The obvious plan is then to turn S at t_0 , and to leave

S alone until t_1 . Even this simple plan hides a number of problems. We required that a plan should be provably correct. On a classical reading, that would mean that the plan is sure to achieve the goal in every model of the premisses, here the description of the situation and its causal relationships. Among these models, there will be some containing non-intended events, such as light turning off spontaneously (i.e. without an accompanying turn of the switch), or a gremlin turning off the switch between t_0 and t_1 . In fact it is impossible to enumerate all the things that may go wrong. No planning formalism is therefore likely to give ‘provably correct plans’ in the sense of classical logic. The most one can hope for is a plan that works to the best of one’s knowledge. The event calculus¹ is a formalism for planning that addresses some of these concerns. It axiomatizes the idea that all change must be due to a cause – spontaneous changes do not occur. It thus embodies one sense of the *common sense principle of inertia*: a property persists unless it is caused to change by an event. That is, if an action a does not affect a property F , then if F is true before doing a , it will be true after. Of course, the crucial issue in this intuitive idea concerns the notion of ‘affect’. This refers to a kind of causal web which specifies the influences of actions on properties. The other difficulty with planning identified above, the possibility of unexpected events, can be treated either in the axiomatic system, or equivalently in the logic underlying the system. The solution of this difficulty is essentially to restrict the class of models of the axiomatic system to those models which are in a sense minimal: only those events happen which are required to happen by the axioms, and similarly only those causal influences obtain which are forced by the axioms. Our treatment of the event calculus will correspond to the division just outlined: we first discuss its formalization of causality, and then move on to introduce the class of its minimal models. Formally, the event calculus requires a many-sorted first order logic with sorts for the following:

- (1) individual objects, such as humans, chairs, tables, . . .
- (2) real numbers, to represent time and variable quantities
- (3) time-dependent properties, such as states and activities
- (4) variable quantities, such as spatial position, degree of sadness, state of completion of a painting, . . .
- (5) event types, whose instantiations (i.e. tokens) typically mark the beginning and end of time-dependent properties.

A few comments on this list are in order. The predicates of the event calculus will be seen to have an explicit parameter for time. We have chosen to represent time by the real numbers, actually by the structure $(\mathbb{R}, <; +, \times, 0, 1)$. This is not to say that humans have conscious access

¹The version used in this paper was developed by Murray Shanahan (building upon earlier work by Kowalski and Sergot [64]) in a series of papers [102], [101], [103] and [100]. Shanahan’s discussion of the frame problem and his proposed solution can also be found in the book [104].

to such a representation of time, although chapter 2 gave us reasons to believe that time can at least be taken to have the topological structure of the continuum. Rather, the reals will provide the raw material out of which the cognitive representation of time is fashioned by the event calculus.

It may furthermore strike the reader that *properties* are reckoned to belong to the ontology of the event calculus, on a par with individual objects and time points. Usually properties correspond to predicates, hence objects of a different type than that of entities. But in the event calculus a property is an object which may itself fill an argument slot in a predicate. (In AI this is known as ‘reification’.) There are several reasons for this, one having to do with the notion of ‘cause’. Consider one of the most complex classes of verbs, the *accomplishments*, examples of which are ‘draw a circle’, ‘write a letter’, ‘cross the street’. Eventualities representing such verbs have an elaborate internal structure. On the one hand there is an activity taking place (draw, write, cross), on the other hand an ‘object’ is being ‘constructed’: the circle, the letter, or the path across the street. Dowty (in [30]) analyzes the progressivized accomplishment

(1) Mary is drawing a circle

as

(2) CAUSE[Mary draws something, a circle comes into existence].

That is, the sentence is decomposed into an *activity* (‘Mary draws something’) and a *partial, changing, object* (‘circle’); it is furthermore asserted that the activity is the *cause* of the change. For Dowty, causality is a relation between propositions, and accordingly he tries, not entirely successfully, to give an account of causality in terms of possible world semantics. By contrast, the event calculus gives an analysis of causality which has its roots in physics, as a relation between events.

The event calculus actually formalizes two notions of cause, and their relation. The first notion of cause is concerned with instantaneous change, as when two balls collide. We are thus concerned with an event (type) *collision*, which for simplicity is assumed to occur instantaneously. An event *type* together with a time at which it occurs (or happens) will be referred to as an event *token*. We furthermore need time-dependent properties such as, for example, ‘ball *b* has momentum *m*’. In the case at hand, the property ‘ball 1 has momentum *m* and ball 2 has momentum 0’ will be true until the time of collision *t*, after which ‘ball 2 has momentum *m* and ball 1 has momentum 0’ is true. Such time-dependent properties are called *fluents*². Intuitively, a fluent is a function of time, which may also contain variables for individuals and reals. An important feature of the formalization is that, even though a fluent is thought of as a function of time, it is represented

²The name is appropriated from Newton’s treatise on the calculus, where all variables are assumed to depend implicitly on time.

in models of the event calculus as an object. If the fluent f contains an additional parameter x , we may provisionally think of $f(x)$ as an honest function which maps x to a fluent-object. Section 2.0.1 will explain how parametrized fluents can be treated as objects in models as well.

We now want to be able to say that fluents are initiated and terminated by events, and that a fluent was true at the beginning of time. If f is a variable over fluents, e a variable over events, and t a variable over time points, we may write the required predicates as

- (1) *Initially*(f)
- (2) *Happens*(e, t)
- (3) *Initiates*(e, f, t)
- (4) *Terminates*(e, f, t)

If events happen instantaneously, these predicates are to be interpreted in such a way, that if $Happens(e, t) \wedge Initiates(e, f, t)$, then f will begin to hold after (but not at) t ; if $Happens(e, t) \wedge Terminates(e, f, t)$, then f will still hold at t , but not after t . For events which are extended in time there is more to be said, for which see Section 4.

The second notion of causality is more like change due to a force which exerts its influence continuously. The paradigmatic example here is the acceleration of an object due to the gravitational field, but other examples abound: pushing a cart, filling a bucket, drinking a glass of wine, writing a letter, As the reader can see from this list, continuous change will be important in providing a semantics for accomplishments.

Continuous change requires its own special predicates, namely

- (5) *Trajectory*(f_1, t, f_2, d)
- (6) *Releases*(e, f, t)

In the *Trajectory* predicate, one should think of f_1 as a force, and of f_2 as a variable quantity which may change under the influence of the force. The predicate then expresses that if f_1 holds from t until $t + d$, then at $t + d$, f_2 holds. In applications, f_2 will generally have a real number as argument.

The predicate *Releases* is necessary to reconcile the two notions of cause with each other. Cause as instantaneous change leads to one form of inertia: after the occurrence of the event marking the change, properties will not change value until the occurrence of the next event. This however conflicts with the intended notion of continuous change, where variable quantities may change their values without concomitant occurrences of events. The solution is to exempt, by means of the special predicate *Releases*, those properties which we want to vary continuously, from the inertia of the first form of causation.

The axioms will be seen to have the form: if there are no ‘ f -relevant’ events between t_1 and t_2 , then the truth value of f at t_1 is the same as that at t_2 . We introduce two special predicates to formalize the notion of ‘ f -relevant’ events. The first predicate expresses that there is a terminating or

releasing event between t_1 and t_2 ; the second predicate expresses that there is an initiating or releasing event between t_1 and t_2 .

$$(7) \textit{Clipped}(t_1, f, t_2)$$

$$(8) \textit{Declipped}(t_1, f, t_2)$$

Lastly, we need the ‘truth predicate’

$$(9) \textit{HoldsAt}(f, t).$$

The intuitive meaning of $\textit{HoldsAt}(f, t)$ is that the fluent f is true at time t . The problem is, however, to ensure that $\textit{HoldsAt}(f, t)$ actually has this meaning in all models of the event calculus. Without further axioms on the $\textit{HoldsAt}$ predicate, it may have very different interpretations, for example that f is false at t ! It will be seen in the next Section that in the usual setup of the event calculus, such defining axioms for the truth predicate are lacking. This unsatisfactory state of affairs will be remedied in Chapter 6, but for the moment we ask the reader to simply assume that $\textit{HoldsAt}(f, t)$ can indeed be forced to have the meaning ‘the fluent f is true at time t ’.

In order to derive predictions, e.g. on when a robot will reach its destination, one needs a theory describing the robot’s situation, conveniently divided in axioms, holding for every situation, and a scenario, laying down properties of a particular situation. We first study the axioms.

2. The axiom system EC

The axioms of the event calculus given below are modified from [104], the difference being due to the fact that we prefer a logic programming approach, whereas Shanahan uses a different technique for obtaining minimal models called circumscription³. In the following, all variables are assumed to be universally quantified.

$$\text{AXIOM 1. } \textit{Initially}(f) \rightarrow \textit{HoldsAt}(f, 0)$$

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

$$\text{AXIOM 3. } \textit{Happens}(e, t) \wedge \textit{Initiates}(e, f, t) \wedge t < t' \wedge \neg \textit{Clipped}(t, f, t') \rightarrow \textit{HoldsAt}(f, t')$$

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

$$\text{AXIOM 5. } \textit{Happens}(e, s) \wedge t < s < t' \wedge (\textit{Terminates}(e, f, s) \vee \textit{Releases}(e, f, s)) \rightarrow \textit{Clipped}(t, f, t')$$

³In [47] we followed Shanahan’s lead and also used circumscription. The reader may consult this paper if she wishes to see a gently paced introduction to this alternative. We will not mention circumscription anymore, except to alert the reader to the fact that despite the suggestive ring of the term ‘minimal’, there are actually very different ways to define minimality.

The set of axioms of the event calculus will be abbreviated by *EC*. We add some explanatory comments on the axioms. The meaning of these axioms can be seen most clearly in the case of axiom 3. Suppose a fluent f is initiated at time $t_1 > 0$, and that no ‘ f -relevant’ event occurs between t_1 and t_2 . Here, ‘ f -relevant’ is rendered formally by the predicate *Clipped*, whose meaning is given by axiom 5. Axiom 3 then says that f also holds at time t_2 . The role of the *Releases* predicate is important here, because it provides the bridge between the two notions of causality. Axiom 3 really embodies the principle of inertia as it relates to the first notion of causality, instantaneous change: in the absence of relevant events, no changes occur. However, continuous change occurs due to a force, not an event, and hence absence of relevant events does not always entail absence of change. The *Releases* predicate then provides the required loophole. Axiom 2 treats the analogous case where we already know that f holds for the first time at t_1 , without having information about f ’s being initiated; such cases may occur as a consequence of axiom 4⁴. The first two axioms taken in conjunction imply that if a fluent holds at time 0 and no event has terminated or released it before or at time $t > 0$, it still holds at t . Axiom 4 is best explained by means of the example of filling a bucket with water. So let f_1 be instantiated by *filling*, and f_2 by *height(x)*. If *filling* has been going on uninterruptedly from t until t' , then for a certain x , *height(x)* will be true at t' , the particular x being determined by the law of the process as exemplified by the *Trajectory*-predicate.

2.1. A model for *EC*. In the absence of further statements constraining the interpretation of the primitive predicates, a simple model for *EC* is obtained by taking the extensions of *Happens* and *Initially* to be empty. If we then set $\neg\text{HoldsAt}(f, t)$ for all f, t we obtain a model. However, this model is not very informative, and to facilitate the reader’s comprehension of the axioms, we will sketch an intuitively appealing class of models of *EC*. The result to be presented is weak, because we cannot yet show at this stage that such nice models are also available when *EC* is extended with some first order theory, for example laying down the meaning of a lexical expression. In order to convey the essential idea, we consider only the part of *EC* concerned with instantaneous change; we leave out axiom 4, in which case axiom 2 reduces to

$$\text{AXIOM 6. } \text{Initially}(f) \wedge \neg\text{Clipped}(0, f, t) \rightarrow \text{HoldsAt}(f, t)$$

We will not give a formal definition of model for the event calculus (‘A model is a quintuple ...’), but refer the reader to the inventory of sorts at the beginning of Section 1. We have to specify the sorts of fluents and event

⁴This axiom is lacking in Shanahan’s version of the event calculus. He tries to obtain its effect by introducing an event which will happen precisely at the moment that f holds for the first time, and applying axiom 3. However, since this axiom entails (in minimal models, for which see below) that a fluent does not hold at the time at which it is initiated, this solution can lead to inconsistencies.

have $(f, s_0) \notin e_f^-$. By definition of s_0 and of terminating event we have, however, that $(f, s_0) \in e_f^-$.

Axiom (3) receives as interpretation:

$$\exists f'((e = e_{f'}^+ \vee e = e_{f'}^-) \wedge (f', t_1) \in e) \wedge t_1 < t_2 \wedge e = e_f^+ \wedge (f, t) \in e \wedge \forall t(t_1 < t < t_2 \rightarrow (f, t) \notin e_f^-) \rightarrow \exists I(t_2 \in I \in f).$$

It suffices to show that

$$(e = e_f^+ \wedge (f, t_1) \in e) \wedge \forall t(t_1 < t < t_2) \rightarrow (f, t) \notin e_f^- \rightarrow \exists I(t_2 \in I \in f),$$

since we have

$$e = e_f^+ \wedge (f, t) \in e \rightarrow \exists f'((e = e_{f'}^+ \vee e = e_{f'}^-) \wedge (f', t) \in e).$$

Argue as in the previous case, but now define

$$s_0 = \sup\{s \mid \exists r < t_2 \exists I(r, s] \subseteq I \in f\}.$$

We must show that $t_2 \leq s_0$. By definition of e_f^- we have: $(f, s_0) \in e_f^-$ and $t_1 < s_0$. If $s_0 < t_2$. The hypothesis of the axiom would give: $(f, s_0) \notin e_f^-$, a contradiction.

The remaining axioms, 5 and 9, are true by definition. \square

This easy construction works only due to the lack of additional axioms: predicates such as *Happens* etc. did not have any further constraints to satisfy. Nevertheless, the model captures an important intuition, namely that fluents can be represented by (finite sets of) intervals as a consequence of ‘the common sense law of inertia’, and we shall come back to the question when models of this type exist. It is worth noting that not all models of *EC* are of this form. For instance, if we have one fluent f , an f -initiating event e_+ and an f -terminating event e_- such that $Happens(e_+, t)$ iff $t \in \mathbb{Q}$ and $Happens(e_-, t)$ iff $t \in \mathbb{R} - \mathbb{Q}$, then for any interval (t_1, t_2) , both $Clipped(t_1, f, t_2)$ and $Declipped(t_1, f, t_2)$ are true, so that the set of t such that $HoldsAt(f, t)$ can be anything. In an intuitive sense, this model transgresses against the principle of inertia, since the initiating and terminating events alternate more or less randomly. This shows that the principle of inertia is not fully captured by *EC* itself; we also need a suitable logic to further restrict the class of models.

3. Scenarios

The above axioms provide a general theory of causality. We also need ‘micro-theories’ which state the specific causal relationships holding in a given situation, and which list the events that have occurred in that situation. For example, in the case of ‘draw a circle’ the situation contains (at least) an activity (‘draw’) and a changing partial object (the circle in its various stages of completion); the micro-theory should specify how the activity ‘draw’ is causally related to the amount of circle constructed. This is done by means of two definitions, of *state*⁶ and *scenario*. In this chapter we provide a somewhat restricted characterization of both notions, suitable for

⁶The term ‘state’ is somewhat overused: in our context it may refer both to the *Aktion-sart* of which ‘know’ is an example, and to a particular form of description of the world

computations with ordinary logic programming with negation as failure. In a later chapter we introduce the proper format for the computations, namely constraint logic programming.

DEFINITION 7. A state $S(t)$ at time t is a first order formula built from

- (1) literals of the form $(\neg)\text{HoldsAt}(f, t)$, for t fixed and possibly different f .
- (2) equalities between fluent terms, and between event terms.
- (3) formulas in the language of the structure $(\mathbb{R}, <; +, \times, 0, 1)$

DEFINITION 8. A scenario is a conjunction of statements of the form

- (1) $\text{Initially}(f)$,
- (2) $S(t) \rightarrow \text{Initiates}(e, f, t)$,
- (3) $S(t) \rightarrow \text{Terminates}(e, f, t)$,
- (4) $S(t) \rightarrow \text{Happens}(e, t)$,
- (5) $S(t) \rightarrow \text{Releases}(e, f, t)$,
- (6) $S(f_1, f_2, t, d) \rightarrow \text{Trajectory}(f_1, t, f_2, d)$.

where $S(t)$ (more generally $S(f_1, f_2, t, d)$) is a state in the sense of definition 7.

These formulas may contain constants for events, fluents and objects, and also for reals and time points. All statements are assumed to be universally quantified with respect to time. The combination of formulas of type 6 with formulas of type 5 is said to define a *dynamics*. One final remark before we consider an example. The definition of ‘state’ refers only to fluents being true or false; it is not allowed to include conjuncts using *Happens*. This may seem strange for a formalism concerned with causality; after all, the archetypical form of causality is one where $\text{Happens}(e_1, t)$ implies $\text{Happens}(e_2, s)$ for some s slightly later than t . There is a formal reason for our choice: allowing arbitrary occurrences of *Happens* in states increases the danger of nonterminating computations. (An example will be given when we introduce the computational machinery.) This restriction is mitigated by the introduction of a class of statements called (*event-*)*definitions*

DEFINITION 9. A definition of an event e is a statement of the form $\varphi \rightarrow \text{Happens}(e, t)$, where φ contains only *Happens* formulas, and e does not occur in φ . A definition of a fluent is defined similarly, with *HoldsAt* substituted for *Happens*.

Definitions in this sense do not introduce looping computations.

3.1. Lexical meaning. Let us now consider how lexical meaning can be expressed in the formalism outlined. We take as an example the accomplishment ‘cross the street’. We claim that the meaning of telic predicates such as achievements and accomplishments is best explained by means of

as formalized in definition 7. Since both uses of ‘state’ are entrenched in the literature, we decided not to introduce terminology of our own.

a goal-plan structure. In the case of ‘cross the street’ the goal is obviously to be on the other side of the street; the way to get there, i.e. the plan, is to traverse a certain distance from the initial position (‘one-side’). As indicated above, the plan cannot be guaranteed to work, due to the possibility of collisions etc. But it is possible to come up with a plan that works assuming no unforeseen events occur. This reference to goals and plans constitutes our resolution of the imperfective paradox: the lexical entry for an accomplishment must specify a goal and a plan, but may remain silent on the attainment of the goal.

- (1) *Initially(one-side)*
- (2) *Initially(distance(0))*
- (3) $HoldAt(distance(m), t) \wedge HoldAt(crossing, t) \rightarrow Happens(reach, t)$
- (4) *Initiates(start, crossing, t)*
- (5) *Releases(start, distance(0), t)*
- (6) *Initiates(reach, other-side, t)*
- (7) *Terminates(reach, crossing, t)*
- (8) $HoldAt(distance(x), t) \rightarrow Trajectory(crossing, t, distance(x+d), d)$
- (9) $HoldAt(distance(x_1), t) \wedge HoldAt(distance(x_2), t) \rightarrow x_1 = x_2.$

The fluent $distance(x)$ is a variable quantity, formally represented by a fluent-valued function, which takes real numbers as arguments. The constant m denotes the width of the street, hence the distance to be traversed, if we assume for simplicity that the crossing takes place in a straight line, perpendicular to the sidewalk. Strictly speaking ‘John’ should also occur as an argument of the fluents *crossing*, *one-side* and *other-side*, but we have suppressed this argument for the sake of readability. Also for simplicity we have assumed uniform velocity in 8. A more general formulation would be

$$(8') \quad HoldAt(distance(x), t) \wedge \rightarrow Trajectory(crossing, t, distance(x + g(d)), d)$$

which gives the distance traversed as a function (namely g) of time⁷. Formula 9 requires some explanation. The fluent $distance(x)$ is a function in the sense that for each concrete x it yields a particular fluent, as an object. Since x is not an explicit function of t , by itself nothing precludes the possibility that for some t , $HoldAt(distance(x_1), t) \wedge HoldAt(distance(x_2), t) \wedge x_1 \neq x_2$, i.e. that someone is in two different locations at the same time. Formula 9 expresses that this is impossible. The reader may remark that strictly speaking 9 is not allowed by the definition of scenario. The proper way to handle such boundary conditions is by means of integrity constraints. These will however be introduced only when we really need them, in Chapter 8 on tense. For now we ask the reader to take on trust that the formal details can be sorted out, and we will proceed to use the simpler formulation, sometimes implicitly assuming it as a background condition.

⁷The function g must be definable in the language of the reals and should satisfy $g(0) = 0$.

In the above scenario, the reader will notice the occurrence of natural language expressions ('crossing', 'reach', ...) as arguments of predicates of the event calculus. As noted above, this is an example of what in AI is called *reification*, the process that turns predicates into objects. Linguists are familiar with this same process under the name of *nominalization*. As will be seen later, there exists a basic division between perfect and imperfect nominalizations, where the latter, but not the former, retain some traces of temporality. This distinction matches the distinction between event types and fluents, where the latter, but not the former, are functions of time. In Chapter 12 it will be shown in great detail that nominalization can be defined formally in such a way that the result is either an event type or a fluent, and furthermore that the distributional properties of nominalizations become derivable. For the moment we ask the reader to note the analogies:

perfect nominal – event type
imperfect nominal – fluent,

and to think of fluents and event types as derived from natural language expressions via nominalization.

Now that we have seen how a theory in the event calculus can give the lexical meaning of the expression 'cross the street', consider a sentence in which this expression occurs

(3) John is crossing the street.

The sentence arises from the VP by adding the present progressive⁸. Here is a first suggestion how to represent the effect of tense: one adds the following formula

(9) *HoldsAt(crossing, now)*

As Chapter 8 will show, this suggestion does not quite work, but it will suffice for our present purpose. The reader is invited to provide an intuitive argument to the effect that (9) must imply that for some $t_0 < now$, *Happens(start, t_0)*. Likewise, if the sentence considered is in the past progressive

(4) John was crossing the street.

one could add the formulas

(9') *HoldsAt(crossing, t₁)*

(10') $t_1 < now$

The progressive can only be applied if there is an activity; the present (past) progressive then says that the activity is (was) ongoing. The progressive tenses thus apply to fluents. The past tense is a different matter (it pertains to events rather than fluents) and will be considered later. Referring to our discussion of the different types of memory in Chapter 1, we note

⁸Or just present tense; for activities or accomplishments there is no semantic difference.

that there is a distinction to be made among the sentences comprising the full scenario (1–8, 9',10'). Some sentences are universally quantified with respect to time, whereas others refer to specific time points. The former, here consisting of the sentences (1–8), fix the lexical relationships between the expressions involved; they would be stored in declarative memory. It is worth spelling this out in slightly greater detail. The lexical part of the scenario must fix (important features of) the meaning (i.e. *sense*) of 'cross the street'. In setting up this part, we have cut a few corners, in particular we have left out the data structure for 'street'. This data structure would have to specify that a street is bounded on both sides by a curb and that it has a width (measured along a perpendicular). That is what we need for this application, but obviously the data structure itself will be much richer⁹. When giving the above scenario, we have already assumed available the component parts of 'street'. Needless to say, semantically speaking, 'the street' in 'cross the street' is not at all a direct object, and 'cross the street' does not arise from the binary predicate 'cross' by filling one argument slot with the NP 'the street'. Matters get worse when we consider an accomplishment like 'build a house', because the NP 'a house' need not refer to an existing house. This example will be discussed in detail later, but it will be clear that the (lexical) semantics of accomplishments cannot be given by copying the surface predicate–argument structure in the semantics.

4. Minimal models

We now turn to an important feature in which the proposed computational semantics differs from, say, Discourse Representation Theory. Discourse Representation Structures are always taken to be substructures of the 'real' world (or a world of fiction, as the case may be). By contrast, the models that we consider are 'closed worlds' in the sense that events which are not forced to occur by the scenario, are assumed not to occur. Later additions to the scenario may overturn this assumption, so that incremental processing of a discourse does not lead to a 'nice' chain of DRSs ordered by the substructure relation. Instead, we obtain a nonmonotonic progression. In [124] it is shown how the peculiar meaning of the progressive form in English can be explained in this way, and, also, that the ubiquitous phenomenon of coercion is a natural consequence of this computational model. This form of nonmonotonicity is easiest explained in terms of planning; we will later return to its linguistic relevance.

Consider a problem with planning identified in the introduction to this chapter: it is impossible to construct a plan which is provably correct in the sense that it works whatever is true in the real world. We can only hope for

⁹The data structure for lexical information typically considered by psycholinguists is the *spreading activation network*. It would take us too far afield to explain the connection here, but specifying meaning by means of a spreading activation network and by means of a scenario are actually strongly related. See Stenning and van Lambalgen [115] for an elaboration of this point.

plans which are correct with respect to the eventualities that are envisaged now, barring unforeseen circumstances. Formally, this means that we must restrict the class of models of event calculus and scenario to models which are *minimal* in the sense that the occurrences of events and their causal influences are restricted to what is required by the scenario and *EC*. Thus if a scenario contains only the following statements involving *Happens*

- *Happens*(*switch-on*, 5)
- *Happens*(*switch-off*, 10)

a non-minimal model of this scenario would be one in which the following events are interpolated between times 5 and 10:

- *Happens*(*switch-off*, 8)
- *Happens*(*switch-on*, 9)

Similarly, if the scenario contains only the following statements involving *Initially*, *Initiates* and *Terminates*

- $\neg\text{HoldsAt}(\text{light-on}, t) \rightarrow \text{Initiates}(\text{switch-on}, \text{light-on}, t)$
- $\text{Terminates}(\text{switch-off}, \text{light-off}, t)$

a non-minimal model of this scenario could contain the additional statements concerning causal influences

- *Initially*(*light-on*)
- $\text{HoldsAt}(\text{light-on}, t) \rightarrow \text{Terminates}(\text{switch-on}, \text{light-on}, t)$

While this intuition about minimality is fairly straightforward, its implementation is much less so. Relevant technical questions are: is the notion of minimality uniquely determined? Can we compute minimal models, given scenario and *EC*? Answers to these questions in general depend on the syntactic structure of the scenario; Section 2 will have some information on this issue. A pertinent question is furthermore: why should the structure of the models, and the existence of minimal models matter to natural language semantics? The main reason has to do with the computational notion of meaning advocated here. It will be shown that in the cases of interest to us, there exists in fact a unique minimal model, which defines the denotations of all expressions given their senses, as codified in the scenario. Moreover, there exists a computable procedure for obtaining the minimal model, so that denotation is in fact computable from sense.

CHAPTER 5

Computing with time and events

In formal semantics for natural language it is not common practice to associate algorithms to expressions¹. It is usually assumed that all one needs to model Frege's idea of *sense*, is the intension of an expression, defined as a (not necessarily computable) function which maps a possible world into an extension of the expression in that possible world. It seems to us that this picture of meaning is too static, by and large cognitively irrelevant, and in any case predictively deficient. In fact, if semantics wants to make contact with the huge psycholinguistic literature on language comprehension and production, it had better become computational. We propose to identify the *sense* of an expression with the algorithm that computes the expression's denotation². This short formula will doubtlessly raise many questions in the reader's mind ('the'? 'algorithm'? 'computes on what input?'), which can only be answered after we have introduced some technical machinery. Before we start doing so, let us briefly consider computability of meaning from a cognitive point of view.

It is the great insight of Kamp's Discourse Representation Theory that language understanding proceeds via the construction of a (mental) representation intermediate between the linguistic utterance and the world, the discourse representation structure (DRS). In fact, some psychological evidence that this level of representation exists was obtained by Bransford, Barclay and Franks [11]. They gave subjects a descriptive sentence, and some time later presented subjects with other sentences, and asked whether they had seen any of these before. They then observed that the recalled sentences often are *inferences* from the explicitly presented material. E.g. when given the sentence

Three turtles rested on a floating log and a fish swam beneath *them*.

subjects later confused it with

Three turtles rested on a floating log and a fish swam beneath *it*.

The substitution of *it* for *them* can easily be explained if subjects construct a model of discourse, and 'read off' what is true there. Furthermore, it

¹The situation is different in the psycholinguistics literature, see e.g. the classic [80].

²This point of view has in recent times been advocated forcefully by Moschovakis. See van Lambalgen and Hamm [125] for discussion.

was observed that recall of a piece of discourse is facilitated if the discourse determines a unique model. Compare the following two pieces of discourse.

The spoon is to the left of the knife.
 The plate is to the right of the knife.
 The fork is in front of the spoon.
 The cup is in front of the knife.

and

The spoon is to the left of the knife.
 The plate is to the right of the spoon.
 The fork is in front of the spoon.
 The cup is in front of the knife.

The first determines a unique model, the second does not. Stenning [112], [113], and Mani and Johnson-Laird [74] observed that while (a) verbal recall of the indeterminate description is better than that for determinate description, by contrast (b) the ‘gist’ of determinate description can be recalled much more accurately than for the indeterminate description. These experiments lend some support to the conclusion that models (for determinate descriptions) and verbal material (for indeterminate descriptions) are both stored in memory. In line with the above considerations, we posit that what gets computed by the sense of an expression (or set of expressions) is a model in which these expressions can be given a denotation. Furthermore, we note that there is apparently a premium on uniquely determined models. As will become gradually clear in the course of the book, there is in fact an intimate connection between tense and aspect, and models of discourse being uniquely determined.

The computational component of the theory is a version of logic programming³, called *constraint logic programming*. To see how one might compute with the event calculus, let us first consider the syntactic form of the axioms and the statements comprising a scenario. The axioms of the event calculus are not Horn clauses, due to the occurrence of the negative literals $\neg \exists s < r \text{ HoldsAt}(f, s)$ and $\neg \text{Clipped}(s, f, t)$ in the bodies of the axioms, and arbitrary first order formulas in the bodies of statements in a scenario. Also bodies may contain equalities, disequalities and inequalities involving reals. It makes sense to do computations with reals outside the logic program, but then we have to provide a format in which both kinds of computations can interact. This is constraint logic programming, which also provides a smooth treatment of negation. The reader can find information on the treatment of negation in standard logic programming (known as

³Readers unfamiliar with standard logic programming may find a short introduction in the appendix, chapter 13. Terminology not explained here is defined in the appendix.

‘negation as failure’) in the appendix. Readers not familiar with this notion are advised to read the appendix first, so as to get a good grasp of the underlying intuitions⁴.

We have chosen to represent time by means of a continuum, for which we take the reals. We furthermore have to fix the relations and operations allowed on the reals. As our underlying structure we will take the model $(\mathbb{R}, 0, 1, +, \cdot, <)$, that is, the set of real numbers with the relation ‘less than’, operations plus and times, and constants 0 and 1⁵. The corresponding language \mathcal{L} has individual constants 0 and 1, a relation symbol for ‘less than’, and function symbols for plus and times. The most important property of the structure $(\mathbb{R}, 0, 1, +, \cdot, <)$ is that it has a complete axiomatization in the language \mathcal{L} . The precise form of the axioms is of no concern to us here; we refer the interested reader to Hodges [50]. The set of axioms will be called \mathcal{T} . Speaking intuitively, $(\mathbb{R}, 0, 1, +, \cdot, <)$ is the structure underlying analytic geometry, or, equivalently, Euclidean geometry. The language allows one to define polynomials (e.g. $xy^3 + yz + 5$), and the axioms fix the properties of the operations $+$, \cdot and determine which polynomials have real solutions. For us, the most important property of this structure is that the sets of reals definable in this structure are always of a very simple kind. For instance, let $\varphi(x)$ be a formula in the language \mathcal{L} containing one free variable x . Now whatever the number of quantifiers in $\varphi(x)$, the set $\{x \in \mathbb{R} \mid (\mathbb{R}, 0, 1, +, \cdot, <) \models \varphi(x)\}$ can always be written as the union of a finite set of intervals and a finite set of points. This is a consequence of Tarski’s celebrated theorem on ‘quantifier elimination for real-closed fields’; the interested reader is referred to chapter 8 of Hodges [50] for the statement and its proof. Its importance for us can be seen from the following consideration. Typically, computing with time and events involves determining the set of t such that $HoldsAt(f, t)$ is true, where f is a fluent which represents, say, an activity. Now $HoldsAt(f, t)$ will be characterized by a logic program, which for present purposes can be equated with a complicated formula in the variable t . Using Tarski’s theorem one may then show that the set of t such that $HoldsAt(f, t)$ can also be written as a finite union of intervals and points. This is intuitively satisfying, since it shows that generally events must have this temporal profile⁶. Of

⁴Lack of space forbids us to go into a very interesting aspect of logic programming, namely its connection with neural networks. Due to the syntactic restrictions inherent in logic programming, its models can be viewed as the stable states of an associated neural network. For more information on this, we refer the reader to Stenning and van Lambalgen [115], [116] and d’Avila Garcez, Broda and Gabbay [5]. In brief, the claim is that the representational structures used here are very close to the spreading activation networks beloved of psycholinguists.

⁵**The structure chosen is a compromise between expressiveness and efficiency. Adding other functions such as \sin would destroy some of the theorems on the structure of the minimal models that are very important for the semantics of tense and aspect.

⁶Allowing functions such as the sine or the exponential would destroy this nice result.

course there is a long-standing debate in linguistics on ‘the proper’ temporal ontology for events. It is often claimed that somehow intervals are more fundamental than points, and formalisms have been proposed which allow one to deal directly with intervals; we have seen some of these in chapter 1. These formalisms are alright as far as they go, but they are not suitable to formalize continuous change, and hence they are unsuitable for treating the progressive; for this we need the underlying structure of the reals. The result alluded to above then says that one can have one’s cake and eat it: the structure of the reals is available to formalize continuous change in, but fluents and events still determine finite sets of intervals and points.

We now return to the formal development. Henceforth, formulas in the language \mathcal{L} will be called *constraints*. For example, the formula $s < t$ that occurs in the axioms of the event calculus is a constraint. We now need a programming language that allows us to mix constraints and predicates of the language of the event calculus. This language will be introduced in the next Section.

1. Logic programming with constraints

Constraint logic programming is in general concerned with the interplay of two languages. In our case these will be the languages $\mathcal{L} = \{0, 1, +, \cdot, <\}$, and the language \mathcal{K} consisting of the primitive predicates of the event calculus. The latter will also be called *programmed* predicate symbols, because we will write logic programs defining the primitive predicates. We will have occasion to extend \mathcal{L} slightly in the course of the formal development, but this choice suffices to fix ideas. Not considering negation for the moment⁷, clauses in a constraint logic program based on \mathcal{L} and \mathcal{K} are generally of the following form

$$B_1, \dots, B_n, c \rightarrow A,$$

where the B_1, \dots, B_n, A are primitive predicates and c is a constraint. Constraints may occur only in the bodies of clauses. Likewise, a query has the logical form

$$B_1 \wedge \dots \wedge B_m \wedge c \rightarrow \perp.$$

We shall use the notation

$$?c, B_1, \dots, B_m$$

for queries, always with the convention that c denotes the constraint, and that the remaining formulas come from \mathcal{K} . The words ‘query’ and ‘goal’ will be used interchangeably.

The aim of a constraint computation is to express a programmed predicate symbol entirely in terms of constraints, or at least to find an assignment to the variables in the programmed predicate which satisfies a given constraint. Thus, unlike the case of ordinary logic programming, the last node of a successful branch in a derivation tree contains a constraint instead of

⁷I.e. restricting attention to so called *definite* constraint logic programs.

the empty clause. To make this precise, we have to spell out the notion of derivation step and derivation tree. Again, the reader not familiar with logic programming, in particular such notions as resolution and completion, is advised to consult the appendix before proceeding.

One difference between standard logic programming and constraint logic programming is its treatment of substitution. In the former, the unification algorithm applied to two atoms determines which terms have to be substituted for the variables occurring in the atoms, in order for the atoms to become identical. In constraint logic programming the treatment is different: when the unification algorithm has determined that a term t should be substituted for a given variable x , one adds a *constraint* $t = x$ but no substitution is effected. If A, B are atoms, we let $\{A = B\}$ denote the set of equations between terms which unify A and B if A and B are unifiable; otherwise $\{A = B\}$ is set to \perp . The constraints are then simply accumulated in the course of the derivation. There are some clear notational advantages to this approach, which avoids nested, possibly unreadable terms. The main advantage is conceptual, however, since it allows a more symmetric treatment of positive and negative information.

A consequence of this approach is that the constraint language \mathcal{L} has to be extended, since constraints in the wider sense may now also involve objects, (parametrized) events and (parametrized) fluents. Henceforth we assume that constants and function symbols denoting objects, (parametrized) events and (parametrized) fluents also belong to the constraint language, and that the theory describing them is Clark's Equality Theory *CET*, also known as the 'uniqueness of names assumption'. This is the following set of statements for function symbols f, g and terms t of the language:

- (1) $f(y_1, \dots, y_n) = f(z_1, \dots, z_n) \rightarrow y_1 = z_1 \wedge \dots \wedge y_n = z_n$
- (2) $f(y_1, \dots, y_n) \neq g(z_1, \dots, z_m)$, where f, g are different
- (3) if y occurs in t , $y \neq t$.

In the context of the event calculus (referring to the example of crossing the street), these statements say for example that $distance(x) = distance(y) \rightarrow x = y$, and that $start$ is different from $reach$.

To summarize: the constraint structure \mathcal{A} consists of the union of $\{0, 1, +, \cdot, <\}$ and the set of objects, events and fluents, and it is described by the union of the theories \mathcal{L} and *CET*. All terms occurring in the programmed predicates must be interpreted in the given constraint structure.

The main derivation rule is *resolution*, which can be formalized as follows. Suppose $?c, B_1, \dots, B_i, \dots, B_m$ is a goal, and $D_1, \dots, D_k, c' \rightarrow A$ a program clause. A new goal

$$?c'', B_1, \dots, D_1, \dots, D_k, \dots, B_m$$

can be derived from these two clauses if the constraint c'' , defined as $c'' = (c \wedge \{B_i = A\} \wedge c')$ is satisfiable in \mathcal{A} . That is, if A can be unified with B_i , one can replace B_i by D_1, \dots, D_k if in addition the given constraint c is narrowed down to contain also the unifying substitution and the constraint

c' . Given this inference rule, the concepts of derivation tree and branch in a derivation tree have straightforward definitions⁸.

DEFINITION 10. *A branch in a derivation tree is successful if it is finite and ends in a query of the form $?c$, where c is a satisfiable constraint; note that the query is not allowed to contain an atom. A branch in a derivation tree is finitely failed if it ends in a query $?c, B_1 \dots B_m$ such that either c is not satisfiable, or no program clause is applicable to the B_i . Otherwise the branch is called infinite.*

Intuitively, this definition applied to the situation of interest means the following. Suppose we start from a query $?HoldsAt(f, t)$ and find a successful branch ending in $?c$. This should mean that for all t , if $c(t)$ is true, then so is $HoldsAt(f, t)$. Likewise, if a branch finitely fails and ends in $?c, B_1 \dots B_m$, we should have, for all t satisfying $c(t)$, $\neg HoldsAt(f, t)$. These intuitions will be bolstered up by theorems to be presented later.

For our purposes, definite constraint logic programs are not yet expressive enough, due to the occurrence of \neg in the bodies of the axioms of the event calculus. Once we allow negation in the body of a clause, we may as well allow (classical equivalents of) first order formulas of arbitrary complexity. That is, we can use all formulas containing only \exists , \neg and \wedge ; of course, each formula is classically equivalent to such a formula, but the representation chosen facilitates a smooth inference procedure. More precisely

DEFINITION 11. *A complex subgoal is defined recursively to be*

- (1) *an atom in \mathcal{K} , or*
- (2) *$\neg \exists \bar{x}(B_1 \wedge \dots \wedge B_m \wedge c)$, where c is a constraint and each B_i is a complex subgoal.*

DEFINITION 12. *A complex body is a conjunction of complex subgoals. A normal program is a formula $\psi \rightarrow A$ of $CLP(\mathcal{T})$ such that ψ is a complex body and A is an atom.*

The form of negation most congenial to constraint logic programming is *constructive* negation ([117]). In the customary negation as failure paradigm, negative queries differ from positive queries: the latter yield computed answer substitutions, the former only the answers ‘true’ or ‘false’. Constructive negation tries to make the situation more symmetrical by also providing computed answer substitutions for negative queries. Applied to constraint logic programming, this means that both positive and negative queries can start successful computations ending in constraints. The full operational definition of constructive negation is somewhat involved

⁸As in the case of standard logic programming, one also needs the concept of a *selection rule*, which determines which atom should be chosen at a particular stage in a derivation. The interested reader may consult [117]; we need not dwell on this topic here.

(see [117]), but we will provide a simplified version⁹, modeled on negation as failure, which suffices for our purposes. The reader not familiar with the latter notion is advised to consult the appendix. The operational meaning of constructive negation may now be given as follows. Suppose the goal $?\psi_1, \dots, \psi_i, \dots, \psi_n$ contains a subgoal of the form $\psi_i = \neg \exists \bar{x} (B_1 \wedge \dots \wedge B_m \wedge c)$, which has been selected for processing. Start a subderivation with goal $?B_1, \dots, B_m$. Assume this derivation tree is finite. Collect the constraints c_1, \dots, c_l occurring on the successful branches of the tree (the finitely failing branches can be disregarded). The children of the goal $?\psi_1, \dots, \psi_i, \dots, \psi_n$ are now of the form

$$?c \wedge \neg c_i, \psi_1, \dots, \psi_{i-1}, \dots, \psi_{i+1}, \dots, \psi_n$$

for all i such that $c \wedge \neg c_i$ is satisfiable. There may be no such i , in which case the goal has no children. The subderivation may itself feature negative goals, so that an abstract definition of a derivation tree allowing constructive negation involves a recursion. We will not provide a definition, but the reader may check that the derivations used in the linguistic applications all conform to the above characterization. A global concept of success for a derivation tree is given by the following definition.

DEFINITION 13. *A query $?c, G$ is totally successful if its derivation tree includes successful branches ending in constraints $c \wedge c_1, \dots, c \wedge c_n$ such that $\mathcal{A} \models \forall \bar{x} (c \rightarrow c_1 \vee \dots \vee c_n)$.*

Intuitively, a query $?c, G$ is totally successful if all instances of the query also succeed; this is much stronger than saying that the query is satisfiable, as one would in standard logic programming. As in the case of negation as failure, the fundamental technical tool in describing the semantics of the above procedure is the *completion* of a program:

DEFINITION 14. *Let \mathcal{P} be a normal program, consisting of clauses*

$$\bar{B}^1 \wedge c_1 \rightarrow p^1(\bar{t}^1), \dots, \bar{B}^n \wedge c_n \rightarrow p^n(\bar{t}^n),$$

where the p^i are atoms. The completion of \mathcal{P} , denoted by $\text{comp}(\mathcal{P})$, is computed by the following recipe:

- (1) choose a predicate p that occurs in the head of a clause of \mathcal{P}
- (2) choose a sequence of new variables \bar{x} of length the arity of p
- (3) replace in the i -th clause of \mathcal{P} all occurrences of a term in \bar{t}_i by a corresponding variable in \bar{x} and add the conjunct $\bar{x} = \bar{t}_i$ to the body; we thus obtain $\bar{B}^i \wedge c_i \wedge \bar{x} = \bar{t}_i \rightarrow p^i(\bar{x})$
- (4) for each i , let \bar{z}_i be the set of free variables in $\bar{B}^i \wedge c_i \wedge \bar{x} = \bar{t}_i$ not in \bar{x}
- (5) given p , let n_1, \dots, n_k enumerate the clauses in which p occurs as head

⁹Disregarding the possibility of infinite derivations.

(6) define $Def(p)$ to be the formula

$$\forall \bar{x}(p(\bar{x}) \leftrightarrow \exists \bar{z}_{n_1}(\bar{B}^{n_1} \wedge c_{n_1} \wedge \bar{x} = \bar{t}_{n_1}) \vee \dots \vee \exists \bar{z}_{n_k}(\bar{B}^{n_k} \wedge c_{n_k} \wedge \bar{x} = \bar{t}_{n_k})).$$

(7) $comp(\mathcal{P})$ is then obtained as the formula $\bigwedge_p Def(p)$, where the conjunction ranges over predicates p occurring in the head of a clause of \mathcal{P} .

The soundness of the operational definition of constructive negation is expressed by

THEOREM 3. *Let \mathcal{P} be a normal program on the constraint structure \mathcal{A} , and let \mathcal{T} be the axiomatization of \mathcal{A} .*

- (1) *If the query $?c, G$ is totally successful, then $\mathcal{T} + \mathcal{P} \models \forall \bar{x}(c \rightarrow G)$.*
- (2) *If the query $?c, G$ is finitely failed, then $\mathcal{T} + \mathcal{P} \models \neg \exists \bar{x}(c \wedge G)$.*

2. Minimal models revisited

In Section 4 we introduced the notion of a *minimal* model, characterized by the fact that the occurrences of events and their causal influences are restricted to what is required by the scenario and the axioms of the event calculus. We argued that the semantics of tense and aspect should be concerned above all with minimal models, because planning is sound only with respect to minimal models, and the minimal models are the ones that can be computed, thus serving as a substrate for the proposed computational notion of meaning. The present Section and the next chapter provide more details on minimal models. Here we concentrate on two-valued, i.e. total, models, in effect assuming that all computations terminate. The next chapter will study the general case of partial models.

Let us now look at the notion of minimality as informally introduced in Section 4 by means of the following example: if a scenario contains only the following statements involving *Happens*

- (1) *Happens(switch-on, 5)*
- (2) *Happens(switch-off, 10)*

a non-minimal model of this scenario would be one in which the following events are interpolated between times 5 and 10: *Happens(switch-off, 8)* and *Happens(switch-on, 9)*. Now that we have seen the notion of a completion of a program, there is an easy way to get rid of the interpolated events: form the completion of the logic program comprised of 1 and 2. Following the recipe given above, the completion becomes

$$Happens(e, t) \leftrightarrow (Happens(switch-on, 5) \vee Happens(switch-off, 10)).$$

The uniqueness of names assumption then guarantees that *Happens(switch-off, 8)* and *Happens(switch-on, 9)* must be false.

The models of the completion $comp(\mathcal{P})$ of a normal logic program \mathcal{P} are thus of particular interest for us. Due to the possible presence of negation in the body of a clause, a model of the completion of a program cannot always be an ordinary two-valued model: the program consisting of only the clause

$\neg A(x) \rightarrow A(x)$ is a counterexample to this. The usual way out is to move to (Kleene's) three-valued logic. However, the natural language examples that we are interested in do not require clauses of the offending type, and we therefore decided to omit a discussion of the three-valued semantics for logic programming. In the one case where it would be necessary, nested uses of the *HoldsAt* predicate arising from iterated nominalization, there exists a trick, due to Robert Stärk, allowing one to use classical models only. This will be explained in Chapter 6.

For the following discussion, fix a domain of the models to be considered. In our case, the domain must contain the reals together with a finite number of objects, event types and fluents. The general theory of logic programming gives us that the set of models (on the given domain) of the completion has a nice ordering in which there is a *least* model. Strictly speaking we would have to formulate this ordering using three-valued logic, but with the simplification we have made, this ordering is given by the relation 'substructure of' or \subseteq , where $\mathcal{M} \subseteq \mathcal{N}$ if the interpretation of a relation symbol R on \mathcal{M} is a subset of the interpretation of R on \mathcal{N} (recall that \mathcal{M} and \mathcal{N} have the same domain). In the ordering \subseteq there exists a minimal structure which is a model of $\text{comp}(\mathcal{P})$. In fact, in most cases of interest to us, $\text{comp}(\mathcal{P})$ has a unique model. The structure of this model is of particular importance.

It can be shown for instance that the example scenario in Section 3.1 determines a unique model \mathcal{M} , in which the fluents *crossing*, *one-side* and *other-side* are represented by finite sets of halfopen intervals with rational endpoints. To describe the structure of the parametrized fluent $\text{distance}(x)$ in \mathcal{M} , one needs the more general notion of a semialgebraic set:

DEFINITION 15. *A subset of \mathbb{R}^n is semialgebraic if it is a finite union of sets of the form $\{x \in \mathbb{R}^n \mid f_1 = \dots = f_k = 0, g_1 > 0, \dots, g_l > 0\}$, where the f_i, g_j are polynomials.*

The reader may check that a finite union of intervals with rational endpoints is indeed semialgebraic; each interval $(p, q]$ can be brought in the form $\{x \mid x - p > 0, q - x > 0\} \cup \{x \mid x = q\}$, which is semialgebraic.

In \mathcal{M} , the fluent $\text{distance}(x)$ determines a semialgebraic subset of $\mathbb{R} \times \mathbb{R}$. Observe also that the events (*start* and *reach*) mark the beginning and end of fluents. The structure of the model \mathcal{M} is thus very similar to that of the canonical model of the event calculus given in Section 2.1.

The question is how far this generalizes. Intuitively at least, fluents corresponding to natural language expressions (e.g. verbs) are semialgebraic, and we would like this to fall out of the setup, without further stipulations. Again speaking intuitively, there seems to be some connection between the property of inertia as formalized in the event calculus, and the simplicity of the sets described by fluents. A fluent which holds at rational points and is false at irrational points is somehow incompatible with inertia, because it would seem to need a great many events to turn it on and off.

As has been remarked above, in natural language semantics it is a contested issue whether the fundamental temporal entities are points or intervals. The event calculus neatly sidesteps this issue, by taking the basic entities to be events and fluents, which are not explicit functions of time and which can be interpreted on structures with very different ontologies for time. Even if we take the structure underlying time to be \mathbb{R} , that does not constitute an ontological commitment to points. Ontological commitment is generated rather by representation theorems, which correlate the events and fluents with point sets in a given structure. It may then very well turn out that, even when time is taken to be \mathbb{R} , fluents and events can be represented as sets of intervals, so that points have no role of their own to play. The situation is slightly more complicated in the case of fluents admitting real parameters, for example fluents representing possibly changing partial objects. One would expect change to be piecewise continuous, with at most a finite number of jumps, corresponding to events explicitly mentioned in the scenario. The kind of change it is possible to program depends on the one hand on the constraint language chosen, on the other hand on the constraint logic program. Now in the structure $\{0, 1, +, \cdot, <\}$ only semialgebraic sets can be defined; but it may well be that more complicated sets are definable in the theory consisting of a normal program \mathcal{P} together with the constraint theory \mathcal{T} . The next few theorems give some pertinent results. This material leans heavily on [117]. The reader for whom this is all new is advised to read only the statements of the theorems and skip the proofs. We will provide informal glosses of the main results whenever possible.

The first definition isolates the kind of programs and queries we are interested in, namely those which make computations finite.

DEFINITION 16. *A query $?c, G$ is finitely evaluable with respect to a program \mathcal{P} , if its derivation tree is finite, i.e. if all branches in a derivation tree starting from $?c, G$ end either in success or in finite failure. A normal program \mathcal{P} is finitely evaluable if every query is finitely evaluable w.r.t. \mathcal{P} .*

A query $?c, G$ may contain both variables over the reals and over objects, events and fluents. For definiteness, we call the former \bar{x} and the latter \bar{y} . We are now interested in the structure of the real part (t and \bar{x}) when the remaining variables (the \bar{y}) are held fixed. The next theorem and its corollary form the technical backbone of this book.

THEOREM 4. *Let \mathcal{T} be the constraint theory describing the structure \mathcal{A} . Let \mathcal{P} be a normal program consisting of the axioms of the event calculus together with a scenario. Let $?G$ be a finitely evaluable query in the language of the event calculus. Let \bar{b} be an assignment to the variables \bar{y} . Then there exists a semialgebraic set defined by a constraint $c(\bar{x})$ such that $\mathcal{T} + \text{comp}(\mathcal{P}) \models \forall(G(\bar{x}\bar{b}) \leftrightarrow c(\bar{x}))$.*

Informally, this theorem says that inertia indeed constrains the sets definable by a logic program to be of a very simple kind.

****PROOF.** By hypothesis the derivation tree whose top node is $?G$ is finite, hence those terminal nodes which are not marked as failures are marked by a constraint from the language of the structure \mathcal{A} , i.e. $\{0, 1, +, \cdot, <\}$ together with a set of objects, (parametrized) events and (parametrized) fluents. It then follows from lemma 7.3 in [117] that there is a constraint $c'(\overline{xy})$ such that

$$\mathcal{T} + \text{comp}(\mathcal{P}) \models \forall(G(\overline{xy}) \leftrightarrow c'(\overline{xy})).$$

Define $c(\overline{x}) := c'(\overline{x\bar{b}})$, then

$$\mathcal{T} + \text{comp}(\mathcal{P}) \models \forall(G(\overline{x\bar{b}}) \leftrightarrow c(\overline{x})),$$

and an easy extension of Tarski's theorem alluded to above shows that $G(\overline{x\bar{b}})$ represents a semialgebraic set. ****** \square

The next corollary spells out what this result means in our context.

COROLLARY 1. *Let \mathcal{P} be as above, and suppose it is finitely evaluable. Then the theory $\mathcal{T} + \text{comp}(\mathcal{P})$ has a unique model on \mathbb{R} . In this model all the (real parts of the) primitive predicates are represented by semialgebraic sets. Actually it suffices to require that for all fluents $f(\overline{x})$ in the scenario, the query $?HoldsAt(f(\overline{x}), t)$ is finitely evaluable.*

****PROOF.** We prove the stronger statement. Since the scenario is finite, it mentions only finitely many fluents (possibly containing parameters for reals or individuals). By the definition of scenario, in $\mathcal{T} + \text{comp}(\mathcal{P})$ every primitive predicate can be defined in terms of *HoldsAt* and relations and functions from the constraint language \mathcal{L} ¹⁰. We therefore have to consider only computations involving $HoldsAt(f(\overline{x}), t)$. Since the derivation tree for $?HoldsAt(f(\overline{x}), t)$ is finite, by the previous theorem $\mathcal{T} + \text{comp}(\mathcal{P})$ implies that $HoldsAt(f(\overline{x}), t)$ is equivalent to a constraint and hence this formula is definable in terms of semialgebraic sets and finitely many constants. It then follows from the results of Stuckey [117], Section 6, which link derivations to fixpoints of a consequence operator, that the model determined by the answer to the queries $?HoldsAt(f(\overline{x}), t)$ (for each f), is a model of $\mathcal{T} + \text{comp}(\mathcal{P})$, which is unique. ****** \square

COROLLARY 2. *Let \mathcal{P} be as above. The following are equivalent:*

- (a) *any model of $\mathcal{T} + \text{comp}(\mathcal{P})$ is completely determined by its restriction to \mathbb{R} ;*
- (b) *all sets defined by a formula of the form $HoldsAt(f(\overline{x}), t)$ are semialgebraic.*

****PROOF.** The direction from (a) to (b) follows from the previous corollary. The converse direction follows from Beth's definability theorem and Tarski's theorem. ****** \square

There is also a corresponding completeness result.

¹⁰Here it is essential that no primitive predicate occurs both in the head and the body of the same clause.

THEOREM 5. *Let \mathcal{P} be as above. The query $?HoldsAt(f(\bar{x}), t)$ is finitely evaluable if $\mathcal{T}+comp(\mathcal{P}) \models \forall t (HoldsAt(f, t) \leftrightarrow c(t))$ for a constraint c .*

****PROOF.** The hypothesis implies that both $\mathcal{T}+comp(\mathcal{P}) \models \forall t (c(t) \rightarrow HoldsAt(f, t))$ and $\mathcal{T}+comp(\mathcal{P}) \models \forall t (\neg c(t) \rightarrow \neg HoldsAt(f, t))$. By theorem 8.4 in [117], the goals $?c, HoldsAt$ and $? \neg c, \neg HoldsAt$ succeed. By lemma 5.6(a) in [117] the goal $?HoldsAt$ has a successful derivation ending in a constraint implied by c , and analogously for $? \neg HoldsAt$.** \square

The hypothesis of finite evaluability is fairly strong; it calls for scenarios which are sufficiently complete. Although it would be pleasant to have a theorem indicating which scenarios lead to terminating computations, in general partiality seems to be an inherent feature of natural language: there is no reason why the lexical and temporal information embodied in the scenario is sufficiently exhaustive to make the scenario complete. In fact, even if a scenario is complete, computations may well stop before a full model has been determined, thus yielding a partial model only. However, since the linguistic examples that we discuss all involve complete scenarios, leading to finitely evaluable queries, we will not introduce the machinery necessary for dealing with the general case. Instead, we discuss by means of an example a possible obstacle to the completeness of a scenario.

**We started out this Section by asking how the structure of the denotation of an expression is affected by the structure of the scenario and the axioms of the event calculus. One may observe that the above proofs would no longer go through literally if we were to allow in scenarios formulas of the form $S(t, t') \wedge Happens(e', t') \rightarrow Happens(e, t)$. We will now investigate this matter in slightly greater detail. Consider a scenario containing

- (1) $Happens(e, 0)$
- (2) $Happens(e, t') \wedge t = t' + 2 \rightarrow Happens(e, t)$.

Consider the query $?Happens(e, s)$, $0 < s < 1$. This query leads to an infinite derivation. It can be made finite by adding the integrity constraint¹¹ that $?Happens(e, s)$, $0 > s$ must fail. In this case the interpretation of $Happens(e, s)$ is the set of even natural numbers, which is not semialgebraic. Extending the scenario to

- (1) $Happens(e, 0)$
- (2) $Happens(e, t') \wedge t = t' + 2 \rightarrow Happens(e, t)$
- (3) $Happens(e', 1)$
- (4) $Happens(e', t') \wedge t = t' + 2 \rightarrow Happens(e', t)$
- (5) $Initiates(e', f, t)$
- (6) $Terminates(e, f, t)$

will make f true on the intervals (1,2), (3,4), (5,6) . . . , and again this set is not semialgebraic. Next, consider a scenario containing

- (1) $Happens(e, 0)$
- (2) $Happens(e, 1)$

¹¹A concept that will be discussed more fully in Section 1 of Chapter 8.

- (3) $\neg \text{Happens}(e, \frac{1}{3})$
- (4) $\text{Happens}(e, t') \wedge \text{Happens}(e, t'') \wedge t = \frac{t'+t''}{2} \rightarrow \text{Happens}(e, t)$
- (5) $\text{Happens}(e', t') \wedge \text{Happens}(e', t'') \wedge t = \frac{t'+t''}{2} \rightarrow \text{Happens}(e', t)$
- (6) $\neg \text{Happens}(e, t) \rightarrow \text{Happens}(e', t)$
- (7) $\neg \text{Initially}(f)$
- (8) $\text{Initiates}(e, f, t)$
- (9) $\text{Terminates}(e', f, t)$.

The events e, e' occur on a dense set of points in the interval $[0,1]$; furthermore we have $\neg \text{HoldsAt}(f, 0)$. But what can one say about the behaviour of $\text{HoldsAt}(f, t)$ on $(0,1]$? Suppose the derivation tree for $? \text{HoldsAt}(f, t)$, $0 \leq t \leq 1$ would contain a successful branch, then there would be a constraint c such that $\forall t(0 \leq t \leq 1 \wedge c(t) \rightarrow ? \text{HoldsAt}(f, t))$. But since c determines a finite set of intervals, there would be an interval contained in $(0,1]$ on which f holds. This however is impossible, since there occur many terminating events inside this interval (because e' occurs on a dense set of points). Similarly there cannot be a branch on which $? \text{HoldsAt}(f, t)$, $0 \leq t \leq 1$ fails finitely. It follows that $\text{HoldsAt}(f, t)$ is undefined on $(0,1]$; we are concerned here with a truly partial model.**

We have seen that allowing *Happens* to occur in both head and body of a clause may lead to vicious loops. Sometimes this syntactic form is harmless however, namely when the scenario contains formulas of the form given in definition 9.

We continue with an example of a complete scenario, leading to finitely evaluable queries only.

3. How to get to the other side of a street

As an application of the preceding material, we now discuss in greater detail the scenario for the sentence ‘John was crossing the street’, as given in Section 3.1. What follows from this sentence if we restrict ourselves to minimal models, that is, models in which nothing unexpected happens? Intuitively, it should follow that the intended goal will be reached.

As explained above, a minimal model of a logic program is defined via the completion of the program. Here, the logic program consists of the full scenario plus the axioms for the event calculus. Giving the full completion of the program would take up too much space, so we do only a few cases and ask the reader to write down the full completion.

First consider the predicate *Initiates*. There is only one formula in the scenario with *Initiates* in the head, namely $\text{Initiates}(\text{start}, \text{crossing}, t)$. Writing this in canonical form gives $f = \text{crossing} \wedge e = \text{start} \rightarrow \text{Initiates}(e, f, t)$. We now want to say that this is the only causal effect of an event onto a fluent: this is achieved by replacing the implication with an equivalence, as follows

$$f = \text{crossing} \wedge e = \text{start} \leftrightarrow \text{Initiates}(e, f, t).$$

Next consider the predicate *Happens*. The scenario contains two clauses with *Happens* in the head, namely $Happens(start, t_0)$ and $HoldsAt(crossing, t) \wedge HoldsAt(distance(m), t) \rightarrow Happens(reach, t)$. Writing these in canonical form gives

- (1) $e = start \wedge t = t_0 \rightarrow Happens(e, t)$
- (2) $e = reach \wedge HoldsAt(crossing, t) \wedge HoldsAt(distance(m), t) \rightarrow Happens(e, t)$

Combining these two statements into a conjunction gives

- (3) $(e = start \wedge t = t_0) \vee (e = reach \wedge HoldsAt(crossing, t) \wedge HoldsAt(distance(m), t)) \rightarrow Happens(e, t)$

Now comes the crucial step: we want to say that an event can *happen* only in virtue of the antecedent of the preceding formula; there are no unforeseen events. This is achieved by replacing the implication by an equivalence:

- (4) $(e = start \wedge t = t_0) \vee (e = reach \wedge HoldsAt(crossing, t) \wedge HoldsAt(distance(m), t)) \leftrightarrow Happens(e, t)$

The preceding formulas allow us to write, using the completion for *HoldsAt*, which is then specialized to $HoldsAt(crossing, now)$:

$$\begin{aligned} & HoldsAt(crossing, now) \\ & \quad \updownarrow \\ & \exists t_0 < now (Happens(start, t_0) \wedge Initiates(start, crossing, t_0) \\ & \quad \wedge \neg Clipped(t_0, crossing, now)). \end{aligned}$$

Note that we have used the fact that $Initially(crossing)$ is false in the completion. Since we know that $HoldsAt(crossing, now)$, it follows that for some time $t_0 < now$, $Happens(start, t_0)$. For this particular t_0 we furthermore have:

$$(*) HoldsAt(crossing, t) \leftrightarrow \neg Clipped(t_0, crossing, t).$$

It follows that *crossing* defines an interval of the form $(t_0, t_1]$ or (t_0, ∞) . We now want to show that for $t_1 = t_0 + m$: $Happens(reach, t_1)$ and $HoldsAt(distance(m), t_1)$. By axiom 2 it follows that $HoldsAt(distance(m), t) \wedge s > t \rightarrow HoldsAt(distance(m), s)$.

From this and property (*) above it follows that for all $t > t_1$: $HoldsAt(distance(m), t)$. Suppose there is no t such that $HoldsAt(distance(m), t) \wedge HoldsAt(crossing, t)$, then for no t , $Happens(reach, t)$, and as a consequence *crossing* is never *terminated*. But since *distance* represents a continuous monotonically increasing function, there will be a t such that¹² $HoldsAt(distance(m), t) \wedge HoldsAt(crossing, t)$, a contradiction. Hence there exists a (unique) t_1 for which $HoldsAt(distance(m), t_1) \wedge HoldsAt(crossing, t_1) \wedge Happens(reach, t_1)$.

By axiom 4, $HoldsAt(distance(t_1 - t_0), t_1)$, and it follows that $t_1 - t_0 = m$, using property 9 of the scenario, or the fact that *HoldsAt* is a truth predicate.

¹²Here we use the fact that *HoldsAt* is a truth predicate.

4. **When do causes take effect?

As an application of the notion of minimality, we are now able to clarify why a fluent is required not to hold at the moment it is initiated. Consider axiom 3, repeated for convenience:

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

In a minimal model as constructed above, the interpretation of *HoldsAt* must satisfy $Happens(e, t) \wedge Initiates(e, f, t) \wedge t < t' \wedge \neg Clipped(t, f, t') \leftrightarrow HoldsAt(f, t')$. Suppose first that e happens instantaneously. It follows immediately that f cannot hold at the time it is initiated. A similar argument shows that f must hold at the time it is terminated. We emphasize that this argument only works in the minimal model¹³. Nothing in the setup of the event calculus as presented here requires that events occur instantaneously. Even if events and fluents are both extended in time, they are distinguished by the roles they play in the event calculus. It is worth bearing this in mind when thinking about linguistic examples. For example, it is sometime maintained that in French, the *Passé Simple* has the effect of making the event described punctual, to be placed inside a temporally extended background provided by a sentence in the *Imparfait*. As against this, Comrie [17] rightly argues that what matters is not punctuality, but lack of internal structure. The *Passé Simple* presents an event as a whole, disregarding internal structure. Of course points do not have internal structure, but punctuality is not a necessary requirement for lack of structure. If the event e is temporally extended, the completion of axiom 3 has the effect of forcing f to be true as soon as there has been an instant such that $Happens(e, t) \wedge Initiates(e, f, t)$. Similarly, the completion of axiom 8 makes f false as soon as there has been an instant such that $Happens(e, t) \wedge Terminates(e, f, t)$. For some, though not all, examples this is somewhat unnatural. E.g. when turning a light on, the light is on only after the end of the action. It is possible to rephrase axiom 3 so that it has this effect:

$$\text{AXIOM 7. } Happens(e, t) \wedge Initiates(e, f, t) \wedge t < t' \wedge \forall s (Happens(e, s) \rightarrow s < t') \wedge \neg Clipped(t, f, t') \rightarrow HoldsAt(f, t').$$

However, since such stipulations depend on the actions considered, we shall refrain from complicating the axioms in this way.**

The next Section contains a number of exercises to allow the reader to check her understanding of the preceding material.

¹³This technical analysis should not obscure the fact that there is a deep philosophical problem behind this, akin to Zeno's paradoxes of motion. If we assume infinite divisibility of time, there cannot be *both* a last instant at which f does not hold, *and* a first instant at which f holds. We therefore have to choose. For a philosophical justification of the solution adopted here, see Le Poidevin [70, p. 111].

5. Exercises for chapters 4 and 5

Our first exercise concerns cause as instantaneous change. Consider a room in which there are two lights and two switches, each serving one light. We have the following event types and fluents:

- (1) $a_1 = \text{switch}_1 \text{ on}$
- (2) $a_2 = \text{switch}_1 \text{ off}$
- (3) $e_1 = \text{switch}_2 \text{ on}$
- (4) $e_2 = \text{switch}_2 \text{ off}$
- (5) $f_1 = \text{light}_1 \text{ on}$
- (6) $f_2 = \text{light}_2 \text{ on}$

At time 5 light 1 is switched on, at time 10 it is switched off. Light 2 is on initially.

EXERCISE 1. Describe the light situation at times t for $0 \leq t \leq 15$. Does anything change if we do not specify anything about light 2?

The next exercises concern cause as continuous change. Suppose we want to show that a bucket of height 10 units, into which water flows continuously, will ultimately overflow. This can be formalised by assuming fluents *filling*, a fluent function $\text{height}(x)$ (where $x \in \mathbb{N}$), and events *overflow*, *tap-on*, *tap-off* which are connected by axioms such as the following (this list is not exhaustive!)

- (1) $\text{Initially}(\text{height}(0))$
- (2) $\text{Happens}(\text{tap-on}, 5)$
- (3) $\text{Initiates}(\text{tap-on}, \text{filling}, t)$
- (4) $\text{Terminates}(\text{overflow}, \text{filling}, t)$
- (5) $\text{HoldsAt}(\text{height}(10), t) \wedge \text{HoldsAt}(\text{filling}, t) \rightarrow \text{Happens}(\text{overflow}, t)$
- (6) $\text{Releases}(\text{tap-on}, \text{height}(0), t)$.

EXERCISE 2. Complete the specification under the assumption that after d units of time the water level has also increased d units; this gives a scenario.

EXERCISE 3. Try to show that $\text{HoldsAt}(\text{height}(10), 20)$. What is the role of axiom 2?

EXERCISE 4. Suppose we would add the statement $\text{Happens}(\text{tap-off}, 10)$ to the above list. Would $\text{HoldsAt}(\text{height}(10), 20)$ still be derivable?

EXERCISE 5. What would happen if the statement $\text{Releases}(\text{tap-on}, \text{height}(0), t)$ were left out?

EXERCISE 6. Give a complete description of the model of EC plus the scenario; that is, describe what happens when, and which fluents are true or false at which time points.

EXERCISE 7. Shanahan uses a more elaborate version of the event calculus which also features the following two axioms to deal with negative information:

AXIOM 8. $Happens(e, t) \wedge Terminates(e, f, t) \wedge t < t' \wedge \neg Declipped(t, f, t') \rightarrow \neg HoldsAt(f, t')$

AXIOM 9. $Happens(e, s) \wedge t < s < t' \wedge (Initiates(e, f, s) \vee Releases(e, f, s)) \rightarrow Declipped(t, f, t')$.

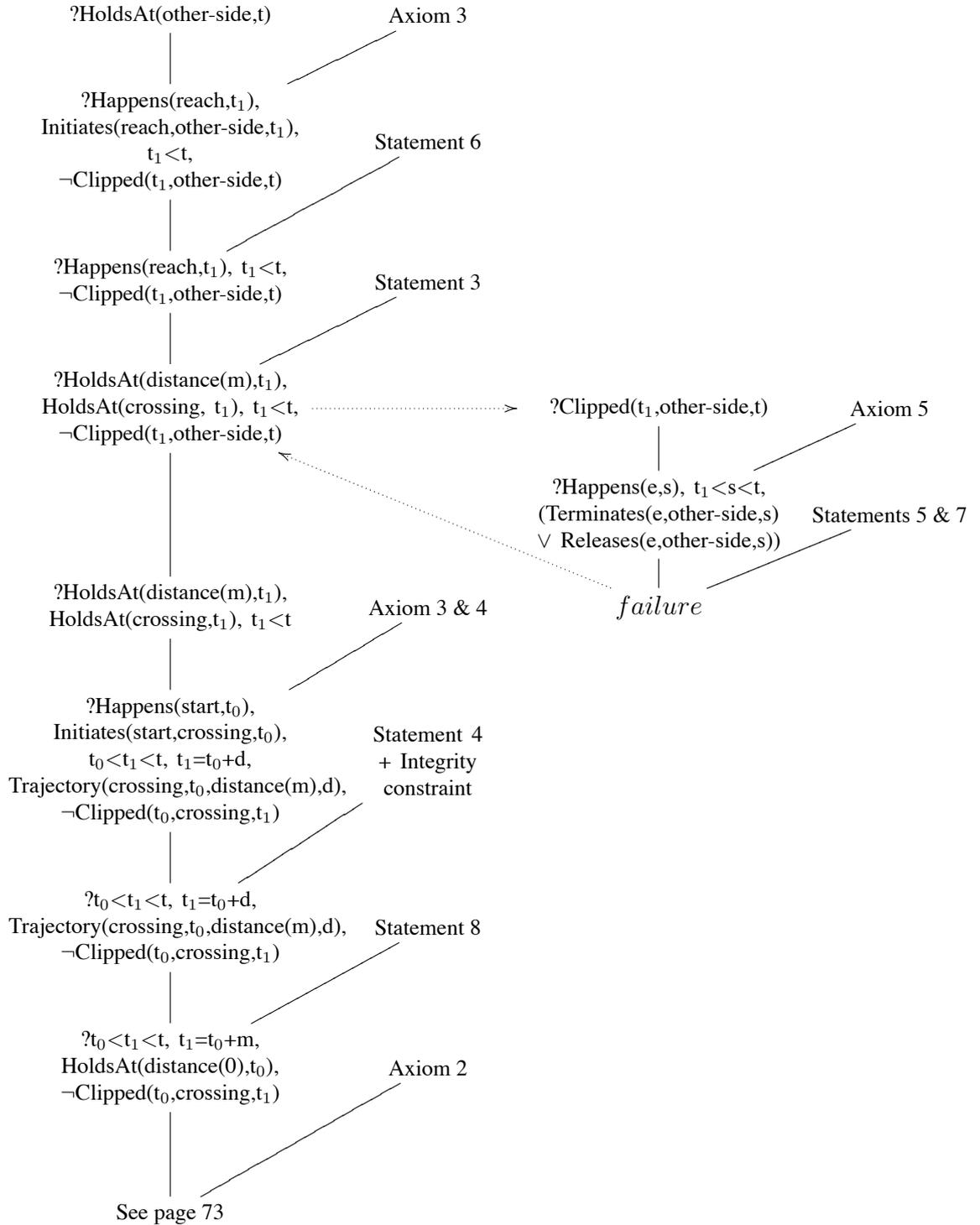
In our setup these axioms are superfluous in the following sense. First, the predicate *Declipped* in axiom 8 can be replaced by its definition using axiom 9. Second, the effect of axiom 8 can be captured in the following form¹⁴: if the query $?Happens(e, t) \wedge Terminates(e, f, t) \wedge t < t' \wedge \neg Declipped(t, f, t')$ succeeds, then so does $? \neg HoldsAt(f, t)$. Try to prove this.

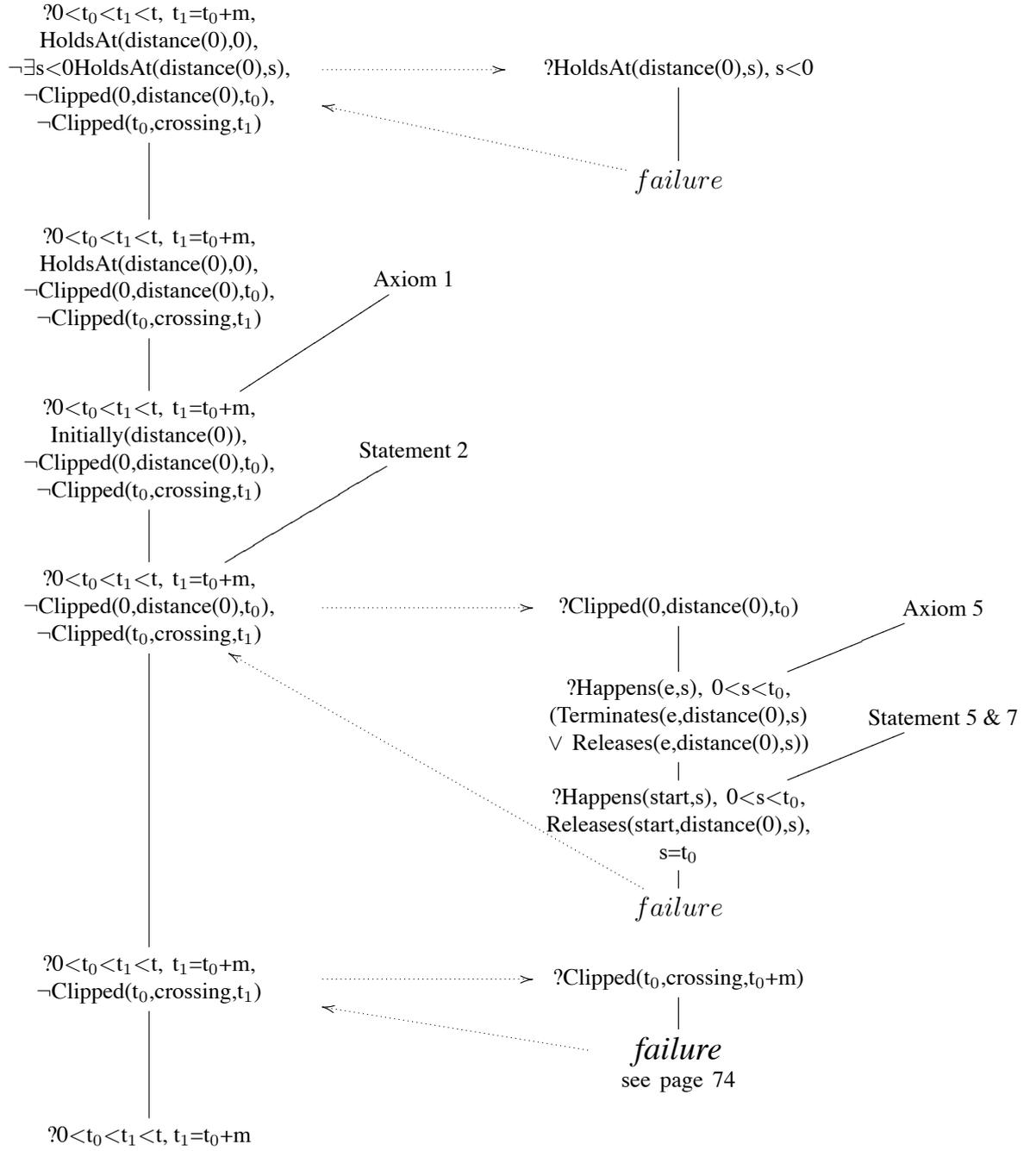
(Hint: You will need the following constraints on the scenario. First, no event should both initiate and terminate the same fluent. Second, no event type must occur at time $t = 0$; the instant 0 therefore acts as a kind of $-\infty$. Note that the joint presence of *Initiates*(e, f, t) and *Terminates*(e, f, t) does not cause outright inconsistency of the scenario, precisely because axiom 8 is not in *EC*, thus introducing an asymmetry between *Initiates* and *Terminates*. Symmetry is restored by means of the above integrity constraint. Consistency of scenarios will be studied further in Section 2, where we also present another way of bringing axiom 8 into a form consonant with logic programming.)

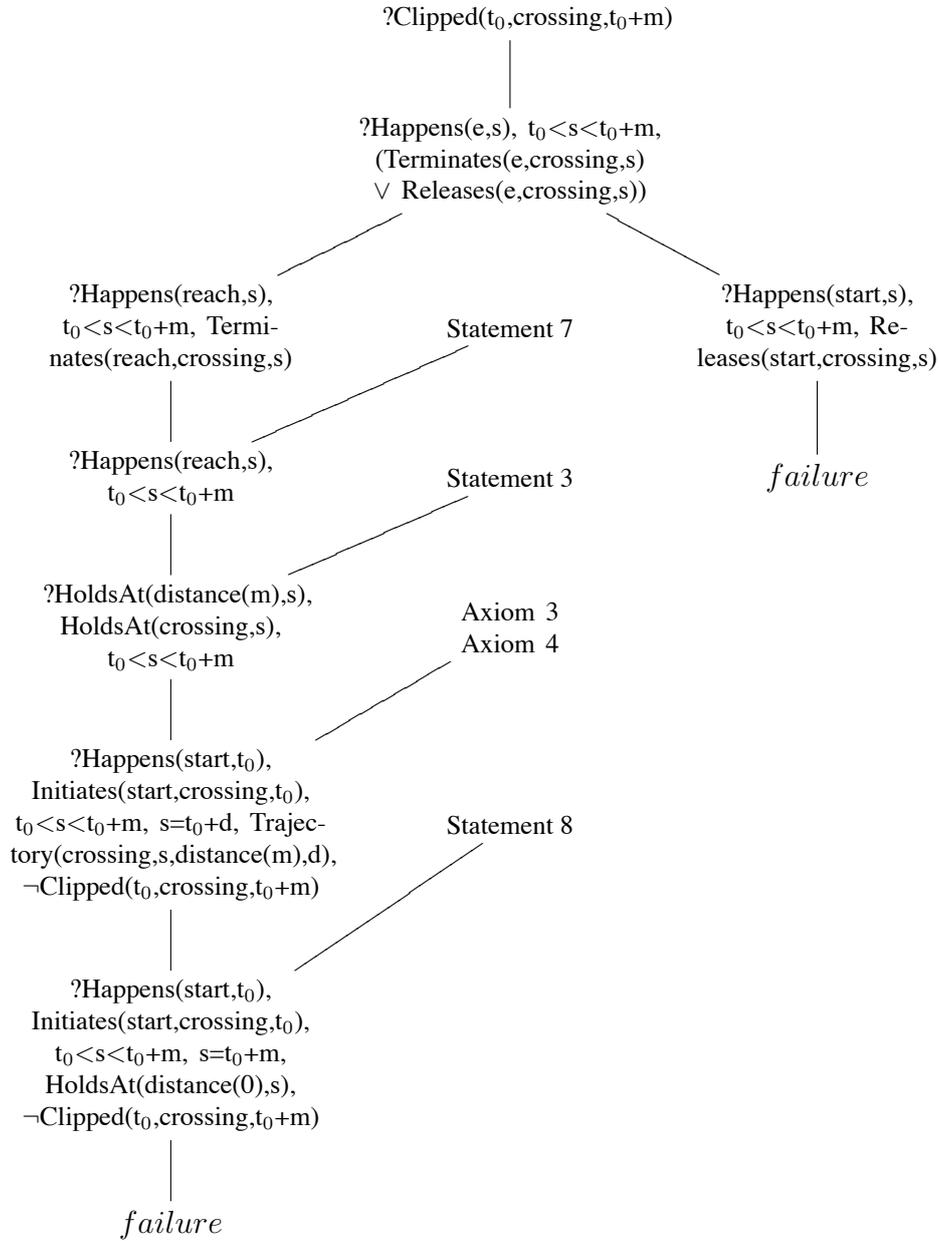
6. Da capo, with feeling

In Section 3 we gave a semantic derivation of the fact that the other side of the street will be reached. We will now go over the same ground again, but use logic programming instead. The resulting derivation is admittedly formidable, but we give it nonetheless, to convince the reader that ours is a fully computational approach.

¹⁴Statements of this form are known as *integrity constraints*; they will play an important role in our formalization of Reichenbach's reference time (see Section 1 of Chapter 8).







CHAPTER 6

Finishing touches

In the preceding two chapters we have used natural language expressions to represent fluents and events; we have for example encountered a fluent *crossing* and an event *reach*. So far these expressions have been mere labels, but we will now show that this labelling procedure can be formalized. On the linguistic side, this formalization corresponds to nominalization. This topic will be treated extensively in chapter 12, but we should say immediately that nominalization plays a fundamental role in our setup, much more so than in other approaches to (formal) linguistics. The reason is that nominalized VPs are the basic units of semantic computation, in their guise as fluents and events. Therefore nominalization is not just a particular linguistic construction, on a par with others; it also plays a fundamental role in other constructions such as tense and aspect. In this chapter we will lay the groundwork for a formal treatment of nominalization. To this end, we must provide a coding mechanism which transforms formulas into objects, and we also need a theory of truth which relates the interpretations of formulas and their coded versions. Indeed, the main truth predicate that we need is the *HoldsAt* predicate in *EC*.

1. Coding VPs as fluents and events

As a starting point for coding we need a sufficiently rich formal language which semantically represents natural language. Our choice reflects some of the basic ideas underlying this approach. We do not take Montague grammar because we believe intensional phenomena should not be treated via possible worlds; the chapters on aspect (10, in particular Section 2) and on coercion (11) give an alternative treatment. We also do not take dynamic semantics: although we believe that there are many dynamic aspects to meaning (cf. Chapter 1), we also believe that it is best to model computations explicitly, instead of abstractly as is done in dynamic semantics. Lastly, while we agree with DRT that the task of semantics is first of all to construct discourse representations, we object to the use of a Davidsonian event argument together with predicates corresponding to thematic roles, because this device is neither capable of representing the structure inherent in events, nor of the different perspectives on events. Instead of an event argument we use an argument for time, because this will allow us to construct the various kinds of events that we need.

We therefore start from an extensional two-sorted first order language in which predicates may have a time parameter, usually represented by the

variables r, s, t . Unfortunately, for lack of space we will also not discuss the very important topic of argument structure, for the most part assuming that subject and object positions are just slots in a predicate (but see Chapter 7 and Section 4 of Chapter 12). It is of course not as simple as that, and as an illustration we give, in Chapter 7, an extensive discussion of the so-called ‘incremental theme’, an example of which is the expression ‘a house’ in ‘build a house’.

We shall borrow some machinery from [38], which was applied in [47] to the semantics of nominalization. A basic feature of nominalization is that it turns VPs into nouns, which may furthermore occur as argument in a predicate, as in

- (1) a. The crossing of the river occurred on April 1.
 b. The crossing of the river proved to be impossible.
 c. Crossing the river was difficult.

Since the nominalizations ‘the crossing of the river’ and ‘crossing the river’ can be an argument of the predicates ‘occur’ and ‘be difficult’, it is useful to have these expressions denote objects in the domain of a first order model. In this way, a complicated hierarchy of types can be avoided. Fortunately the languages that we work with are sufficiently rich to allow for the coding of formulas as terms.

Let L_0 be some first order language, extending the language of the reals, and S_0 a theory formulated in L_0 , containing at least axioms for $+$ and \times . This setup is strong enough to allow formulas of L_0 to be coded as natural numbers, hence as terms of L_0 , via the device of Gödel numbering. It is furthermore possible to define a binary pairing function π in L_0 , together with two unary projection functions π_1 and π_2 ¹. We shall often write (τ_1, τ_2) for $\pi(\tau_1, \tau_2)$, and (x, y) for $\pi(x, y)$. By definition, these functions are related as follows:

$$(2) \quad \pi_1(x, y) = x \wedge \pi_2(x, y) = y$$

One may now define tuples inductively by putting: $(\tau) = \tau$ and $(\tau_1, \dots, \tau_{k+1}) = ((\tau_1, \dots, \tau_k), \tau_{k+1})$. Similarly, one may define the corresponding projection operations π_i^k ($1 \leq i \leq k$) such that: $\pi_i^k(x_1, \dots, x_k) = x_i$.

DEFINITION 17. *We write $\ulcorner \varphi \urcorner$ for the Gödel number in L_0 of φ in L_0 . This notation will be used interchangeably both for the term in L_0 and for the object denoted by that term in a model M_0 for L_0 .*

¹The details of the Gödel numbering are of no importance; the interested reader may consult Boolos and Jeffrey [10]. One example of a pairing function is $\pi(x, y) := \frac{1}{2}(x + y)(x + y + 1) + x$, but again the details do not matter.

We will now put this machinery to work. Let φ be a formula with free variables among $x_1, \dots, x_k, y_1, \dots, y_n$. The L_0 -term $(\ulcorner \varphi \urcorner, y_1, \dots, y_n)$ contains x_1, \dots, x_k as bound variables and y_1, \dots, y_n as free variables. Since x_1, \dots, x_k are bound by abstraction, the following notation makes sense:

DEFINITION 18. $\Delta_n \varphi[\hat{x}_1, \dots, \hat{x}_n, y_1, \dots, y_m] = (\ulcorner \varphi \urcorner, y_1, \dots, y_m)$.
For $n = 1$ we will use standard set theoretical notation $\Delta_1 \{x \mid \varphi(x, y_1, \dots, y_n)\} = \varphi[\hat{x}, y_1, \dots, y_n]$. If both m and n are equal to 0, we write $\ulcorner \varphi \urcorner$.

Finally, we are able to explain what all this has to do with the event calculus. Assume that some predicates in L_0 have a parameter for time. For instance, the verb *run* could be represented in L_0 as $run(x, t)$. Thus, as noted, unlike event semantics in the Davidsonian tradition, we do not assume that verbs have an event parameter; rather, various kinds of eventualities will be constructed from the verb with its time parameter. Of particular importance are the *event type*², given formally by $\exists t.run[x, t]$, and the *fluent*³ $run[x, \hat{t}]$. Note the difference between the two abstractions: if a concrete individual j (for ‘John’) is substituted for x , the second is a function of time (to truth values), while the first is an object. The fluent $run[x, \hat{t}]$ may occur as an argument of *HoldsAt*, but not of *Happens*, while for the event type $\exists t.run[x, t]$ it is the reverse. One last remark before we move on to a discussion of truth: if $\varphi(x)$ is a formula, Feferman’s coding trick allows one to introduce the set-like object $\varphi[\hat{x}]$, alternatively written as $\{x \mid \varphi(x)\}$. It is important to realize, however, that these sets are unlike classical sets in that they do not necessarily satisfy the axiom of extensionality

$$\forall x(x \in a \leftrightarrow x \in b) \rightarrow a = b.$$

That is, fluents may be extensionally equal, i.e. may have the same temporal profile without being intensionally identical. This will turn out to be a very useful feature.

2. Consistency, truth and partiality

In the proposed setup the basic computational entities are events and fluents, not formulas in the language L_0 , because it is the former that enter into causal relationships, not the latter. Nevertheless, the results must be able to be lifted to models for L_0 . For example, suppose we have used the fluent $run[j, \hat{t}]$ in a computation in the event calculus; this means that we have determined the set of s such that $HoldsAt(run[j, \hat{t}], s)$. We would now like to conclude that for the same set of s , $run(j, s)$. More generally, one would like to have for any formula $\varphi(t)$ of L_0 (possibly containing parameters):

$$(1) \quad HoldsAt(\varphi[\hat{t}], s) \leftrightarrow \varphi(s)$$

²Also called *perfect nominal*.

³Also called *imperfect nominal*.

Even this formulation is not general enough, for, as we shall see below, sometimes *HoldsAt* occurs in iterated form, so that the formula φ is itself of the form *HoldsAt*(f, t). But even equation 1 is a problematic principle for the event calculus, as we shall proceed to show by means of a few examples.

Let $A(s)$ be a predicate in L_0 , and define the fluents f_1, f_2 by $f_1 := A[\hat{s}]$, $f_2 := \neg A[\hat{s}]$. Now suppose one takes the scenario to be empty. By negation as failure one then has $\neg \text{Initially}(f_1)$, $\neg \text{Initially}(f_2)$, and for all t , the computations $?HoldsAt(f_1, t)$ and $?HoldsAt(f_2, t)$ fail. But the statement $\forall t(\neg HoldsAt(f_1, t) \wedge \neg HoldsAt(f_2, t))$ clearly contradicts the intended meaning of the fluents; a contradiction that becomes explicit upon applying principle 1.

For another example, let $\varphi(x, t)$ be any L_0 formula, and put $f_1 := \varphi[a, \hat{s}]$, $f_2 := \exists x \varphi[x, \hat{s}]$. Suppose the scenario consists of

- (1) *Initially*(f_2)
- (2) *Initiates*(e, f_1, t)
- (3) *Terminates*(e, f_2, t)
- (4) *Happens*($e, 1$)

We then have $HoldsAt(f_1, t) \leftrightarrow t \geq 1$ and $HoldsAt(f_2, t) \leftrightarrow t \leq 1$. This can be turned into an outright contradiction by applying principle 1 twice: $HoldsAt(\exists x \varphi[x, \hat{s}], t) \leftrightarrow \exists x \varphi(x, t) \leftrightarrow \exists x HoldsAt(\varphi[x, \hat{s}], t)$, from which it follows that $HoldsAt(f_1, t) \rightarrow HoldsAt(f_2, t)$. This contradicts what we derived about the behaviour of f_1 (true on $(1, \infty)$) and f_2 (true on $[0, 1]$).

These examples could be multiplied. They show that consistent scenarios⁴ can lead to computations which are uninterpretable.

Making *HoldsAt* play the role of a truth predicate also leads to a problem of a more general kind. Tarski has shown that one cannot consistently add a truth predicate to first order logic while keeping classical semantics. Roughly speaking, once one has a truth predicate and the possibility to code formulas as terms, it is possible to construct a sentence saying something like ‘I am false’, and this sentence can be neither true nor false. In our context, this problem may arise if we consider fluents which have been derived from formulas which itself contain the *HoldsAt* predicate. Iterated nominalization, as in ‘John’s supporting his son’s not going to church’, sometimes requires fluents of this kind.

We first give the general idea of adding a truth predicate; afterwards we discuss how this works out in our particular situation. To formalize the general truth definition, we add predicates T_n to L_0 . The intuitive meaning of $T_n(x_1, \dots, x_n, z)$ is: the tuple (x_1, \dots, x_n) satisfies (the formula coded by) z . This meaning can be codified in the following axiom scheme:

⁴Note that the scenarios are also consistent with principle 1; this can be seen by adding *Initially*(f_1) to the first scenario, and *Releases*(e, f_1, t) to the second. It is only the *completion* of scenario plus axioms of the event calculus which is inconsistent with that principle.

AXIOM 10. For all formulas φ in the language $L := L_0 \cup \{T_n \mid n \in \mathbb{N}\}$ and for all n

$$(T_n A) \quad T_n(x_1, \dots, x_n, \varphi[\hat{u}_1, \dots, \hat{u}_n, y_1, \dots, y_m]) \leftrightarrow \varphi(x_1, \dots, x_n, y_1, \dots, y_m)$$

In the formula φ , the variables u_1, \dots, u_n are bound, and the y_1, \dots, y_m are free. The latter variables are the parameters of a particular instance of the scheme. The free variables x_1, \dots, x_n occurring in T_n have to be substituted for the bound variables u_1, \dots, u_n of φ .

Important special cases are the axioms for T_0

$$(3) \quad \begin{array}{l} \text{a.} \\ \quad T_0(\varphi[y_1, \dots, y_m]) \leftrightarrow \varphi(y_1, \dots, y_m) \\ \text{b. For } m = 0: \\ \quad T_0(\ulcorner \varphi \urcorner) \leftrightarrow \varphi \end{array}$$

and for T_1 ,

$$(4) \quad T_1(x, \varphi[\hat{u}, y_1, \dots, y_m]) \leftrightarrow \varphi(x, y_1, \dots, y_m)$$

T_1 will be of special importance for us, since the *HoldsAt* predicate of the event calculus is a special case of T_1 . Most of the formal development will therefore be formulated in terms of T_1 ; generalization to all T_n is immediate. We shall also assume that, whenever necessary, occurrences of *HoldsAt* in scenario and axioms of the event calculus are replaced with T_1 . Thus consistency of scenario and axioms with the logic program for T_1 given below is nontrivial.

AXIOM 11. We reformulate (4) as a logic program.

The logic program \mathcal{T}_1 is characterized by the following clauses

- (1) $T_1(x, \varphi[\hat{u}, \bar{y}]) \wedge T_1(x, \psi[\hat{u}, \bar{y}]) \rightarrow T_1(x, (\varphi \wedge \psi)[\hat{u}, \bar{y}])$
- (2) $T_1(x, \psi[\hat{u}, \bar{y}]) \rightarrow T_1(x, \varphi \vee \psi[\hat{u}, \bar{y}])$
- (3) $T_1(x, \varphi[\hat{u}, \bar{y}]) \rightarrow T_1(x, \varphi \vee \psi[\hat{u}, \bar{y}])$
- (4) $\neg T_1(x, \varphi[\hat{u}, \bar{y}]) \rightarrow T_1(x, \neg\varphi[\hat{u}, \bar{y}])$
- (5) $T_1(x, \varphi[\hat{u}, \bar{y}z]) \rightarrow T_1(x, \exists z\varphi[\hat{u}, \bar{y}z])$
- (6) $\neg\exists z\neg T_1(x, \varphi[\hat{u}, \bar{y}z]) \rightarrow T_1(x, \forall z\varphi[\hat{u}, \bar{y}z])$
- (7) $T_1(x, \varphi[\hat{u}, \bar{y}]) \rightarrow T_1(x, T_1[\hat{v}, \varphi[\hat{u}, \bar{y}]])$

The logic program \mathcal{T}_1^- is \mathcal{T}_1 with the condition 7 deleted.

The logic program \mathcal{T}_1^+ is \mathcal{T}_1 with the addition of the following clause, for all atomic formulas $A(x, \bar{y})$

$$T_1(x, A[\hat{u}, \bar{y}]) \rightarrow A(x, \bar{y}).$$

Note that \mathcal{T}_1 is a correct constraint logic program in the sense of definition 12 of Chapter 5. Also, the fluents occurring on the right hand side of the clauses belonging to \mathcal{T}_1 are of greater syntactic complexity than those occurring on the left hand side. This immediately leads to the following observation.

LEMMA 2. *Let SCEN be a scenario in which all fluents are derived from atomic formulas of L_0 . Then the completion of the logic program consisting of SCEN, the axioms of the event calculus and T_1^- is consistent.*

This easy observation is deliberately formulated in terms of the theory T_1^- , which does not allow iterations of the truth predicate. In the general case, the presence of the liar formula causes considerable problems. It follows that computations performed on fluents can be interpreted as applying to real formulas, not only to the fluents that are derived from them.

COROLLARY 3. *A model \mathcal{M} of SCEN + EC which also satisfies T_1^- may be expanded to a model \mathcal{N} of L_0 in the following manner: if a predicate $A(x, t)$ corresponds to a fluent $f(x)$ in the scenario, interpret $A(x, t)$ on \mathcal{N} by the set $\{(x, t) \mid \text{HoldsAt}(f(x), t)\}$; otherwise let the interpretation of A be arbitrary.*

In this way the validity of axiom 11 is ensured, and therefore also the axiom scheme (4). Since the hypothesis of the lemma is that the scenario only has information about fluents derived from atomic formulae, the behaviour of complex formulae on \mathcal{N} is trivially consistent with the predictions of the scenario.

If one does not assume that the only fluents occurring in the scenario are those derived from atomic formulas, the preceding considerations must be formulated as follows

LEMMA 3. *Suppose the completion of the logic program consisting of SCEN, the axioms of the event calculus and T_1^- is consistent. Then a two-valued model \mathcal{M} can be expanded to a two-valued model \mathcal{N} for L_0 also satisfying the axiom scheme (4).*

We now have to consider how to incorporate iterated applications of T_1 , that is, we must consider how to extend SCEN + EC with T_1 . Because of Tarski's theorem on the undefinability of truth, we can no longer expect the preceding lemma to go through; there may not be a two-valued model of the extension. One way out, also used in logic programming, is to resort to Kleene's three-valued logic instead of two-valued classical logic. We prefer to remain in the classical realm, and work with positive and negative extensions of the truth predicate T_1 , which will henceforth be denoted by T_1 and \bar{T}_1 , respectively. These predicates are mutually exclusive, but their union does not necessarily exhaust the domain. T_1 will correspond to a set of successful queries, whereas \bar{T}_1 corresponds to a set of finitely failing queries. This ensures that the predicates are mutually exclusive, i.e.

$$(2) \quad \forall x, y \neg (T_1(x, y) \wedge \bar{T}_1(x, y))$$

but since some queries, in particular those involving the liar formula, may lead to infinite derivations, one does not have

$$(3) \quad \forall x, y(T_1(x, y) \vee \bar{T}_1(x, y))$$

Thus \bar{T}_1 functions as an antonym to T_1 . We now have to expand axiom 11 in terms of the predicates T_1 and \bar{T}_1 . That the expansion given below is indeed correct follows from the results in Stärk [106, pp. 453–462]; we omit the proof. The only important point to remember here is that iterated use of the truth predicate calls for an antonymic negation; in other words, that we are now concerned with an essential form of partiality. This observation will be put to use in chapter 12, on nominalization.

AXIOM 12. *The logic program $T_1(+/-)$ is characterized by the following clauses*

- (1) $T_1(x, \varphi[\hat{u}, \bar{y}]) \wedge T_1(x, \psi[\hat{u}, \bar{y}]) \rightarrow T_1(x, (\varphi \wedge \psi)[\hat{u}, \bar{y}])$
- (2) $\neg T_1(x, \varphi[\hat{u}, \bar{y}]) \wedge \neg T_1(x, \psi[\hat{u}, \bar{y}]) \rightarrow \bar{T}_1(x, (\varphi \vee \psi)[\hat{u}, \bar{y}])$
- (3) $\neg T_1(x, \varphi[\hat{u}, \bar{y}]) \rightarrow \bar{T}_1(x, \varphi[\hat{u}, \bar{y}])$
- (4) $T_1(x, \psi[\hat{u}, \bar{y}]) \rightarrow T_1(x, \varphi \vee \psi[\hat{u}, \bar{y}])$
- (5) $T_1(x, \varphi[\hat{u}, \bar{y}]) \rightarrow T_1(x, \varphi \vee \psi[\hat{u}, \bar{y}])$
- (6) $\neg T_1(x, \psi[\hat{u}, \bar{y}]) \rightarrow \bar{T}_1(x, \varphi \wedge \psi[\hat{u}, \bar{y}])$
- (7) $\neg T_1(x, \varphi[\hat{u}, \bar{y}]) \rightarrow \bar{T}_1(x, \varphi \wedge \psi[\hat{u}, \bar{y}])$
- (8) $T_1(x, \varphi[\hat{u}, \bar{y}z]) \rightarrow T_1(x, \exists z\varphi[\hat{u}, \bar{y}z])$
- (9) $\neg T_1(x, \varphi[\hat{u}, \bar{y}z]) \rightarrow \bar{T}_1(x, \forall z\varphi[\hat{u}, \bar{y}z])$
- (10) $\neg\exists z\neg T_1(x, \varphi[\hat{u}, \bar{y}z]) \rightarrow T_1(x, \forall z\varphi[\hat{u}, \bar{y}z])$
- (11) $\neg\exists zT_1(x, \varphi[\hat{u}, \bar{y}z]) \rightarrow \bar{T}_1(x, \exists z\varphi[\hat{u}, \bar{y}z])$
- (12) $T_1(x, \varphi[\hat{u}, \bar{y}]) \rightarrow T_1(x, T_1[\hat{v}, \varphi[\hat{u}, \bar{y}]])$
- (13) $\neg T_1(x, \varphi[\hat{u}, \bar{y}]) \rightarrow \bar{T}_1(x, T_1[\hat{v}, \varphi[\hat{u}, \bar{y}]])$

The logic program $T_1^+(+/-)$ is $T_1(+/-)$ with the addition of the following clauses, for all atomic formulas $A(x, \bar{y})$

- (14) $T_1(x, A[\hat{u}, \bar{y}]) \rightarrow A(x, \bar{y})$
- (15) $\bar{T}_1(x, A[\hat{u}, \bar{y}]) \rightarrow \neg A(x, \bar{y})$

The main observation then is

LEMMA 4. *Suppose SCEN only contains fluents derived from atomic formulas (possibly including T_1, \bar{T}_1). Suppose EC + SCEN is consistent with $T_1(+/-)$. Then a model of EC + SCEN + $T_1(+/-)$ can be expanded to a model of EC + SCEN + $T_1^+(+/-)$. Thus, the results of computations involving fluents can be applied to the corresponding formulas.*

An analogous result can be formulated without the first supposition; we leave this to the reader.

We conclude this Section with a simple application of the above material. In exercise 7 we mentioned that Shanahan uses the following two axioms to deal with negative information:

$$\text{AXIOM 13. } \text{Happens}(e, t) \wedge \text{Terminates}(e, f, t) \wedge t < t' \wedge \neg\text{Declipped}(t, f, t') \rightarrow \neg\text{HoldsAt}(f, t')$$

AXIOM 14. $Happens(e, s) \wedge t < s < t' \wedge (Initiates(e, f, s) \vee Releases(e, f, s)) \rightarrow Declipped(t, f, t')$.

In the exercise, the reader was asked to verify that, provided certain consistency assumptions are satisfied, axiom 13 could be added as an integrity constraint. We now have a more elegant option open to us: reformulate axiom 13 as

AXIOM 15. $Happens(e, t) \wedge Terminates(e, f, t) \wedge t < t' \wedge \neg Declipped(t, f, t') \rightarrow \overline{HoldsAt}(f, t')$

and leave axiom 14 as it is. In the presence of the new axiom, a scenario cannot both contain $Initiates(e, f, t)$ and $Terminates(e, f, t)$.

2.0.1. ****Parametrized fluents.** In Chapter 4 we freely used fluents with one or more free variables, for instance when discussing the mechanism of continuous change. As used there, a parametrized fluent $f(x)$ is a function which for each value of x yields a specific fluent (i.e. an object in the model). Accordingly, we used expressions of the form $HoldsAt(f(x), t)$. However, for certain applications of the formalism having to do with coercion (see Chapter 11) this formalization is not quite adequate, because the parametrized fluents must be objects as well, rather than functions. Suppose a parametrized fluent derives from a formula $\varphi(x, t)$ by nominalization, resulting in the *object* $\varphi[\hat{x}, \hat{t}]$. It is then clear how to substitute particular values for x : use $T_2(\varphi[\hat{x}, \hat{t}], x', t')$. We identified $HoldsAt$ with T_1 , and we may extend $HoldsAt$ as well to T_2 , thus allowing an expression of the form $HoldsAt(\varphi[\hat{x}, \hat{t}], x', t')$. Furthermore, we need not restrict parametrized fluents to those derived by nominalization. In the model, we may have a special sort (consisting of objects!) of fluents parametrized by x , a sort of fluents parametrized by x, y, z , etc. In terms of this notation, a dynamics is now given by a formula of the form

$$HoldsAt(f_2, x, t) \rightarrow Trajectory(f_1, t, f_2, x', d),$$

where the variable x' , formerly contained in f_2 now takes its own argument place in $Trajectory$.**

Part 3

A marriage made in heaven – linguistics and robotics

The semantics of tense and aspect is profoundly shaped by concerns with goals, actions and consequences . . . temporality in the narrow sense of the term is merely one facet of this system among many.

Mark Steedman [109, p. 932]

The discussion of the linguistic phenomena in Chapter 3 and of the cognitive psychology of time in Chapter 1 have both adduced evidence that supports Steedman's contention which serves as our motto. In the coming chapters we provide a strictly formal and computational theory of meaning which will show *how* 'goals, actions and consequences' shape tense and aspect. The theory takes as its starting point the lexical/phrasal level. Each VP comes with a default scenario in the sense of the event calculus (cf. definition 8). This default scenario determines the *Aktionsart* of the VP. It is important to note the qualification '*default* scenario': we do not believe that there is a fixed association between VPs and *Aktionsarten*; rather, temporal and aspectual operators (but also additional lexical material) may change *Aktionsart*, the phenomenon known as coercion. In general a scenario will mention a goal, and actions or activities which are instrumental in achieving that goal. Thus, in the scenario for 'build a house' there will be a representation for a goal-event (the finishing of the construction) and for a goal-fluent which results from the goal-event (the house having been built). The goal-event occurs in the scenario as an event type, not as an event token. Whether the goal-event happens or not is thus left open, and this is where tense and aspect have their various parts to play. This procedure will give an easy solution to the imperfective paradox, to mention but one example.

The theory of tense and aspect proposed here has necessitated a new representational format for meaning. This format can be seen as a computational version of Frege's distinction between sense and reference, with an added twist. The sense of an expression, say 'build a house', determines an algorithm which, upon given additional information such as tense, computes the minimal model in which that expression can be evaluated, given the context. The reference of the expression can then be read off from the model. The added twist is that here the reference is also conceived of cognitively, not realistically: understanding a piece of discourse involves constructing a cognitive model of that discourse. This point of view is of course familiar from DRT, and there exists good evidence that something like this must be going on in cognition, as we have seen in chapter 5. The most important difference with DRT is the latter's notion of truth: a DRS is in general a partial structure which is true if it can be embedded in the real world. Because our models are minimal, so that missing information has been filled in, in general they cannot be embedded in the real world. Moreover, gradually extending a discourse does not lead to an inclusion chain of models; there may occur many revisions along the way. Partial truth has thus been replaced by nonmonotonic truth.

CHAPTER 7

Aktionsart

Ever since Vendler's famous classification of verbs with respect to their temporal schemata, linguists have distinguished four or five lexical-aspectual classes or, as we prefer to say, *Aktionsarten*. These *Aktionsarten* comprise states, activities, accomplishments, achievements and points¹, which are exemplified in (1) to (5).

- (1) *States*: know, love, be beautiful, be on time².
- (2) *Activities*: run, push a cart, draw.
- (3) *Accomplishments*: cross the street, build a house, write a letter.
- (4) *Achievements*: begin, reach, arrive.
- (5) *Points*: flash, spot, blink.

There exist some linguistic tests designed to distinguish among *Aktion-sarten*. For example, adverbial modification with *for two hours* is possible with activities ('John ran for two hours') but in general not with accomplishments ('*John built a house for two years') or achievements ('*John reached the top for two days'). Temporal modification with the adverbial *in two hours* shows the opposite pattern. We will not pay a great deal of attention to these tests, the reason being that they are not watertight due to the possibility of coercion. For example, the verb 'flash' normally refers to a pointlike event. However, in the following sentence, the point has been coerced into an activity:

- (6) The light flashed for half an hour.

Aktionsart is thus not a permanent feature attached to a V or a VP. It is therefore best to give a language-free characterization of *Aktionsart*, and then to explain how a default association between a linguistic expression and an *Aktionsart* may arise. The required characterization can be given in the event calculus, by suitably composing events and fluents into a new structure, called *eventuality*.

¹This category was introduced by Moens and Steedman, see [109].

²We do not make a distinction between states which are extended in time and those which are not, i.e. point-states. This follows naturally from the decision to represent states by certain types of fluents; see Section 2.

1. Eventualities

Informally speaking, verbs (or VPs) refer to events, where an event is a way of conceptualizing a certain portion of space-time. We have seen in chapter 2, especially Section 1, that to a large extent, human conceptualization of events is driven by goals. In this Section we propose a formalization of this notion in the event calculus. However, since the term ‘event’ already has a formal meaning in the event calculus, events in the informal sense will henceforth be called *eventualities*, following Bach’s usage in [6].

DEFINITION 19. *An eventuality is a structure (f_1, f_2, e, f_3) , where*

- (1) f_1 is a fluent which represents an activity, something which exerts a force
- (2) f_2 is a parametrized fluent, representing a parametrized object or state, which is driven by the force f_1
- (3) e is the culminating event, representing a canonical goal
- (4) f_3 is a fluent which represents the state of having achieved the goal

Accordingly, one may associate to each VP a quadruple, each element of which is of the form ‘-’ (indicating that this slot may remain empty), ‘e’ (third argument only) or ‘f’ (first, second and fourth argument)³.

This is almost like the ‘event nucleus’ of Moens and Steedman (cf. [109, p. 903]): f_1 corresponds to their ‘preparation’, f_3 to the ‘consequent’, and e to the ‘event’ proper. There are some differences however. The first difference is that we have added the ‘parametrized object/state’, which plays a prominent role in the treatment of accomplishments and of the progressive. The second difference is that the language of the event calculus allows one to express important relationships between the elements of the quadruple. Suppose that a VP is represented by a structure of the form (f_1, f_2, e, f_3) , i.e. all slots filled. The informal explanation of definition 19 can be captured formally by means of a scenario in the sense of definition 8. For instance, the fact that e is the culminating event of the activity is rendered formally by the property *Terminates* (e, f_1, t) . The three fluents in the quadruple play different roles in a scenario. f_1 is not allowed to occur in *Releases* and in the third argument of *Trajectory*, but may occur in the latter’s first argument. This is the formal correlate of the intuitive notion that an activity may function as a cause. For f_2 and f_3 it is the other way around. Since we conceive of f_2 and f_3 as being driven by a cause, not themselves causes,

³For simplicity we have treated here only the case where the direct *object* is partial and changing. A similar phenomenon can occur in subject position however, as in the well-known causative/inchoative alternation

- (i) a. The cook thinned the gravy.
b. The gravy thinned.

Passives will be treated below, in chapter refCOE.

they are not allowed as the first argument in *Trajectory*. Usually, the consequent state f_3 will be a particular instantiation of the parametrized fluent f_2 , therefore it shares the latter's syntactic restrictions. The relationship between the fluents f_1 and f_2 is given by the dynamics, i.e. a statement of the form $S(f_1, f_2, t, d) \rightarrow \text{Trajectory}(f_1, t, f_2, d)$, where $S(f_1, f_2, t, d)$ is a state in the sense of definition 7. We have not added a slot for an event initiating f_1 , since there may be many of them, for example if the activity f_1 is interrupted from time to time. It is of course implicit that the activity f_1 is nontrivial, so that the scenario must contain an implicit reference to an initiating event. In a large part, this will be accomplished by tense; see chapter 8.

Note that an eventuality *determines* the scenario in the sense that these should at least fix the meaning of the positive components of the quadruple. Thus, for a quadruple of the form $(f_1, f_2, -, -)$ nothing needs to be said about a culminating event or goal, but this changes when the remaining slots are filled, for example when an activity (*draw*) is coerced into an accomplishment (*draw a circle*). Often the scenario will need to contain more, for example a reference to an initiating event. As will be seen in the following Sections, different *Aktionsarten* correspond to the various ways in which the slots in the quadruple can be filled.

2. Formal definition of *Aktionsarten*

We will define *Aktionsarten* as specific types of eventualities, not as verb classes, because we believe, as mentioned in the introduction, that verbs can by and large be coerced into any *Aktionsart*. As we have seen an eventuality can contain an event and up to three fluents; one may then distinguish types of eventualities according to the slots filled in the quadruple. To do so, we first have to distinguish formally between the roles of the fluents occurring in the quadruple. This is done in the following table, where the behaviour of the fluents with respect to the predicates *Releases* and *Trajectory* is specified⁴. In the table, + means 'is allowed to occur', and - 'is not allowed to occur', and Traj_i refers to the i^{th} argument of *Trajectory*.

	<i>Releases</i>	Traj_1	Traj_3
f_2, f_3	+	-	+
f_1	-	+	-

One consequence of this definition together with the metatheorems about the event calculus, is that activities will be represented by fluents that are extended in time, whereas states can be pointlike, as they should be: the sentence

(7) It is eight o'clock.

⁴There are no restrictions on occurrences in the other primitive predicates.

involves a state⁵. Section 4 below will explain how to introduce such states in the framework.

The traditional inventory of *Aktionsarten* can now be tabulated as follows (where – indicates that a slot is empty, + that it is filled with an object of the appropriate category).

State	(–, –, –, +)
Activity (strict)	(+, –, –, –)
Activity (wide)	(+, +, –, –)
Accomplishment	(+, +, +, +)
Achievement	(–, –, +, +)
Point	(–, –, +, –)

We will add a few comments on this table.

- (1) The distinction between states and activities is thus given by the different syntactic roles they play in a scenario. The distinction is relative to a given scenario; the same fluent may have changed its role in a different scenario.
- (2) More generally, one should think of the association between VP and quadruple as a default connection only. The quadruple is shorthand for the stored lexical entry for the VP. This lexical entry, as retrieved from declarative memory, is only the starting point of a computation, and in the course of the computation working memory may contain a very different *Aktionsart* for the VP. This is what happens in the case of coercion, to be studied in Chapter 11.
- (3) The difference between an activity in the strict sense (cf. *push*) and in the wide sense (*push a cart*), is that in case of the latter what Dowty [32] calls the ‘incremental theme’ has been added: not the cart itself, of course, but the changing position of the cart.
- (4) The quadruples occurring in the above list can all be described by VPs. It is an interesting question whether some of the other quadruples also characterize VPs. Observe that there are quadruples which characterize eventualities corresponding to *sentences*. Here are a few examples. A quadruple like (–, +, –, +) is exemplified by the sentence

- (8) For a while she remembered him dimly, but she soon forgot him completely.

In the same vein, the quadruple (–, +, +, +) corresponds to

- (9) She felt some affection for him for a while, but after that incident, her love was dead.

⁵Pace Comrie [17, p. 50].

In these examples we have used quasi-Boolean operations such as ‘but’; the question is whether there are simple VPs which need to be characterized by such quadruples.

We now aim to give a number of scenarios for concrete lexical expressions. To do so we need a preliminary step, namely turning natural language expressions into fluents and event types, building on the material in Chapter 6. The reader who has skipped Chapter 6 must at this point take our word for it that there is a canonical way to associate an event type and a fluent to a given verb, and may proceed directly to Section 2.1. We first make some general remarks on the roles of time and event variables. A more elaborate discussion of these issues can be found below, in the chapter on nominalization; here we include only what is strictly necessary for the formal development.

It has become customary, following Davidson [23] to provide the formal correlate of verbs with an event argument. Later work also assumes that the set of events is provided with some relations such as temporal inclusion and precedence. For our purposes this is not a good strategy, for one thing because we want to model the process of deriving various event-descriptions from VPs, i.e. nominalization; this cannot be easily done by starting from an abstract event and modifying it so that it becomes either a perfect or an imperfect nominal.

Nevertheless, the proposed model needs a starting point which gives a verb a richer structure than just a predicate with slots for subject and (in)direct object. This is because we would like to explain why the possibilities for nominalization are so severely restricted. Merely stipulating that, say, $run(x)$ can be associated to both an event type and a fluent, does not explain much. If however we allow a time parameter in the verb, the procedures available in the Feferman calculus strongly suggest that there exist only two ways of abstracting over the time parameter, corresponding to perfect and imperfect nominals respectively.

Thus, the verb run has the structure $run(x, t)$, where x denotes the subject position and t the time parameter. Of particular importance are then the *event type*, given formally by $\lceil \exists t. run(x, t) \rceil$, and the *fluent* $run[x, \hat{t}]$. Note the difference between the two abstractions: for concrete x , the latter is a function of time (to truth values), while the former is an object.

It should be emphasized that the formal representation $run(x, t)$ has no direct relation to tense. It is an auxiliary construct whose meaning does not correspond to a natural language expression. As will be seen below, tense is expressed using the *Happens* and *HoldsAt* predicates of the event calculus, so applies only after the construction of event type and fluent from a representation such as $run(x, t)$. The problem, often discussed, whether linguistic time can be modelled by the reals, does not arise here, because it is not claimed that we have direct access to the time parameter. We do have direct access to event types and fluents, and the main representation

theorem shows that *their* temporal profile is very well-behaved: they can be represented by definable finite unions of intervals and points.

We are now ready to provide sample scenarios corresponding to different *Aktionsarten*.

2.1. Example of an accomplishment: ‘build a house’. An accomplishment is characterized by a quadruple $(+, +, +, +)$, so that the scenario must contain a reference to all these components.

With this in mind, we need the following terms in the language of the event calculus.

- (1) *build* is an activity fluent, derived by (imperfect) nominalization from the corresponding verb
- (2) *house*(x) is a parametrised fluent representing the construction stage x of a house
- (3) c is a real constant indicating a construction stage at which the house is considered finished; thus *house*(c) is the fluent representing the consequent state.
- (4) similarly, a is a real constant indicating the stage at which the building starts⁶.
- (5) *start* is any event initiating building.
- (6) *finish* is the canonical event terminating building, namely when the house is finished.
- (7) g is a monotone increasing real-valued function relating the building activity to the construction stage.

Note that so far we have not introduced an object *house*; all we have is a fluent *house*(c). However, nothing prevents us from stipulating that *House*(*house*(c)), thus effectively turning *house*(c) into an object as well.

One possibility for a scenario is then given by the following set of statements.

- (1) *Initially*(*house*(a))
- (2) *Initiates*(*start*, *build*, t)
- (3) *Initiates*(*finish*, *house*(c), t)
- (4) *Terminates*(*finish*, *build*, t)
- (5) *HoldsAt*(*build*, t) \wedge *HoldsAt*(*house*(c), t) \rightarrow *Happens*(*finish*, t)
- (6) *Releases*(*start*, *house*(x), t)
- (7) *HoldsAt*(*house*(x), t) \rightarrow *Trajectory*(*build*, t , *house*($x + g(d)$), d)

The last two statements, 6 and 7, are jointly called the *dynamics*; this is characteristic of both activities (in the wide sense) and accomplishments. Accomplishments are distinguished from activities (in the wide sense) by statements describing the behaviour of the canonical goal, here statements 3 – 5. As remarked above in Section 3.1, the formulation chosen in 7 presupposes the validity of

⁶For example, the foundations of the house may have been laid already.

$$\text{HoldsAt}(\text{house}(x), t) \wedge \text{HoldsAt}(\text{house}(y), t) \rightarrow x = y,$$

a property that can be enforced by integrity constraints as introduced in chapter 8. Here we will simply assume that this can be done.

The scenario should be thought as being part of the lexical entry of ‘build a house’. The full entry will be much more complex, since it must add details concerning the building process. In fact, talking about ‘*the full entry*’ is apt to be misleading, since it suggests uniqueness. What we mean is something more modest: the lexical information concerning the expression ‘build a house’ stored in someone’s brain at a particular moment. Thus variation from person to person, and from moment to moment is allowed to a certain extend.

2.2. Example of an achievement: ‘reach the top’. Here we need a terminating event type⁷ *reach*, derived by (perfect) nominalization from the corresponding verb, and a fluent *be-at-the-top*, related by

$$(1) \text{Initiates}(\text{reach}, \text{be-at-the-top}, t).$$

More may be said about the resulting state, but we reserve this for our discussion of states below.

2.3. Example of an activity: ‘push a cart’. This is what we called an activity in the wide sense, characterized by a quadruple $(+, +, -, -)$: a force is exerted (‘push’) and as a result an object changes position. Accordingly, the terms we need are the activity (in the narrow sense) *push*, derived by (imperfect) nominalization from the corresponding verb, a parametrized fluent *position*(x), and an injective real valued function g . In contrast to accomplishments, there is no canonical goal here, so the main component of the scenario is the dynamics given by

$$(1) \text{HoldsAt}(\text{position}(x), t) \rightarrow \text{Trajectory}(\text{push}, t, \text{position}(x + g(d)), d).$$

2.4. Examples of states: ‘know’, ‘love’, ‘be sad’. At first the distinction between state and activity seems obvious: an activity involves change, and a state doesn’t. This characterization fits the perceived difference between ‘run’ and ‘know’. In our setup, states and activities must both be represented by fluents, and the question is how to account formally for the difference. The formula we found for this difference is that ‘states are causally inert’, so that they cannot occur as first argument of the *Trajectory* predicate. The reader may well wonder whether this is really what is at issue: don’t we say things like: ‘His excitement caused him to write the paper overnight’? Or ‘Loving her caused him endless sadness’?

One does indeed say such things, but we submit that ‘cause’ is not used here in the sense formalized in the event calculus. Rather, it functions as a

⁷‘Terminating’ with respect to an activity not explicitly mentioned.

precondition, in this case having the effect of increased sensitivity, so that we get something like

$$\text{HoldsAt}(\text{love}, t) \rightarrow \text{Initiates}(e, \text{sadness}, t)$$

where e is some action on the part of the female character (there may be many such actions, and corresponding statements in the scenario). Furthermore it must be noted again that we do not propose a fixed association between VPs and *Aktionsarten*. A state is primarily a cognitive category. Whether a state or an activity is associated to ‘love’ in the last resort depends on context. In fact we shall see below that there are contexts where ‘love’ is forced to be an activity.

A second point to be noted about states is that they usually are not that static after all. Sadness usually subsides without further aggravating events, and in democratic countries with a presidential system, one is president for an amount of time which is fixed beforehand. We must therefore investigate how this ‘decay’ of states (either continuous or discontinuous) can be modelled in the event calculus. One way to model continuous decay is to introduce the special fluent *decay*, which is syntactically an activity, and a monotone increasing function g , governing the decay rate. If we now conceive of ‘be sad’ as a parametrized fluent $\text{sad}(x)$, we may write the following formula for the dynamics of decay

$$\text{HoldsAt}(\text{sad}(x), t) \rightarrow \text{Trajectory}(\text{decay}, t, \text{sad}(x - g(d)), d).$$

The fluent *decay* will be initiated as soon as *sad* is initiated, and may be terminated by events increasing sadness. We thus do not agree entirely with Comrie’s characterization of states, when he writes

With a state, unless something happens to change that state, then the state will continue: this applies equally to standing and to knowing. With a dynamic situation [i.e. activity], on the other hand, the situation will only continue if it is continually subject to a new input of energy: this applies equally to running and to emitting a pure tone⁸, since if John stops putting any effort into running, he will come to a stop, and if the oscilloscope is cut off from its source of power it will no longer emit sound. To remain in a state requires no effort, whereas to remain in a dynamic situation requires effort, whether from the inside (in which case we have an agentive interpretation, e.g. *John is running*), or from the outside (in which case we have a nonagentive interpretation, e.g. the oscilloscope is emitting a pure tone) [17, p. 49].

On this characterization, ‘be excited’ and ‘be sad’ would not be states but activities, because their propensity to decay means that they require input of

⁸This refers to an example considered problematic by Comrie

- (i) The oscilloscope is emitting a pure tone at 300 cycles per second.

The use of the progressive indicates an ongoing activity, or continuous change, but the layman who does not know about sinus waves may well believe a tone is a static phenomenon.

effort or energy to remain steady at a given level. One may argue that there is an evident difference in scale here: running comes to an abrupt halt when there is no input of energy, whereas excitement is slower to trail off. This is more or less true. ‘More or less’ because it embodies an idealization: when stopping to run, physical inertia takes over for a short while, thus making the situation analogous to that of decaying excitement. Thus, a more detailed *EC* model of ‘run’ might specify that an event that terminates running releases $distance(x)$ and initiates a period of continuous change governed by inertia. The real difference between state and activity seems to lie in the perspective taken. If we view an eventuality as an activity, the input of effort or energy is profiled; if we view an eventuality as a state, the result of that input is profiled. This distinction is captured in the event calculus by assigning different syntactic roles to the corresponding fluents. It is however a bit of an idealization to identify a state with a single fluent, completely omitting the input. We will see below, in chapter 11, that it does not take much to shift the emphasis away from a state and toward the input.

States which are inherently bounded, and thus decay in a discontinuous fashion, such as ‘be president’, require a separate treatment. The technique introduced here will also be helpful for the semantics of the temporal adverbials ‘in’ and ‘for’, given in Section 4 of chapter 11. Let f be a fluent standing for ‘be president’. We introduce a parametrized fluent $time_f(x)$, and an activity fluent $clock$, with the following meaning. When f is initiated, i.e. when the president is inaugurated, the $clock_f$ starts ticking, and drives the fluent $time_f(x)$ by means of a linear dynamics. If the president serves for a term of, say, 4 years, the fluents $time_f(4yrs)$ and $clock_f$ will jointly trigger the event *finish*. This leads to the following scenario.

- (1) $Initially(time_f(0))$
- (2) $Releases(inauguration, time_f(0), t)$
- (3) $Initiates(inauguration, f, t)$
- (4) $Initiates(inauguration, clock_f, t)$
- (5) $HoldsAt(time_f(x), t) \rightarrow Trajectory(clock_f, t, time_f(x + d), d)$
- (6) $HoldsAt(time_f(4yrs), t) \wedge HoldsAt(clock_f, t) \rightarrow$
 $Happens(finish, t)$
- (7) $Terminates(finish, f, t)$
- (8) $Terminates(finish, clock_f, t)$

Clearly these conditions only enforce a term of 4 years in a minimal model. A coup could overrule condition 6, and impeachment may lead to a drastic shortening of the term by adding another terminating event to the scenario.

2.5. Intermediate cases of *Aktionsart*? Do the five *Aktionsarten* studied above really comprise all the possibilities? That is, can any VP be assigned unambiguously to one of the five classes? Comrie [17, p. 47-8] doubts this for English, citing the verbs ‘die’ and ‘persuade’ as examples. Vendler considers ‘die’ to be an achievement, i.e. an event followed by a

steady state, but what seems to speak against this is that one can unproblematically use the progressive, as in example (10-a) below. This suggests that ‘die’ is an accomplishment, but Comrie argues against this possibility on the grounds that (10-b) seems odd:

- (10) a. John was dying.
 b. ?John was dying, but the discovery of a new medicine led to his recovery.

Our informants differ on this issue, some finding (10-b) unexceptionable. Here, it is of interest to remark that in Russian, the sentence corresponding to (10-b) is not at all problematic. According to Comrie [17, p. 47-8], (10-b) is odd because

... the process preceding the event [i.e. death] is so intimately bound up with the event that once the process is under way the event cannot be prevented from occurring.

He proposes to consider such a combination of process and culminating event as a separate *Aktionsart*. A less controversial example is ‘persuade’:

- (11) a. John was persuading me to join him.
 b. *John was persuading me to join him, but I didn’t yield.

In this case, process and goal are inseparable.

Comrie’s intuition appears to us to be correct, although we would adduce a slightly different argument. Prototypical accomplishments such as ‘cross the street’ can be combined felicitously with manner adverbs such as ‘studiously’, ‘carefully’, ‘deliberately’ or ‘intentionally’. For ‘die’ this is out of the question, since one cannot oneself interfere with this process; in other words, the culminating event is not a goal which can be achieved by a plan to which a manner adverb can be applied. In this sense his remark about the inseparability of process and culminating event quoted above is entirely correct. However, in our setup it is not necessary to introduce a separate *Aktionsart* to cover such cases; rather, one may view it as a special case of accomplishment, obtained by putting an additional condition on the scenario. One way to do this is to introduce sortal distinctions among event types, by distinguishing between natural events, actions of self and actions of others. The scenario must then satisfy the requirement that no actions of self can terminate the process (although they may initiate it). The second example, ‘persuade’ can be analyzed similarly. For example, one may fix the completion of *Terminates* a priori by requiring that the culminating event is the *only* event terminating the process. One can think of many such boundary conditions on the scenario, and in this sense the *Aktionsarten* form a continuum, organized around five prototypes, rather than a discrete set.

3. Perfective and imperfective eventualities

Comrie usefully defines aspect as ‘[the] different ways of viewing the internal temporal constituency of a situation’ [17, p. 3]. The main opposition here is between perfective and imperfective aspect:

... perfectivity indicates the view of the situation as a single whole, without distinction of the various separate phases that make up that situation; while the imperfective pays essential attention to the internal structure of the situation [17, p. 16].

This distinction is useful also for languages, such as English, where it is not grammaticalized, in particular with regard to tense. If pure tense is defined, following Comrie [18, p. 1], as ‘the grammaticalization of location in time’, it seems as if there are two possibilities for such localization: either an eventuality is located in time as a whole, without regard for its constituent parts (perfective use of tense), or the eventuality (possibly extended in time) is anchored to a point on the time line (imperfective use of tense). It is easy to handle the latter case in the event calculus by means of the *HoldsAt* predicate, but the former case is not so easy. As a preliminary to a formal treatment of tense, we provide a formal way of viewing a complex eventuality, such as an accomplishment, as a single whole. This is done by means of a trick called *hierarchical planning*, an instance of the general notion of hierarchical organization of events as discussed in chapter 2.

If one wants to program, say an office robot, to pick up empty soda cans in offices, it is not very efficient to plan immediately at the level of executable instructions to the motors, because this increases the search space immensely. It is much more efficient to distinguish levels of detail, each with its own manageable search space. For instance, at a coarse level one might have the actions ‘go from room R_i to room R_j ’ (for adjacent rooms R_i, R_j) and write a plan in terms of these actions only. Suppose a plan has been constructed, featuring the action ‘go from room R_1 to room R_2 ’. We now look at a finer level of detail to write a plan for this action; here we need to include information about the door connecting R_1 and R_2 etc. The important point to note is that such information would be superfluous, indeed harmful, at the coarser level of granularity where the order of rooms to be visited is planned. This distinction in levels seems to embody an eminently reasonable cognitive principle, promoting efficiency of processing, and we propose that tense taps into the coarser levels of this hierarchy.

We must now show how an eventuality can be viewed as a single event. Let us first do the most complex case, the accomplishment, say, ‘write a letter’. We have been at pains to dissociate the goal, the finished letter, from the activity leading up toward the goal, writing; this dissociation is in fact the main ingredient of our solution to the imperfective paradox. The task at hand is to fuse these separate components into a single event. The general scheme to achieve this goes as follows. Let f be a fluent representing an activity, $start_f$ a starting event, and $finish_f$ the canonical terminating event,

representing achievement of the goal. Define (in the sense of definition 9) a new event type e by

DEFINITION 20. $Happens(start_f, s) \wedge Happens(finish_f, r) \wedge s < t \leq r \wedge HoldsAt(f, t) \rightarrow Happens(e, t)$.

The new event type can be seen as the perfect nominal of the accomplishment, that is, ‘the writing of the letter’⁹. The definition of its temporal profile uses key ingredients of the scenario for the accomplishment: the activity and the canonical goal. After completion, this definition has the effect of making $Happens(e, t)$ false for all t if the $finish_f$ and/or $start_f$ events does not occur. If they do occur, the temporal extension of e will be set equal to the period between $start_f$ and $finish_f$. For simplicity, we shall henceforth assume that the *start*–event occurs, thus avoiding triviality. Whether or not the *finish*–event occurs is often nontrivial, so this assumption will be mentioned explicitly whenever necessary.

One may think of statement 20 as an addition to the scenario for the accomplishment; in this case one has descriptions of the eventuality at different levels of granularity within the same scenario. If the language is restricted to the predicates of the event calculus plus the constant e , the fine structure is no longer visible. This construction should make clear that viewing an eventuality under perfective aspect is not the same as taking it to be punctual, in the sense that the temporal extent of the event is a point on the real time line, an issue to which we return below. Here we should note, however, that using definition 20 has as a consequence that e may have some structure after all. If the writing consists of several phases, interrupted by periods of rest, the temporal profile of e will be a set of intervals. If we also want to abstract from these interruptions, we end up with the condition

DEFINITION 21. $Happens(start_f, s) \wedge Happens(finish_f, r) \wedge s \leq t \leq r \rightarrow Happens(e, t)$.

This definition only uses the canonical goal from the scenario.

The analysis for an activity, represented by a fluent f , proceeds more or less along parallel lines, the only difference being that there is no canonical terminating event $finish_f$. However, if one replaces the canonical terminating event $finish$ by a non-canonical terminating event $stop_f$, the same analysis can be pushed through. As an example, consider the activity ‘run’. The fluent corresponding to ‘run’ may consist of a great many intervals; for example, each interval may correspond to a morning run. However, one cannot use (12-a) to mean (12-b)

- (12) a. *John ran, and he still does.
 b. John was running, and he still does.

⁹We will discuss the relation between nominalization and *Aktionsart* in chapter 12 on nominalization below.

even though he is running at this very moment. This shows that the past tense of the activity refers to a particular episode that has come to an end. The construction just given has precisely the effect of isolating a particular episode from a possibly larger collection of episodes described by the fluent.

Lastly, we note that in some cases a state may also be transformed into an event, namely when an initiating and a terminating event are given, as in ‘be president for 4 years’; compare the analysis of this example in Section 2.4. The construction then proceeds as for activities. Some states, such as ‘be beautiful’, do not lend themselves easily to such a perfectivization, hence for these states tense will be defined on the fluent, not on the derived event.

We now return to the issue whether and in what sense an event, viewed perfectly, can be considered to be punctual. We have already pointed out that such events need not be points *on the real line*. If we apply either the Kamp/Russell or the Walker construction from Chapter 1 to the events and fluents of a model of the event calculus, we can make good sense of the intuition that events viewed perfectly are in a sense punctual. To this end, define a binary precedence relation P on the set of event types and fluents mentioned in a given scenario, by the conditions

- (1) $P(e, e')$ iff $\forall t \forall t' (Happens(e, t) \wedge Happens(e', t') \rightarrow t < t')$
- (2) $P(e, f)$ iff $\forall t \forall t' (Happens(e, t) \wedge HoldsAt(f, t') \rightarrow t < t')$
- (3) $P(f, e)$ iff $\forall t \forall t' (HoldsAt(f, t) \wedge Happens(e', t') \rightarrow t < t')$
- (4) $P(f, f')$ iff $\forall t \forall t' (HoldsAt(f, t) \wedge HoldsAt(f', t') \rightarrow t < t')$

If we now apply the Kamp/Russell construction to the event structure defined by P we get a time line consisting of finitely many points where typically an event occupies a single point, and a fluent several points. This is because events will seldom overlap, whereas it is often the case that a fluent overlaps with several events. Since we have not yet computed actual models, we must ask the reader to take this statement on faith; the computations in chapters 9 and 10 will bear out our claim. Furthermore, as we pointed out in Chapter 1, the Walker construction allows us to add some detail to this picture, by giving the possibility of representing an event as an open interval not containing any instants (so that the event is in a sense punctual), while retaining the possibility to refer to its endpoints. In conclusion, we may note that it makes sense to consider an event, viewed perfectly, as ‘punctual’, as long as one resists the temptation to turn ‘punctual’ into an absolute notion.

CHAPTER 8

Tense

In this chapter we will consider the tenses of English, a language in which we find tense in a relatively pure form, without a prominent contribution from aspect. The present chapter will however include a discussion of the future tense in English, although it is controversial whether this is a proper tense at all. Be that as it may, the future tense is ideally suited to illustrate the roles of goals and plans in the tense system. The formalization that we propose again uses the event calculus and logic programming, but also introduces a new concept, *integrity constraint* (borrowed from database theory), which allows us to take account of the anaphoric nature of tense and Reichenbach's reference point *R*.

Comrie defines tense as 'grammaticalized location in time', a definition that provides us with three notions to discuss. In the last chapter we concluded that it is often *event types* that get located, that is, eventualities viewed perfectly. That is, we showed how to associate an event type to activities and accomplishments, and claimed that it is these event types that are important for tense. Achievements and points come with a distinguished event type, and we assume for the moment that it is this event type that is located by tense; although some twists will be added to this idea later. Locating a state in time is an instance of the imperfective use of tense. It does not involve the preliminary construction of an event type: in this case tense acts directly on the fluent.

In chapter 1 we discussed several representations of time, and following Walker we indicated how continuous time may arise as an idealization of event structure. This gives us two basic options for formalizing 'location in time'. We may locate an event in the given event structure, or we may locate the event in the corresponding representation by means of instants, so, idealizing a bit, in the reals. In principle, the former option can be made to work, since the tense system uses only relations such as 'precedes', 'simultaneous with' and 'abut'¹, which are present (or can be defined) in the Walker setup. However, aspect definitely needs some sort of continuous structure like the reals, as we saw in the last chapter, and so the combination of tense and aspect would require us to consider the event structure and the representing instant structure simultaneously. We believe that this is how things are organized in cognition, and that the relative paucity of the primitive relations used for tense is evidence for this. It is however a

¹This relation is necessary for the French Passé Antérieur.

nightmare from the expository point of view to have to work both with an event structure and its representation as a continuum, and so we choose the second option: tense is concerned with the location of event types in the reals. If she so wishes, the reader may verify that everything we say can be reformulated in terms of the first option.

Lastly, we have to discuss the meaning of ‘grammaticalized’. Comrie explains this in terms of two default conditions: (a) the expression of tense is obligatory, and (b) the expression of tense is morphologically bound. These are default conditions in the following sense: the prototypical instances of grammaticalized tense satisfy both (a) and (b), cases satisfying either (a) or (b) but not both are more marginal, and languages satisfying neither (a) nor (b) must be said to express temporal location lexically instead of grammatically. An interesting variant of this definition is given by Stassen [107, ch. 9]

DEFINITION 22. [*Definition of the Tensedness Parameter*]

- a. If a language has a grammatical category of tense which
 - (1) is morphologically bound on verbs, and
 - (2) minimally involves a distinction between past and nonpast time reference, then that language is tensed.
- b. In all other cases, a language is non-tensed.

That is, languages exhibiting only a future/nonfuture split are excluded, on the ground that the so-called future tense in these languages is often a form of irrealis. Comrie [18, p. 46] disagrees, citing Hua as a counterexample (a language not discussed in Stassen [107]), but it seems to be a fact that languages having a pure future tense, i.e. one defined in terms of ‘precedes’ only, are few and far between.

The question then arises, whether the semantics of the tenses of a given language, should depend on whether or not tense is grammaticalized in that language. In other words, the question is whether one may assume a semantics of tense which is uniform for both tensed and tenseless languages. We have no strong opinions on this matter, but note that the analysis given here only works if verbs can be assumed to have a temporal parameter at some level of representation.

In the remainder of the chapter we explain how the English tenses can be expressed in the formalism of the event calculus.

1. Reichenbach’s reference time R

Reichenbach’s great insight into tense was his realization of the importance of the reference time, on a par with event time and utterance time. The reference time R is a marker for the time, context or situation that we are talking about. R must be known by the participants in order for the temporal discourse to make sense. Reichenbach noticed that the reference time can be different from the event time, as for instance in the present perfect

- (1) I have caught a flu.

Here the infection-event lies in the past, but the reference time is identical to the utterance time: the sentence is meant to have present relevance, e.g. as an explanation for my being bad-tempered. We now have to investigate how the reference time is to be formulated in our framework. This is not at all easy, as the following example will make clear.

Suppose we try to model the present perfect, in a very simple situation, where there is an event type e (say a viral infection) which initiates a consequent state f e.g. having a flu); there are no further events or fluents. The scenario therefore contains only the statement

- (2)
- $Initiates(e, f, t)$
- .

Suppose the utterance time is denoted by a constant now , to be interpreted on the reals; this constant belongs to the constraint language. The present relevance of the present perfect then suggests that the contribution of this tense to the scenario is the addition of formula (3-b), so that the scenario becomes

- (3) a.
- $Initiates(e, f, t)$
-
- b.
- $HoldsAt(f, now)$
- .

We would like to derive from (3), using the axioms of the event calculus, that for some $t < now$, $Happens(e, t)$. Naively one might reason as follows: completing the axioms of the event calculus plus the scenario seems to give us

$$Happens(e, t) \wedge Initiates(e, f, t) \wedge t < t' \wedge \neg Clipped(t, f, t') \leftrightarrow HoldsAt(f, t'),$$

so that the desired result would follow after applying the given (3-b) to the right hand side. Although this argument embodies an important intuition, it cannot be pushed through as stated, since the completion of the scenario must take account of (3-b), so that we get

$$HoldsAt(f, t')$$



$$[Happens(e, t) \wedge Initiates(e, f, t) \wedge t < t' \wedge \neg Clipped(t, f, t')] \vee [now = t'],$$

from which nothing can be derived. Thus the contribution of the reference time cannot simply be an *addition* to the scenario.

To clarify the contribution of the reference time, we need a small excursion in database theory, taking an example from Kowalski [65, p. 232].

An *integrity constraint* in a database expresses obligations and prohibitions that the states of the database must satisfy if they fulfill a certain condition. A simple example is a database of family relationships, which

should satisfy constraints such as ‘everybody has a genetic father’ (obligation) and ‘no one is both father and mother’ (prohibition). The operational meaning of such integrity constraints is that each time the database is updated, it must be checked whether the constraints still hold.

Kowalski [65, p. 232] advocated the use of integrity constraints in the slightly different context of reactive agents, who have to perform appropriate actions given certain sensory input. Here the integrity constraint is used to request an update (e.g. stating that an action has been performed), rather than to regiment the updates as in the examples above. For instance, the ‘obligation’ to carry an umbrella when it is raining, may be formalized by the integrity constraint

$$(4) \quad \textit{HoldsAt}(\textit{rain}, t) \rightarrow \textit{HoldsAt}(\textit{carry-umbrella}, t + \epsilon).$$

The intended meaning is that if it rains at t , then the agent should carry an umbrella soon after.

The crucial point here is the meaning of \rightarrow in 4. The formula 4 cannot be an ordinary program clause, for in that case the addition of $\textit{HoldsAt}(\textit{rain}, t)$ would trigger the consequence $\textit{HoldsAt}(\textit{carry-umbrella}, t)$ which may well be false, and in any case does not express an obligation. Below we will therefore not use \rightarrow in an integrity constraint, but only the expression IF ... THEN, whose meaning is defined operationally.

A good way to think of an integrity constraint expressing an obligation is to view the consequent as a constraint that the database must satisfy if the antecedent holds. Similarly, an integrity constraint expressing a prohibition can be taken to mean: if the antecedent holds, then the database should not satisfy the consequent. In other words, in the first case the integrity constraint imposes an obligation on the given clauses in the database to establish that the consequent should hold. This entails in general that the database has to be updated with a true statement about the world; which statement that is, has to be found out by abduction. To return to our example, there will be an action *take-umbrella*, whose meaning is given by the database clause

$$\textit{Initiates}(\textit{take-umbrella}, \textit{carry-umbrella}, t).$$

Suppose the database is updated with $\textit{HoldsAt}(\textit{rain}, \textit{now})$, that is, with the antecedent of the integrity constraint 4. The integrity constraint then requires us to set up a derivation starting from the query

$$?\textit{HoldsAt}(\textit{carry-umbrella}, \textit{now} + \epsilon).$$

Applying the event calculus we can reduce this query to

$$?\textit{Happens}(\textit{take-umbrella}, \textit{now}), \neg \textit{Clipped}(\textit{now}, \textit{carry-umbrella}, \textit{now} + \epsilon).$$

We now have to update the database in such a way that the query succeeds. This can be achieved if we *only* add the clause

$$\textit{Happens}(\textit{take-umbrella}, \textit{now}),$$

and no other occurrences of events. For in this case the query

$?Happens(take-umbrella, now), \neg Clipped(now, carry-umbrella, now + \epsilon)$

reduces to

$? \neg Clipped(now, carry-umbrella, now + \epsilon),$

and this query can be shown to succeed by applying negation as failure to

$? Clipped(now, carry-umbrella, now + \epsilon).$

Indeed, the latter query fails because of the way we updated the database. Of course, it is assumed that a statement such as $Happens(take-umbrella, now)$ only gets added when in fact the action has been performed.

As this example makes clear, an integrity constraint requires us to update a database in a particular way. A derivation is started with the consequent of the integrity constraint as the top query. Then resolution with clauses from the database is applied for as long as possible. The derivation will in general end with a query that cannot be further resolved. If this were an ordinary derivation we would then apply negation as failure to the top query. In the case of an integrity constraint we use the unresolved bottom query instead to suggest an addition to the database which will make the top query succeed after all. The procedure chosen has the effect of making a minimal update of the database to ensure success of the top query: the computation exploits as much of the database as is possible, and only plugs in facts when absolutely necessary.

These considerations lead us to the following definition of integrity constraint as it applies in our context.

DEFINITION 23. *Let $R, R', R'' \dots$ be a finite set of constants each denoting a reference time; these constants belong to the constraint language. An integrity constraint is a statement of the form*

(\dagger) IF $? \varphi$ succeeds THEN $? \psi(R, R', R'' \dots)$ succeeds/fails, or

(\dagger') IF $? \varphi$ fails THEN $? \psi(R, R', R'' \dots)$ succeeds/fails,

where φ and ψ are formulas of the event calculus.

The operational meaning of (\dagger) is that if the scenario satisfies φ , the goal $? \psi(R, R', R'' \dots)$ must be made to succeed, or to fail finitely. To determine whether the scenario satisfies φ , one has to investigate whether the goal $? \varphi$ succeeds².

A typical application is where³ $\psi(t, t', t'' \dots) = HoldsAt(f, t), t < now$. In this case we require that the goal $? HoldsAt(f, R), R < now$ succeeds (or fails finitely) by either of the following strategies.

²It is also possible to have an integrity constraint without a condition, i.e. where φ is a tautology. An entry in my diary like ‘appointment in Utrecht, Friday at 9.00’ expresses an unconditional obligation to satisfy $HoldsAt(be-in-Utrecht, Friday\ at\ 9.00)$, and presented with this integrity constraint, my internal database comes up with a plan to satisfy the constraint.

³From now on, we will usually drop ‘ \wedge ’ in a query in favour of a simple comma.

(1) One may update the scenario with *Happens*, *Initiates* and \neg *Clipped* formulas (assumed to be true), using the axioms of the event calculus; this is the strategy of choice if f is an activity fluent. It has the effect of making the temporal denotation of f extended in time. Here, updating with ' \neg *Clipped* formulas' means that the update procedure should be such that the relevant *?Clipped* goals fail.

(2) If the first strategy fails, the scenario may also be updated with true *HoldsAt* or *Initially* formulas, or (in)equalities in R in the language of the reals. This is a possible strategy if f represents a state, since states do not have to be caused by events. For example, they may be a particular instance of a parametrized state, which evolves continuously driven by a dynamics.

Another typical application of integrity constraints in our context is where the goal which must succeed or fail finitely is of the form $?Happens(e, R)$, $R \leq now$. Again there are two possible strategies for handling the goal:

(1) if $Happens(e, t)$ occurs in the head of a clause with nontrivial body $\theta(e, t)$, proceed by resolving the query $? \theta(e, R)$;

(2) otherwise, replace $Happens(e, R)$ by a set of true (in)equalities in R .

A common feature of these two strategies is that a goal $?G$ is resolved as far as possible using the given program P . If the integrity constraint is that $?G$ must succeed, whereas it fails finitely from P , then one extends P to a program P' from which $?G$ can be made to succeed. The required extension can in principle be read off from the failing derivation tree, although other extensions are possible. Note that a goal $?G$ that fails finitely from P can be made to succeed by moving to an extension of P , but that a goal $?G$ that *succeeds* from P cannot be made to fail from an extension P' . To achieve the latter, we would have to move to a subprogram of P . In the applications that we envisage this will not be necessary.

We now illustrate the linguistic relevance of integrity constraints using the example of the perfect: if f is the resultant state, the integrity constraint for the present perfect is that the query

(4) $?HoldsAt(f, R)$, $R = now$,

must succeed, whereas for the pluperfect success is required for the query

(5) $?HoldsAt(f, R)$, $R < now$.

In both cases the logic programming mechanism starts a computation from the given query by applying the axioms of the event calculus. For instance, applying axiom 3 to (4) means that the database searches for an event e and a time t_0 such that $Initiates(e, f, t_0)$, $Happens(e, t_0)$ and $\neg Clipped(t_0, f, now)$. If this query does not succeed, the database may ask the outside world for input. In the above example of 'I have caught a flu', the scenario consists only of the formula $Initiates(e, f, t)$ only. The database then asks for input

of a true clause $Happens(e, t_0) \wedge t_0 < R$. This is the computational meaning of the perfect. (As we will see below, it is also part of the computational meaning of the progressive, where the fluent f represents an activity and e an initiating event.) If the database would be unable to find a formula $Initiates(e, f, t)$, it could also ask the world for input of a true formula of this type. Alternatively, it could forego the search for an f -triggering event and ask for a set of (in)equalities in R or (using axioms 1 and 2) a true *Initially* formula instead. These strategies may have to be applied if f represents a stative verb, for in that case f need not be triggered by an action or event.

The upshot of the preceding discussion is that a reference time is characterized by a set of fluents which must hold at that time. This stipulation captures the idea that the role of a reference time is to fix the situation or context that we are talking about. In general such situations are only partially determined. Suppose the reference time is characterized by fluents f_1, \dots, f_n , i.e. by the integrity constraint

$$?HoldsAt(f_1, R), \dots, HoldsAt(f_n, R).$$

If we want to stipulate that another fluent f , say the result fluent involved in the perfect, holds at R , the only way to do this is to enlarge the integrity constraint to

$$?HoldsAt(f_1, R), \dots, HoldsAt(f_n, R), HoldsAt(f, R),$$

i.e. f must occur in a subgoal in the integrity constraint. In general we will mention only the immediately relevant part of the integrity constraint, in this case $?HoldsAt(f, R)$, and leave out the contextually given part⁴.

Before we move to a discussion of the various tenses, let us take stock. The principal function of the scenario is to contribute lexical information, which is general and does not talk about specific times. The addition of temporal information is required to construct a sentence out of lexical material. Contrary to first impressions, the reference time cannot be added as a fact to a scenario, as we have seen in the case of the perfect. The reference time is an integrity constraint formulated in terms of *fluents*, which typically puts constraints upon possible temporal locations of *event types*, including the events constructed by hierarchical planning as in Section 3. We have to exercise some care here: the traditional phrase ‘event time’ obscures the fact that there are at least two different kinds of events, what we have termed here fluents and event types. Localizing stative verbs or VPs in the progressive, involves anchoring a fluent, whereas the simple tenses locate event types. This is related to an important issue: the role of *Aktionsart* in defining tense. Some authors, such as Comrie [18], prefer to define tense abstracting from aspectual features, the idea being that tense

⁴In the presence of integrity constraints, the model whose existence is posited in theorem 4 of Chapter 5 depends of course on the constraints used to make the goal in the integrity constraint succeed or fail.

talks only about localization in time; an example will be given below. We try to go a different route, and allow for the possibility that tense works out differently for different *Aktionsarten*.

2. Event time and the sentence

Reichenbach knew only one single event time, but since the event calculus operates with both fluents and event types, we have to distinguish two ‘event times’ as well. But the event calculus explicitly embodies this distinction in the form of the two predicates *Happens* and *HoldsAt*. Reichenbach’s E is therefore either of the form $E = \{t \mid \text{Happens}(e, t)\}$ or of the form $E = \{t \mid \text{HoldsAt}(f, t)\}$. The integrity constraints above thus express that the reference time falls inside an event time.

Armed with these definitions we may now give a preliminary account of the contribution of tense in the construction of a sentence. Very roughly speaking, the semantics we propose is of the dynamic variety, in which the meaning of a sentence is its context-change potential. Consider a very simple example:

(6) John ran.

obtained by applying the past tense to ‘John run’. It will be seen that in this case the past tense operates on an event type e representing an episode of running, constructed as in Section 3. Semantically, the past tense requires that the event time coincides with the reference time, and that both are situated before *now*. A first formalization⁵ of (6) could then be given by the integrity constraint

(7) The query $?Happens(e, R), R < now$ must succeed,

where R may also occur in some other integrity constraint functioning as background. Formally, if $?Happens(e, R), R < now$ succeeds, R must be contained in the temporal extension of e . This is as it should be, but what does it mean to represent a sentence as an integrity constraint and not, as one might naively think, as a formula?

The integrity constraint is our version of the insight of dynamic semantics that a sentence generates an update of the context in which it is interpreted. As has been explained above, the requirement that a query such $?Happens(e, R), R < now$ succeeds, entails in general that the scenario to which the constraint is added must be updated in order to ensure success. Together, the scenario and the axioms of the event calculus strictly regiment the update process, so that always a minimal extension is chosen.

⁵We choose this formulation for didactic purposes only. The actual formalization is slightly more involved.

DEFINITION 24. *The notions ‘true’ and ‘false’ can be applied to tensed sentences in the following manner. The tensed sentence is true if the integrity constraint can be satisfied; it is false if there is no way to satisfy the integrity constraint*⁶.

Since sentences are modelled as integrity constraints, the entailment relation between sentences φ_k, φ_l translates into an entailment relation between integrity constraints IC_k, IC_l for instance of the form ‘given a scenario S , if IC_k succeeds, then IC_l succeeds’, or ‘given a scenario S , if IC_k fails, then IC_l succeeds’ – there are four cases in all. Take the first example. We are given a scenario S , and we assume that IC_k can be made to succeed. This leads to a minimal extension S' of S in which IC_k in fact succeeds. We then require that IC_l also succeeds in S' . The other cases have a similar meaning. For example, in Section 2 we will study the inference from (8-a) to (8-b)

- (8) a. John is crossing the street.
b. John will have crossed the street.

The first sentence is characterized by the integrity constraint that the reference time equals *now* and is contained in the activity *crossing*, while for the second sentence the reference time is in the future and is situated in the state resulting from the completion of the crossing. Dowty posited that the inference from (8-a) to (8-b) is valid with respect to ‘inertia worlds’ (see the discussion in Section 2 below); here, validity is with respect to scenarios and their minimal models. In particular, given the scenario S for ‘cross the street’, sentence (8-a) introduces the integrity constraint that $?HoldsAt(crossing, now)$ must succeed. This has the effect of extending S with a statement $Happens(start, t_0)$ for some $t_0 < now$. In the resulting extended scenario S' , the query $?Happens(reach, t), t > now$ (representing sentence (8-b)) leads to a successful computation.

3. Present tense

In the following it will be useful to distinguish between the syntactic and the semantic present tense. By definition, in semantic present tense the utterance time, the event time and the reference time all coincide. The use of the syntactic present tense is however not confined to cases where this meaning is appropriate. There is for instance the future use of the present tense in *Tomorrow I fly to London*, which will be treated below in Section 5. Furthermore, accomplishments and activities in the syntactic present tense generally have habitual meaning, as in *I sing on Sundays*. It is doubtful whether tense is used deictically here, referring to *now* (for a contrary view see Comrie [18, [p. 39]]). It rather seems to be anaphoric to a time *period*, as in

⁶**Some computations may loop, and therefore tensed sentences may in principle be neither true nor false. In practice this does not present a problem.**

- (9) This year I sing on Sundays, but last year I sang on Wednesdays.

The characteristic feature of habituals appears to be that they explicitly refer to a plan to be executed repeatedly during a given period. Some planned executions of the plan may fail to take place, but that does not make (9) false. There also appears to be a subtle difference in the ways plans are involved in habituals and in the semantics for accomplishments. An expression such as ‘write a letter’ intends to refer to the real process of writing a letter, even though that process is coded semantically by means of a plan, i.e. the steps that have to be executed. In contrast to this, the habitual refers to the plan itself, not so much to the real-world events constituting the execution of the plan. In principle it is possible to represent this distinction in our framework, since we have enough machinery available to code a plan (i.e. a scenario plus the axioms of the event calculus) into a single object, but we shall not do so here since it is somewhat involved.

In English, use of the syntactic present tense is rather restricted. The syntactic present tense cannot be used with accomplishments and activities to express that utterance time, reference time and event time coincide, even though this semantic content can be expressed by means of the present progressive, as we will see below. When achievements and points are used in the syntactic present tense, the resulting construction can have a meaning superficially similar to that of the semantic present tense. This happens for instance if the achievement is a performative verb such as ‘name’ or ‘promise’:

- (10) a. I name this ship the ‘Titanic’.
b. I promise to give you ten pounds before tomorrow 12 noon.

because the act of promising clearly coincides with the utterance of the sentence. But especially in the case of (10-b) this is not exactly the semantic present tense, because the reference time, which corresponds to ‘give you ten pounds’, is situated in the future, and (hopefully) before tomorrow 12 noon. In fact this is almost a paradigm case of the formalization of reference time as an integrity constraint, since we are clearly concerned with an obligation here.

Another case in which the present tense is used, is in a simultaneous report of events that are currently going on, as in a live broadcast of an Arsenal match

- (11) Henry reaches Bergkamp. Bergkamp takes the ball, shifts it to his other foot, shoots⁷ ... goal!

Again this is a very special use of the present tense: obviously utterance time and event time coincide, but in this case reference time does not even have a meaning. Apart from these cases, use of the semantic present tense

⁷In this context, the first two verbs are achievements, the other two points.

with achievements and points is not felicitous (Croft [21, p. 70]). Consider the following sentences

- (12) a. Bill *reaches/just reached the top.
b. Bill *shatters/just shattered the windowpane.

An analysis of this phenomenon in our framework would go like this. An achievement is characterized by a quadruple $(-, -, e, f)$. At the moment e happens, nothing is achieved yet, since f starts to hold after e has occurred. The utterance time S must therefore be located inside the consequent state, although it may be ‘infinitesimally close’ to the event time. This explains why the past tense but not the present tense can be felicitously applied to achievements. In fact one may also argue that the past tense has the meaning of a present perfect here: the event time lies in the past, but the reference time coincides with the utterance time.

All in all, this leaves us with only states to account for. Here are some examples

- (13) a. It is 11.34am.
b. The weather is beautiful.
c. I am glad you made it.

Let us see how integrity constraints can be useful here. In example (13-a), we need the parametrized time-fluent $time(x)$ as introduced in Section 2.4. The integrity constraint is given by the query $?HoldsAt(time(11.34am), R)$, $R = now$, which can be solved successfully by adding the equality $R = 11.34am$, if indeed it is now 11.34am. For example (13-b), let f be the fluent corresponding to beautiful weather. The corresponding integrity constraint now requires that the query $?HoldsAt(f, R)$, $R = now$ must succeed. If the scenario contains no further information about f , the query can only be solved by checking it against reality. If sentence (13-b) occurs in a context, as for instance in

- (14) Now that a thunderstorm has cleared the sky, the weather is beautiful.

attempts to solve the query may use material from the scenario, as in example (13-c). This sentence refers to the state of me being glad represented by the fluent f , which is caused by your having arrived safely and/or in time e . Sentence (13-c) implies that there is a causal connection between f and e , which can be formalized as

- (15) $Initiates(e, f, t)$

Here the integrity constraint requires that the query $?HoldsAt(f, R)$, $R = now$ must succeed, which, via axiom 3 of the event calculus translates into the requirement that $?Happens(e, t)$, $t < now$, $\neg Clipped(t, f, now)$ must succeed. In particular we get that e must have happened sometime in the past.

It is worthwhile to emphasize the last point. The reader may have wondered why we chose the roundabout route via integrity constraints to formalize the present tense, instead of simply adding a clause $HoldsAt(f, now)$. The reason is that adding the clause does not activate axiom 3, and so one cannot conclude that e must have happened already. The situation is entirely analogous to that of the present perfect, discussed informally in Section 1, and the reader who is having trouble here, is advised to refer back to that Section.

The preceding considerations can be formalized in the following definition.

DEFINITION 25. *Let a VP be given, whose default Aktionsart is a state, represented by a fluent f . The present tense of the VP is then defined by the integrity constraint $? HoldsAt(f, R)$, $R = now$. We will often write this constraint in the simpler form $? HoldsAt(f, now)$.*

3.1. Present progressive. The preceding discussion covers only the simple present, not the present progressive. It is important to note here that the progressive applies unproblematically only to activities and accomplishments. We claim that this is the case because the progressive requires for its application the presence of a dynamics in the sense of definition 8 of Chapter 4. That is, the progressive can be applied only if there are an activity fluent f_1 and a parametrized fluent f_2 related by a condition of the form

$$HoldsAt(f_2(x), t) \rightarrow Trajectory(f_1, t, f_2(x'), d).$$

Such a pair of fluents exists when we are concerned with an activity or an accomplishment. If the progressive is applied to a VP belonging to different *Aktionsart*, it has the effect of extending the given scenario with a dynamics, effectively coercing that VP to either activity or accomplishment. This case will be treated in Chapter 11.

The purely temporal feature of the present progressive is then captured by

DEFINITION 26. *Suppose a given VP is either an activity or an accomplishment; let f be the associated activity fluent in the corresponding quadruple. Then the present progressive of the VP is the integrity constraint $? HoldsAt(f, R)$, $R = now$.*

The reason why the formulation in definition 26 is preferred over the addition of a fact $HoldsAt(f, now)$ is the same as indicated for the case of states, and even more urgent here. If $HoldsAt(f, now)$ were added as a fact, f could be true at the point *now* only. This contradicts the intuitive notion of an activity. If we add the integrity constraint that $? HoldsAt(f, now)$ must succeed, the computation leads us back, via axiom 3, to the event which started f , and in this way f becomes extended in time.



FIGURE 1. Past tense

4. Past tense

The function of the English past tense at first sight appears to be to locate an event in its entirety prior to the speech point, and simultaneous (or at least overlapping) with the reference point, as in figure 1.

Before we attempt to formalize this idea in terms of integrity constraints, it should be noted that Comrie [18, p. 41] favours an interpretation of the past tense slightly different from the one given here.

... the past tense simply locates the situation prior to the present moment, and says nothing about whether the past situation occupies just a single point prior to the present moment, or an extended period prior to the present moment, or indeed the whole time up to the present moment ... It should also be noted that use of the past tense only locates the situation in the past, without saying anything about whether that situation continues to the present or into the future, although there is often a conversational implicature that it does not continue to or beyond the present.

The examples that he gives

- (16) a. John used to live in London.
b. John was eating his lunch (when I looked into his office).

are special in that they refer to an habitual (or state) and a progressive respectively. Other *Aktionsarten* seem to behave differently. Indeed, as we noted before, one cannot use (17-a) to mean (17-b):

- (17) a. *John ran, and he still does.
b. John was running, and he still does.

and hence activities behave differently from states in this respect. Comrie wants to abstract from aspectual features when defining tense. As noted, our strategy is different, and aims to define tense for each *Aktionsart* separately.

Even if it is agreed that the past tense requires the entire event to lie in the past, figure 1 by itself does not, however, adequately capture the meaning of the past tense. One problem is that the English past tense cannot be used in isolation, at least not without sufficient common ground between speaker and hearer to establish the identity of *R*. Consider Comrie's examples [18, p. 41]:

- (18) a. John was in Paris.
b. John has been in Paris.

Sentence (18-a)

implies the existence of a specific occasion on which John was in Paris, the ability to refer to which is shared by speaker and hearer'; sentence (18-b) 'simply indicates that there is some time in the past, not necessarily further identifiable by speaker or hearer, at which the proposition *John be in Paris* held [18, p. 41].

Steedman [109, p. 906] makes the same point when he writes

... the past tense, unlike the perfect, demands that the past reference point be explicitly established, either by a modifier, such as a *when* clause, or by the preceding discourse. Thus (19-a) below, is inappropriate as the first utterance of the discourse, except that the reader *accommodates* a temporal referent, in Lewis' [72] sense of the term – that is, *introduces* an appropriate individual in the database, as one often must at the beginning of a modern novel. But (19-b) is appropriate, on the assumption that the hearer can identify the time in the *when* clause:

- (19) a. *Chapman breathed a sigh of relief.
 b. When Nixon was elected, Chapman breathed a sigh of relief.

These observations give further support to the idea that tenses must be defined by means of an integrity constraint, instead of a program clause. Indeed, they suggest that reference time is more fundamental than event time – that the reference time must be fixed first by an integrity constraint, to provide an anchor for the event time.

For simplicity, we first consider Comrie's definition, which requires only that the event time is at least partially in the past.

DEFINITION 27. *Let a VP be given. Let e be the event associated to an activity or accomplishment by the procedure of Section 3, or the canonical event implied in an achievement or a point. The past tense of the VP is given by the integrity constraint*

$?HoldsAt(f_1, R), \dots, HoldsAt(f_n, R), R < now, Happens(e, R) \text{ succeeds.}$

The effect of the integrity constraint is that R and the event time will have a common instantiation which is situated in the past. It is possible that R and the event time extend beyond *now*⁸. If we want R to be completely situated in the past, we have to reformulate the integrity constraint governing R as the demand that

$?HoldsAt(f_1, R), \dots, HoldsAt(f_n, R), R \geq now$ fails.

For the past tense we then get

⁸It is an interesting problem to formalize the conversational implicature that, since the present is more important than the past, the event does not continue to or beyond the present unless this is explicitly said so. Unfortunately we do not have a good formalization in our framework of conversational implicature, so we must let the matter rest.

DEFINITION 28. (a) For VP either an activity, an accomplishment, an achievement or a point, the past tense of the VP locates the event type associated to that VP (as constructed in Section 3) in time.

(b) Let e be the event associated to an activity or accomplishment by the procedure of Section 3, or the canonical event implied in an achievement or a point. The past tense of the VP is given by an integrity constraint, as follows: if the query $?Happens(e, t)$ succeeds, then the query

(*) $?HoldsAt(f_1, R), \dots, HoldsAt(f_n, R), R \geq now, Happens(e, R)$ fails.

Note that the query $?Happens(e, t), t \geq now$ will fail finitely if a derivation starting from $?Happens(e, t)$ ends in constraints which are incompatible with $t \geq now$. This will occur for instance if the query $?Happens(e, t)$ fails, that is, as has been explained in Section 3, if the terminating event implied in the definition of e has not happened. This case has been excluded by adding the demand that the goal $?Happens(e, t)$ succeeds as the antecedent of the integrity constraint. The full query (*) then can fail only if it ends in a constraint $c(R)$ incompatible with $R \geq now$, i.e. satisfying $\forall t(c(t) \rightarrow t < now)$. If the query (*) does not fail, given the scenario, it means that material must be added. In either case, the constraint $c(R)$ provides the explicit reference time demanded by the past tense.

In the previous discussion we assumed, as is customarily done, that reference time and event time must coincide. More complicated arrangements are possible: for instance the situation where the reference time is at least partially in the past, and the event time fully in the past. For R one then takes definition 27, and the event time is further determined by the integrity constraint that $?Happens(e, t), t \geq now$ must fail. The reader can easily do the other combinations herself.

It remains for us to discuss the past tense of states. There seem to be two typical uses for the past tense here, as exemplified by (20-a) and (20-b)

- (20) a. I lived on Bruntsfield Avenue in 1999.
b. In two minutes, I was fast asleep.

Sentence (20-a) picks out a particular ‘point’ (i.e. 1999) inside an episode, and nothing is implied about the period before or after that point. By contrast, sentence (20-b) is an example of the ingressive use of the past tense: it focusses attention on the first moment at which the state holds. We consider (20-a) to be the default meaning of the past tense applied to states, and we will show later, in chapters 9 and 11, how the ingressive meaning can be derived from this by a process of coercion.

DEFINITION 29. (a) If a state is bounded, as in ‘be president for four years’, the past tense applies to the event constructed from the state (cf. Section 3), and is defined as in 28.

(b) If the state is not explicitly bounded, the past tense applies to the corresponding fluent f , and is modelled by the integrity constraint which says

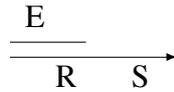


FIGURE 2. Past progressive

that the query

$$?HoldsAt(f, R), R < now$$

must succeed.

Comrie's examples of cases where the use of the past tense does not imply that the event has terminated were states or past progressives. We agree with his judgments here, so the integrity constraint demands only that part of f is in the past, allowing that some part of f lies in the future.

4.1. Past progressive. The purely temporal meaning of the past progressive is illustrated by figure 2.

The reference point is now inside the eventuality. Since the eventuality is represented by an activity fluent f , the reference point must be found by means of an integrity constraint involving the *HoldsAt* predicate, as in definition 29.

DEFINITION 30. *The past progressive applies to the activity component of a quadruple, say the fluent f , and is represented by the integrity constraint which says that the query*

$$?HoldsAt(f, R), R < now$$

must succeed.

It is now easy to see that past tense and past progressive differ in their implications when applied to an accomplishment:

- (21) a. John built a house.
b. John was building a house.

Sentence (21-a) entails that a house has been built, whereas no such inference can be drawn from (21-b). This difference can be reproduced formally, since by construction the past tense applies to an event type which entails the completion of the building. By contrast, the past progressive applies to an activity fluent, and does not support this entailment.

4.2. When clauses. We can test the above ideas for the case of *when* clauses. Consider the following set of examples (all from Google), where in the annotation (\dots, \dots) , the first component refers to the *Aktionsart* of the subordinate clause, and the second component refers to the *Aktionsart* of the main clause.

- (22) a. When John entered the program, he did not have any kids. (achievement, state)

- b. Robby and Mike were jamming, when John entered the room. (achievement, activity)
- c. When John Dewey entered his sophomore year, the family moved to #178 South Prospect Street. (achievement, achievement)
- d. Dawn was just breaking over the horizon when John entered the Black Water Swamp. (achievement, accomplishment)
- e. When his wife was ill in 1868, he walked to Lyttleton and back to get medicine for her. ((bounded) state, activity)
- f. Elvin Jones: "When John [Coltrane] was ill, he wasn't there physically but his spirit certainly was." (state, state)

Traditionally it is assumed that the *when* clause introduces the reference time in Reichenbach's sense which then serves as a temporal anchor for the event time represented by the main clause. In the first four examples, the reference time is clearly tied to an event. Since we analyzed bounded states as events derived by hierarchical planning, one could also take the reference time to be an event here; alternatively, it could be a fluent, in particular without the adverbial 'in 1868'. We will choose the latter interpretation.

In (22-a), (22-b), (22-d) and (22-e) the reference time introduced by the *when* clause strictly contains the event time of the main clause. In (22-c) however, Bill is absent *after* the entering of John; this holds as well in Steedman's example (19-b) quoted above. This pattern we will now try to explain.

Given what we have said about reference time, it is clear that a *when* clause introduces an integrity constraint, of the form $?Happens(e, R)$ or $?HoldsAt(f, R)$, as the case may be. The examples in (22) will be represented as pairs (e, e') , (e, f) , (f, e') or (f, f') , where e (f) represents the event (fluent) introduced by the *when* clause and e' or f' the *event* or *fluent* expressed by the main clause, depending on *Aktionsart*, tense and grammatical aspect of the main clause. By the definition of past tense, we get *fluents* as second component of the pair for (22-a), (22-b) and (22-d) and an *event* for (22-c) and (22-e).

The definition of the past tense then furthermore implies that a sentence of the form 'When e, e' ', 'When e, f' ', 'When f, e' ' or 'When f, f' ' must be translated as an integrity constraint

$$?Happens(e, R), Happens(e', R), R < now \text{ succeeds,}$$

or

$$?Happens(e, R), HoldsAt(f', R), R < now \text{ succeeds,}$$

or

$$HoldsAt(f, R), Happens(e', R), R < now \text{ succeeds,}$$

or

$$HoldsAt(f, R), HoldsAt(f', R), R < now \text{ succeeds,}$$

respectively. In the second case, we automatically get that the event of the subordinate clause takes place inside the event of the main clause. In the

first case, applied to example (22-c), Bill is indeed absent after the entering of John, if the scenario contains the condition *Initiates(Bill-leave, Bill-absent, t)*. However, in those cases where *when* relates two events, other temporal orderings between the two events involved seem possible, for example in

(23) Bill had already arrived when John came in.

The formal correlate of this example will be given in Section 1.2 of chapter 10. In some cases it is implied, though not said explicitly as in (23), that the temporal location of the event mentioned in the main clause is not preceded by that of the subordinate clause, as in

(24) When I won my only game against Bobby Fisher, I used the Ruy Lopez opening. (Steedman [109, p. 928])

If e is the event of winning, and e' is the event of using the Ruy Lopez opening, then the relation between e and e' is governed by the integrity constraint

$$?Happens(e', t'), Happens(e, t), t' \geq t \text{ fails.}$$

5. Future tense

It is a moot point whether English has a grammaticalized pure future tense, pure in the sense of being concerned only with location in time. The constructions involving the auxiliaries *will* and *be going to* seem to express something more, and the future uses of the present tense and present progressive ipso facto do not represent grammaticalized location in time. Nevertheless, the present framework allows much to be said about the meaning of the 'future' constructions in English, so we will not be unduly bothered by the question whether these represent a true future tense⁹.

We begin with a long list of examples, which furnish constraints on the formal interpretation.

- (25) a. The train leaves at 5.27 pm *(but will leave at least 20 min. later).
 b. The train will leave at 5.27 pm *(but will leave at least 20 min. later).
 c. The train is scheduled to leave at 5.27 pm (but will leave at least 20 min. later).
- (26) (Google) Tomorrow I am sleeping over at Samantha's house.
- (27) a. The sun rises at 5.50 a.m.
 b. ?The sun will rise at 5.50 a.m.
- (28) a. *It rains tomorrow.
 b. It will rain tomorrow.
- (29) a. John flies to Chicago tomorrow.
 b. John is flying to Chicago tomorrow.

⁹This Section owes much to conversations with Phil Schogt and Darrin Hindsill.

- c. John will fly to Chicago tomorrow.
- (30) a. *I go to Chicago unless my boss forbids me¹⁰.
 b. (Google) I am going unless some unknown demand stops me.
 c. (Google) I will go unless there is severe or dangerous weather.
- (31) a. *I fly to Chicago if my boss asks me.
 b. ?*I am going if you go¹¹.
 c. I am going if my health allows me/if I am able.
 d. (Google) Barak said to Deborah, "I will go if you go with me. I will not go if you don't go with me."
 e. (Google¹²) The young man thought for a moment and then he said "I will go - if you will go with me".
- (32) a. I will fly to Chicago tomorrow.
 b. ě I am going to fly to Chicago tomorrow.
 c. I was going to fly to Chicago tomorrow, but my boss forbade me.
- (33) a. Bill will throw himself off the cliff.
 b. Bill is going to throw himself off the cliff.
- (34) a. Pieper was going to be Chief Executive Officer of Philips in 2 years time.
 b. (Google) Tony Blair in 1997: 'I am going to be a lot more radical in government than people think'.
- (35) (Comrie) If/When you go out/*will go out in the rain, you will get wet.
- (36) (Comrie)
 a. If you are going to do the shopping, I'll give you money.
 b. If you do the shopping, I'll give you money.
 c. If you'll do the shopping, I'll give you money.
- (37) (Comrie)
 a. *If it'll rain, you should take your umbrella.
 b. If it's going to rain, you should take your umbrella.
- (38) a. #Harry moves to Philadelphia.
 b. Next Tuesday, Harry moves to Philadelphia.

One can distinguish two main dimensions along which future events can be classified. The first dimension concerns two possible perspectives on future events in so far as they can be affected by humans: as events per se, and as goals, to be achieved by a plan (which may possibly fail). In very rough outline one may say that the use of the present tense emphasizes the first perspective. A good example of this is (25-a), where (usually against one's

¹⁰Habituals such as 'I drink the water wherever I go – unless the natives come running at me, screaming "DON'T DRINK THE WATER OR YOU'LL DIE"' (Google) are of course perfectly fine.

¹¹That is, Google did not find any instances of 'I am going if you go', or indeed any instance 'I am going if (event phrase)'. The examples found were all of the kind illustrated in (31-c).

¹²There are very few such examples in Google. Cf. also sentence (35).

better judgement) one considers only the event of the train's departure, abstracting from delays. Other examples are (27-a) and (29-a). Sentence (26) furnishes an example in the same vein, except that the 'event' is actually more like a state¹³.

Examples (30-a) and (31-a) show that the present tense is no longer allowed if even a mild form of conditional planning is introduced. By contrast, sentences (30-c) and (31-d) show that the auxiliary *will* is fine with planning. Indeed, the auxiliaries often indicate that some amount of planning is involved, but here an orthogonal dimension comes into play. Suppose we view a future event from the perspective of goals and plans; the other case will be treated later. If *will* is used in contexts such as (32-a), it is indicated that no actions of self interfere with the execution of the plan. On the other hand, if *be going to V* is used in that same context ((32-b) and (32-c))¹⁴, the possibility of an obstacle arising is deliberately left open. Thus sentence (33) is false if Bill in the end does not jump off the cliff, unlike sentence (33-b): as Comrie [17, p. 64-5] remarks, the second sentence can be shouted as a warning and an injunction to do something to prevent Bill from jumping, whereas the first sentence cannot be used in this way. Sentence (34-a) is a true statement which expresses an intention that was once actual, but which was never realized¹⁴. The sentences in (32) make a similar point: the conjunction of an intention together with the denial of its execution has to be expressed by *be going to*, as in (32-c), not by *will*. The use of *will* indicates the presence of a plan, which may have all sorts of preconditions; but it does not allow the possibility of the agent's interference with the preparation. Sentence (30-c)¹⁴ shows that such plans may have preconditions,

Now suppose we view a future event purely from the perspective of its occurring. Then it is no longer possible to talk about possible obstacles to the execution of a plan; but we may still classify events as to whether their occurrence is regarded as certain or doubtful. The present tense is indicated in the first case, the auxiliaries in the second. Sentence (27-a) is a good example of the first, with (27-b) being distinctly awkward¹⁵. By contrast, sentence (28-a) is excluded, and its meaning has to be expressed by (28-b).

We have not said much so far about the purely temporal features of the three ways to express future tense. Steedman's example (38) from [109, p. 908] shows that the use of the syntactic present tense for future tense needs the preliminary establishment of a temporal referent, just as we saw in the case of the past tense. We will therefore model this version of the future

¹³Google actually yielded very few such examples.

¹⁴By contrast, the sentence

- (i) Pieper would be CEO of Philips in 2 years time.

introduces a reference time R such that Pieper definitely is CEO at time $R + 2yrs$.

¹⁵Oddly enough, this is how (27-a) is expressed in meteorologists' jargon (Darrin Hindsill p.c.).

tense by means of an integrity constraint, similarly to what we did for the past tense. By contrast, if the future tense is expressed by means of auxiliaries it is not anaphoric in this sense, and a slightly different formalization is called for.

The examples involving the interaction of future tense with conditionals, taken from Comrie [18, p. 117-121], again show that it is almost impossible to separate the purely temporal features of the future tense from the planning features. The default temporal interpretation of a conditional is that (a) the subordinate clause sets up the temporal reference point, and (b) the event described by the main clause is located after this reference point. This explains the distribution seen in (35). Similarly, (36-b) expresses that the ‘you’ does the shopping first, and will be paid back later. It is possible to liberalize the order, as in (36-c), where, at least according to Comrie [18, p. 119]¹⁶, the relative timing of the events is not fixed. However, the possibilities for doing so are severely restricted, as (37-a) shows. This is a particularly complicated set of data, which will be analyzed in depth in Section 5.3.1. The preceding considerations now have to be cast into formal definitions.

5.1. Events without frills. Consider the future tense as applied to the occurrence of events *per se*, abstracting from the possibility to plan. We have seen that the syntactic present tense is the preferred mode of expression for this case. We have already seen that due to the anaphoric element in examples such as (38), a formalization involving integrity constraints is called for, just as we did for the simple past tense.

DEFINITION 31. (a) For all Aktionsarten except states, the future use of the syntactic present tense is represented by an integrity constraint of the form: if the query $?Happens(e, t)$ succeeds, then the query $?Happens(e, R)$, $R \leq now$ must fail. Here e is either derived from an activity or an accomplishment by hierarchical planning, or the canonical event of achievement or point.

(b) If the fluent f represents a state, the future use of the present tense is defined by the integrity constraint that the query $?HoldsAt(f, R)$, $R > now$ must succeed.

Turning now to the perspective from which events are viewed as the result of planning, we have three modes of expression, by means of *be going to*, *will* and the (present) futurate progressive.

5.2. Be going to VP. This periphrastic construction is of special interest to us, because its evident spatial origin ties in neatly with the spatial origin of the event calculus as a formalism for path planning. It has been claimed that *be going to VP* is a prime example of a fundamental metaphoric transformation $SPACE \longrightarrow TIME$, but the situation is much more subtle, as has been clearly explained by Bybee, Perkins and Pagliuca [12, p. 268].

¹⁶On this issue there is disagreement among our informants.

The temporal meaning that comes to dominate the semantics of ['be going to VP'] is already present as an inference from the spatial meaning. When one moves along a path toward a goal in space, one also moves in time. The major change that takes place is the loss of spatial meaning. Here... the function of expressing intention comes into play. When a speaker announces that s/he is going somewhere to do something, s/he is also announcing the intention to do that thing. Thus intention is part of the meaning from the beginning, and the only change necessary is the generalization to contexts in which an intention is expressed, but the subject is not moving spatially to fulfill that intention.

We similarly hypothesize that the mental planning apparatus was first used for path planning, and then extended to cover other domains as well. If there is indeed such a close connection between *be going to VP* and the event calculus, in particular the part involving the *Trajectory* predicate, then the following construction naturally suggests itself. In the event calculus, the meaning of 'go to' can be modelled by means of a scenario linking an activity fluent f_1 and a parametrized fluent $f_2(x)$, which represents the path followed driven by the action of f_1 . The main component of the scenario is thus a dynamics. Since the auxiliary is 'be going to', the definition of the progressive has to be applied, which means that the integrity constraint must apply to the fluent f_1 . Combining the auxiliary with a verb, as in the expression 'be going to fly to Chicago', we have to add an event type corresponding to 'fly to Chicago'. The event and the fluent are 'glued together' by means of a scenario: the event will be the culmination of the preparatory phase f_1 . The scenario provides the activity fluent with the meaning appropriate to the context, that is, it determines the precise activity involved. In the case of a stative verb, as in 'be going to be CEO', the possibility of an inchoative reading arises because the event calculus allows one to define an event marking the beginning of the state, thus reducing this case to the previous one. These considerations will now be turned into a formal definition.

DEFINITION 32. *The future use of be going to VP can apply either to a fluent f (if the default Aktionsart of VP is a state) or an event e (all other cases).*

First consider the case where VP is an event e (e.g. 'fly to Chicago'); then be going to VP requires the presence of a plan, comprising

- (1) *a fluent f_1 representing the preparatory activity*
- (2) *a parametrized fluent $f_2(x)$ representing the (continuously changing) result of that activity*
- (3) *a certain stage $f_2(c)$ of $f_2(x)$ which triggers e via a formula*

$$\text{HoldsAt}(f_1, t) \wedge \text{HoldsAt}(f_2(c), t) \rightarrow \text{Happens}(e, t),$$

(4) a dynamics linking the fluents $f_1, f_2(x)$ which is of the form

$$\text{HoldsAt}(f_2(x), t) \rightarrow \text{Trajectory}(f_1, t, f_2(x'), d).$$

(5) a reference point R given by an integrity constraint. If *be going to V* is used in the present tense, the reference point is now, so that the integrity constraint becomes

$$?\text{HoldsAt}(f_1, \text{now}) \text{ succeeds.}$$

If it is used in the past tense (i.e. in the form *was going to V*), the integrity constraint becomes

$$?\text{HoldsAt}(f_1, R), R < \text{now} \text{ succeeds.}$$

If *be going to* occurs in the antecedent of a conditional¹⁷ the reference time is determined by the antecedent plus contextual material¹⁸.

In order to apply *be going to VP* to a stative verb (e.g. ‘*be CEO*’) represented by a fluent f , we need an event e , such that f is consequent upon e , where e can be seen as the culmination point of a preparatory activity f_1 . We think of f as being of the form $f_2(c)$, such that f_1 and $f_2(x)$ are related by a dynamics as above. Since in this way an event e which marks the beginning of f , is introduced into the discourse, it is possible to have an inchoative reading of *be going to VP* as applied to states.

In both cases, the scenario to which the above statements are added should not imply that f_1 is terminated before e happens. This does not preclude the possibility that terminating events are introduced after these statements are added.

Definition 32 makes the meaning of *be going to VP* similar to that of the progressive, for which see Section 2. It has the effect of making e happen in minimal models of the scenario, which exclude unforeseen terminating events. *Be going to V* can be used as a future tense because if the reference time is set to *now*, the event e will happen in the future. Of course, the scenario may be expanded with events untimely terminating the preparatory activity f_1 , so that e never actually happens. This is the difference between *be going to* and *will*.

Armed with this definition, let us now reconsider some examples involving *be going to VP*.

- (39) a. Bill is going to throw himself off the cliff.
b. I was going to fly to Chicago, but my boss forbade me.

¹⁷As in the example from Google:

- (i) If it is going to rain that night, make sure that the plastic ground cloth under your tent does not hang out beyond the edges of the tent.

Clearly the conditional is to be modelled by an integrity constraint here.

¹⁸In this case the time of putting up the tent before the deictic ‘that night’.

- c. Tony Blair in 1997: 'I am going to be a lot more radical in government than people think'.
- d. Howard Dean is going to be president next year.

In line with Bybee et al.'s suggestion that the future meaning of 'be going to' arises from a process of semantic bleaching, we assume that the activity fluent f_1 can stand for motivation or intention. In (39-a) there is an intention in the present, formalized as a fluent f_1 satisfying the integrity constraint

$?HoldsAt(f_1, now)$ succeeds.

In (39-b), there was an intention in the past, formalized as a fluent f_1 satisfying the integrity constraint

$?HoldsAt(f_1, R), R < now$ succeeds.

In (39-a) and (39-b) the intention prepares, via a dynamics, for a culminating *event*: in (39-a) this is 'throw oneself off the cliff', and in (39-b) 'fly to Chicago'. In a minimal model of (39-a), no obstacles occur, and Bill will indeed throw himself off the cliff, but there may be other models in which someone comes to rescue him. In any model of (39-b), an obstacle occurs before *now*. This means that f_1 is terminated between R and *now*.

Sentences (39-c) and (39-d) involve the stative VPs 'be a lot more radical in government than people think' and 'be president next year'. In the latter case, there is a rather prominent inchoative reading, which arises formally because *be going to* is applied to an event type derived from the VP, not to the VP itself.

We add some comments on the notion of 'plan' as used in the preceding definition. The examples in our list all mostly refer to animate subjects engaged in bringing something about. Obviously the use of *be going to VP* is not confined to these cases. Here are some more examples (from Google):

- (40)
- a. It is important for the catcher to know what the ball is going to do when it hits the fence.
 - b. If the universe is going to contract, then 98% of the universe must not be visible.
 - c. Prof. Paul Davies: 'Over the last few decades, mostly the evidence has been that the Universe is going to go on expanding.'
 - d. I am going to be 50 years old this year and I have noticed that men pay much more attention to me, even young men.

In all four cases there is no plan in the sense of conscious formation of an intention or goal and the mental computation of steps toward that goal. Nevertheless, the same formal structure as the one given in definition 32 is present: e.g. in the first example the dynamics can be taken literally, in its physical sense. Examples (40-b) and (40-c) are subtly different: they appear to refer to a physical process, but they actually refer to the physicist's *model* of that process. Therefore terminating events are possible, and

this explains the use of ‘be going to’ instead of ‘will’. The fourth example similarly allows for an event terminating the dynamical development toward the speaker 50th anniversary. The first example is interesting in this respect. After the ball has been hit by the bat it follows a deterministic, easily computed trajectory; hence ‘when’ instead of ‘if’. (Although terminating events remain possible, as when in a Dutch soccer game the ball hit a seagull.) But after the expected collision with the fence, chaotic dynamics sets in, and what the ball is going to do is anybody’s guess. The upshot of this discussion is that underlying all uses of *be going to VP* is a common formal structure, which in the case of animate subjects forming intentions works out as a goal-plan structure.

5.3. Will VP. A sentence such as ‘I will fly to London tomorrow’ is *false* when at some instant after speech time, the ‘I’ terminates the preparations for flying, unlike ‘I am going to fly to London tomorrow’. The following (repeated) examples (41-a) and (41-b) show that *will* can however be used felicitously if a plan has preconditions whose satisfaction is as yet uncertain:

- (41) a. I will go unless there is severe or dangerous weather.
 b. Barak said to Deborah, "I will go if you go with me. I will not go if you don't go with me."
 c. What Tony Blair did not say in 1997: ‘I will be a lot more radical in government than people think’.

The auxiliary *will* shares with *be going to* the presence of an intention or a preparatory activity. It differs in that no *actions of self* terminating the preparation are envisaged. This leads to the following definition.

DEFINITION 33. *The semantic contribution of will in will VP is defined as that for be going to, except that the following integrity constraint schema is added: for all e representing actions of self except for the canonical culminating event terminating f_1 , if $?Happens(e, t)$ succeeds, then*

$$?Terminates(e, f_1, t) \text{ fails,}$$

where f_1 is the fluent representing the preparatory activity for VP as in definition 32. This conditional integrity constraint may be reformulated as: for all e except for the canonical culminating event terminating f_1 ,

$$?Happens(e, t), Terminates(e, f_1, t) \text{ fails.}$$

This is a schema, that is, one concrete integrity constraint for each concrete e (representing an action of self) that occurs in the scenario for the VP. Henceforth we omit the cumbersome phrase except for the canonical culminating event terminating f_1 , but it is always understood¹⁹.

¹⁹See also exercise 8, where the reader is asked to reformulate the integrity constraint in terms of the *Clipped* predicate.

Suppose one were to add a clause $Terminates(e, f_1, s_0)$ (for some constant s_0) to the scenario, then the query $?Terminates(e, f_1, t)$ would succeed, so that there is no way to satisfy the integrity constraint in 33. In this way untimely occurrences of terminating actions of self render a sentence involving *will* false (cf. definition 24), while preserving its plan-like contribution. This could explain why Blair was careful not to formulate (34-b) as (41-c). Obviously, *will* is also a future tense: since the execution of a plan takes time, and the reference time is *now*, the use of *will* has the effect of locating the event time in the future – or more generally, in the future of the reference time.

The above characterization of *will* focussed on its use in referring to actions of the speaker or others. There is of course also the impersonal use, as in

(42) ěIt will rain tomorrow.

Here it is mostly actions interfering with weather-dynamics that are of importance, but the integrity constraint has the same structure. Nevertheless there is a difference in truth conditions between (42) and (32-a), repeated here for convenience

(43) ěI will fly to Chicago tomorrow.

Because the integrity constraint refers only to actions of self cutting short the preparation phase, (43) can be true at utterance time if the flight is cancelled due to an Al Qaeda alert, say. By contrast, (42) is false if it does not rain tomorrow. This pure, non-modal, future tense arises because in this case there is no meaningful restriction on the events allowed in the integrity constraint.

As our examples show, the use of *will* in a main clause is felicitous if the plan has preconditions ('if ...') or hedges ('unless ...'). Our next task is to see how these fit in the framework; here it is useful to consider the interaction of conditionals with future tense more generally. The key to the analysis is the connection between the modal character of *will* and the form of the integrity constraint in definition 33.

5.3.1. *Future tense in subordinate clauses.* Let us see whether we can detect a regularity in examples (44-a) – (44-k):

- (44)
- a. (Comrie) If/When you go out in the rain, you will get wet.
 - b. (Comrie) *If you will go out in the rain, you will get wet.
 - c. (Comrie) *When you will go out in the rain, you will get wet.
 - d. (Comrie) If you'll do the shopping, I'll give you some money.
 - e. (Google) The young man thought for a moment and then he said "I will go - if you will go with me".
 - f. (Google) Jeremy Ord: "I think the biggest lesson is, when you are going to go out into that big wide world you must be prepared for the good stuff and the bad stuff."

- g. (adapted from Comrie) *If it'll rain, I'll give you my umbrella.
- h. (Google) You will also need a raincoat if it is going to rain.
- i. (Google) I sent a Valentine to some little girl I liked. It read, "If you're as sweet as you're good looking, I'll do the plowing if you'll do the cooking"
- j. (Google) I will go if I get a scholarship.
- k. (Google (from the Torah)) 'When you will go out to war against your enemies, and HaShem your God will give them into your hand, and you will capture his captives.' (Devorim 21.10)

In this list we seem to have four kinds of conditionals:

- (1) promise or obligation ((44-d), (44-e), (44-i), (44-k))
- (2) precondition for an action ((44-j); cf. also (41-a) above)
- (3) causal relation (44-a)
- (4) conditional obligation ((44-f),(44-h))

To set the stage, let us start with a discussion of two of the 'when'-sentences in the above list, namely (44-a) and (44-c). If we apply the recipe for the formalization of 'when'-sentences given in Section 4.2 to the future tense, a natural way to formalize the 'when'-version of (44-a) is as

$$?Happens(e, R), HoldsAt(f_1, R), R > now \text{ succeeds,}$$

together with: for all e' ,

$$?Happens(e', t), Terminates(e', f_1, t) \text{ fails,}$$

where e is the event type corresponding to 'go out in the rain', and f_1 is the preparatory fluent given by definition 33. This is because the subordinate 'when'-clause sets the reference time, which according the definition 33 sits inside the preparatory fluent. The second integrity constraint schema is necessary to highlight the contribution of *will* over and above *be going to*.

Pursuing the same line of thought, the formalization of (44-c) would be as an integrity constraint of the following form

$$?HoldsAt(f_1, R), HoldsAt(f'_1, R), R > now \text{ succeeds,}$$

where f_1 and f'_1 are the preparatory fluents given by the definition of *will*, together with two integrity constraint schemata: (1) for all e' ,

$$?Happens(e', t), Terminates(e', f_1, t) \text{ fails}$$

and (2) for all e'' ,

$$?Happens(e'', t), Terminates(e'', f'_1, t) \text{ fails.}$$

But (44-c) is not acceptable, so why doesn't the proposed analysis work for (44-c)? One reason must be that locating the reference point R inside the preparatory fluents gives us no clue as to the temporal ordering of the events that are being prepared, beyond that both must occur in the future. Indeed, it is consistent with the integrity constraint that the event e' for which f'_1 prepares occurs before the culminating event e for f_1 . But the purpose of sentence (44-c) is obviously to state some form of causal relationship which

also determines the temporal order of the events; since the semantics cannot capture this feature, sentence (44-c) is not felicitous. Comrie [18, p. 119] remarks that in cases such as sentence (44-f) this is not a problem; in fact, if the temporal ordering of the events described by the clauses is to be left indeterminate, the use of *be going to* is mandatory. The reader can easily verify for herself that the integrity constraint corresponding to the ‘when’ version of (44-a) determines the order completely: first e happens, and then the event for which f_1 prepares.

Moving on to ‘if’, as a first guess we may take the effect of ‘if p ’ to be an update of the context with p , as in dynamic semantics. This idea has a natural formulation in our framework, by using conditional integrity constraints. In fact it works to perfection in examples like (44-h), which is just a variant of the example that Kowalski used to introduce integrity constraints (see Section 1). Here the proper formalization would be the conditional integrity constraint²⁰

$$\begin{array}{c} \text{IF} \\ ?\text{HoldsAt}(f_1, t), t > \text{now} \text{ succeeds} \\ \text{THEN} \\ ?\text{HoldsAt}(f_1, R), \text{HoldsAt}(f'_1, R), R > \text{now} \text{ succeeds} \\ \& \\ \text{for all } e', ?\text{Happens}(e', t), \text{Terminates}(e', f'_1, t) \text{ fails} \end{array}$$

In this formula, f_1 represents the preparatory fluent for ‘be going to rain’, and f'_1 is the corresponding preparatory fluent for ‘will need a raincoat’. The second component in the consequent of the implication is required by definition 33. Informally, the combined expression says that if the knowledge base is updated with information about impending rain, one should start preparations for taking a rain coat, not allowing any action to interfere with these preparations. Sentence (44-f) can be analyzed similarly.

Next consider a causal relation, as in the ‘if’-version of (44-a). This sentence can be formalized as

$$\begin{array}{c} \text{IF} \\ ?\text{Happens}(e, t), t > \text{now} \text{ succeeds} \\ \text{THEN} \\ ?\text{Happens}(e, R), \text{HoldsAt}(f_1, R), R > \text{now} \text{ succeeds,} \\ \& \\ \text{for all } e', ?\text{Happens}(e', t), \text{Terminates}(e', f_1, t) \text{ fails,} \end{array}$$

where e is the event type corresponding to ‘go out in the rain’, and f_1 is the preparatory fluent given by definition 33 as applied to ‘will get wet’. Why

²⁰In the following, IF ... THEN has the operational meaning introduced in definition 23. To economize on brackets, we assume that the conjunction & binds stronger than IF ... THEN.

is it impossible to formulate the intended relation as (44-b)? Semantically, (44-b) would correspond to

IF
 $?HoldsAt(f_1, t), t > now$ succeeds
 &
 for all e , $?Happens(e, t), Terminates(e, f_1, t)$ fails,
 THEN
 $?HoldsAt(f_1, R), HoldsAt(f'_1, R), R > now$ succeeds,
 &
 for all e' , $?Happens(e', t), Terminates(e', f'_1, t)$ fails,

where f_1, f'_1 are the preparatory fluents representing ‘will go out in the rain’ and ‘will get wet’, respectively. The difference between the conditional integrity constraint for (44-b) and that for (44-a) is that in the former case the event for which f_1 prepares is consequent upon e , whereas the ordering of the culminating events in the latter case is not determined – just as in the case of the ‘when’-versions. Sentence (44-b) and its formalization thus do not qualify as representations for a causal relationship.

Next consider the use of the conditional as providing a precondition for an action. In example (44-j) the precondition is an event, that of getting a scholarship²¹. The precondition is best analyzed as a shift in the reference time. As we have seen, the unconditional use of *will* locates the reference time inside the preparatory fluent, thus implying that preparations have started. If *will* is used conditionally, the reference time is set by the antecedent of the conditional. Example (44-j) may then be formalized as follows. Let e be the event type ‘get a scholarship’, and f_1 the preparatory fluent for going. We then get

IF
 $?Happens(e, t)$ succeeds
 THEN
 $?Happens(e, R), HoldsAt(f_1, t), t > R > now$ succeeds
 &
 for all e' , $?Happens(e', t), Terminates(e', f_1, t)$ fails.

Observe that the first component of the consequent can be made to succeed if a statement of the form $Initiates(e, f_1, s)$ is added to the scenario.

The last case we have to consider is where the conditional expresses an obligation, as in examples (44-d), (44-e), (44-i) and (44-k). As has often been observed, the obligation expressed in such sentences is mutual, even when the sentences involve only a conditional, not a biconditional (see for example Geis and Zwicky [42]); the reader can easily extend the analysis to incorporate this feature. We concentrate on (44-d), reproduced here for convenience:

²¹Exercise 9 asks the reader to do the case where the precondition is a state.

(45) If you'll do the shopping, I'll give you money.

Sentences such as this are highly interesting, because they involve both the modal and temporal meaning of *will*. More precisely, the formal analysis proposed here shows that these meaning components are inextricably intertwined. Consider again the integrity constraint characteristic for *will*: for all e except for the canonical culminating event terminating f_1 ,

$?Happens(e, t), Terminates(e, f_1, t)$ fails.

It is easy to derive from this that aspect of the meaning of *will* which involves promising. A phrase like 'I'll give you money' could be analyzed as 'I have started preparations for a transfer of money to you, and no action of mine can interfere with the preparation'. If in the above integrity constraint, e runs over actions of self only, one gets precisely this meaning. We can also see how to formalize (44-d) as a conditional integrity constraint. Let f_1 be the fluent representing preparation for shopping, and likewise let f'_1 be the fluent preparing for transfer of money. The integrity constraint then becomes

IF
 $?HoldsAt(f_1, t), t > now$ succeeds
 &
 for all e , $?Happens(e, t), Terminates(e, f_1, t)$ fails
 THEN
 $?HoldsAt(f_1, R), HoldsAt(f'_1, R), R > now$ succeeds
 &
 for all e' , $?Happens(e', t), Terminates(e', f'_1, t)$ fails.

The proposed formalization has the consequence that the order of events is not determined, consistent with Comrie [18, p. 119]. We leave the analysis of the other cases of obligation to the reader.

5.4. Futurate progressive. The third manner in which a future event may be viewed from the perspective of planning, is the futurate progressive, as in examples (29-b), (30-b) and (31-b), repeated here for convenience:

- (46) a. John is flying to Chicago tomorrow.
 b. (Google) I am going unless some unknown demand stops me.
 c. ?*I am going if you go.

Sentence (46-a) shows that the futurate progressive is indeed a future tense in the sense that the event time lies in the future. Sentence (46-c) however shows that the reference point must be set to *now*, and cannot be supplied by the antecedent of a conditional; this is the difference between the futurate progressive and the constructions 'be going to' and 'will'. Lastly, sentence (46-b) shows that the plan may fail to achieve its goal, without the sentence being false; this is the difference between the futurate progressive and the construction using 'will'.

Unlike the construction ‘be going to’, however, the futurate progressive shares the applicability and coercibility conditions of the ordinary progressive: its default application is to activities and accomplishments, and if it is applied to other *Aktionsarten* these are coerced into one of the former.

These considerations can be summarized in the following

DEFINITION 34. *The futurate progressive requires the presence of an activity fluent f_1 , and a parametrized fluent $f_2(x)$ which are linked by a dynamics of the form*

$$\text{HoldsAt}(f_2(x), t) \rightarrow \text{Trajectory}(f_1, t, f_2(x'), d),$$

and a condition for the occurrence of the culminating event e of the form

$$\text{HoldsAt}(f_1, t) \wedge \text{HoldsAt}(f_2(c), t) \rightarrow \text{Happens}(e, t),$$

where c is some constant. The event e triggers an activity fluent f_3 via a condition of the form $\text{Initiates}(e, f_3, t)$.

The reference point is given by the integrity constraint that the query

$$?\text{HoldsAt}(f_1, \text{now})$$

must succeed.

The scenario to which the above statements are added should not imply that f_1 is terminated before e happens.

This definition gives the bare minimum of what has to be added to the scenario in case the futurate progressive is used, but it is worthwhile to add some detail concerning the precise nature of the dynamics. Consider the example (46-a). From our perspective, what we see here is a form of coercion. The formal mechanism behind coercion is explained in Chapter 11, but we will give an informal explanation here. The straightforward interpretation of the progressive would set the reference point equal to *now* and also place it inside the activity fluent *flying*. The temporal adverb ‘tomorrow’ renders this interpretation inconsistent. Formally, this means that the unification of the activity fluent f_1 with *flying* does not lead to a successful computation, and must therefore be abandoned. Since that was the only possibility for unifying f_1 with a constant, it must remain in the scenario as a variable, together with its companion, the parametrized fluent $f_2(x)$. But now the activity fluent f_3 can be eliminated via unification with *flying*. In this sense the use of the progressive introduces an additional fluent roughly meaning ‘preparation’. The reader may note that the same argument can be pushed through if we leave out the ‘tomorrow’ in the example, i.e. revert to sentence (46-b). The scenario will now contain an integrity constraint expressing that an activity of this kind can only be forbidden before it has begun, and again a contradiction can be derived. In Chapter 11, the reader will be asked to formalize this argument with the tools made available there.

For some more examples, consider

- (47) a. Tomorrow, I am flying to Chicago, unless my boss forbids me.

- b. ?*Tomorrow, I am flying to Chicago, if my boss allows me.
- c. Tomorrow, I am flying to Chicago if my health allows me.

In all cases, the reference time is *now*, which means that the intention to fly has been formed now, even though flying itself will start only tomorrow. This is the difference with the present progressive, where the reference point sits inside the *activity* to which the progressive is applied. Here we must assume the existence of a fluent $f_1 = \textit{preparation}$ which drives a parametrized fluent $f_2(x)$; if x has reached a certain threshold value, the canonical event e terminating the preparation and initiating the flying is triggered. The condition on the reference point means that $? \textit{HoldsAt}(\textit{preparation}, \textit{now})$ must succeed, and since *preparation* is an activity fluent, this also means that preparations must have started. The preparation fluent may however be terminated later. We can now see why sentence (47-b) is slightly odd: it would imply that preparations have started even though the permission to go is not yet given. For a preparation in *this* sense it cannot be true that if it goes on for long enough, the threshold value for the parametrized fluent $f_2(x)$ will be reached; it will not if the boss continues to withhold his permission. Therefore the content of (47-b) would normally be expressed by different means. But then how to account for (47-c), where the condition now refers to a state? The solution is left as an exercise, which is actually quite analogous to exercise 9.

6. Exercises

EXERCISE 8. *Reformulate the characteristic integrity constraint for will (cf. definition 33) in terms of the predicate Clipped.*

EXERCISE 9. *Give a formal analysis of*

- (48) a. (Google) *I will go if I have the money.*
- b. (Google) *I will go unless there is severe or dangerous weather.*

EXERCISE 10. *Give a formal analysis of*

- (49) *Tomorrow, I am flying to Chicago if my health allows me.*

CHAPTER 9

Tense in French: Passé Simple and Imparfait

1. Introduction

In this chapter¹ we apply the concepts developed in the preceding chapters to the formal semantics of the French tense system in particular Passé Simple and Imparfait. Much work has been done on French tenses within the framework of DRT (cf. Kamp and Rohrer's unfortunately unpublished [59]) or extensions thereof such as SDRT (for which see for example [27] and [4]). To explain the peculiar ways in which events described by sentences in Passé Simple form can be ordered in time, SDRT uses so-called rhetorical relations such as 'elaboration' or 'explanation'. Rhetorical relations have been evoked earlier (in [68]) to explain the temporal order of events in examples such as

- (1) Max fell. John pushed him.

The second sentence can be read as providing an explanation for the event described in the first sentence, and hence the temporal order is the reverse of the sentence order. But it is still legitimate to ask why this piece of discourse should be read as an explanation. This must be related to the meaning of the component sentences, since in the following two-sentence discourse a different interpretation is most salient:

- (2) Max fell. John held on to a rope.

These sentences describe a mountaineering accident in which the two events can occur simultaneously; the second event definitely does not provide an explanation for the former. The implied ordering of events seems to be derived from the meanings of the component sentences and the anaphoric relations. For instance, if the example is

- (3) Max fell. John held on to *the* rope.

a different interpretation becomes salient, one which John tries to break Max' fall by trying to secure the rope to which Max is attached.

It is claimed here that a much more insightful description can be obtained by taking a fully computational point of view, in which the event ordering is computed from the meaning of the sentences, world knowledge

¹Based on joint work with Fabrice Nauze, published as [86].

and anaphoric relations. Rhetorical relations are the output of such computations, not input. The computational framework is the one introduced in previous chapters: the event calculus formulated in constraint logic programming, with tenses described by integrity constraints.

2. Data

In this Section we provide some data pertinent to Passé Simple and Imparfait. We begin with a discussion of the Passé Simple (henceforth abbreviated as PS), and continue with examples of the interplay between Imparfait (Imp) and Passé Simple.

2.1. Examples involving the Passé Simple. We will start our discussion with a typical example of a narrative discourse using the PS where the events described are in temporal succession:

- (4) Pierre se leva, monta dans sa chambre, ferma la porte et alluma la radio.
(4×PS)

What can be said about the role of the PS in this example? Obviously, the PS conveys the information that all events are located in the past. More interestingly, it is often claimed that these events are to be viewed as punctual in the sense that there are no other events which could partition them (cf. our discussion of this notion in Chapter 3). The internal constitution of the events is not important; this means that the PS views events as perfective. The PS imposes a view of the events 'from the outside' and from a distance. This is then claimed to explain why multiple uses of the PS imply a succession of the events described. As the events are seen as punctual, irreducible and viewed from the outside, it is then natural to expect that two events in the PS are not simultaneous, and so that one is happening before the other. Then the obvious choice is to place first things first (unless explicitly stated otherwise). Hence in (4), the getting up of Pierre precedes his going up in his room, etc... This is why the PS is often considered to imply narrative succession.

Let us try to describe the above in a more formal manner. The most evident effect of the PS is to place the eventuality in the past of the speech time (this is what is known as "pure" tense information). We have now two options to account for the succession effect. We may assume, as in early versions of DRT, that the PS introduces a new reference point placed after a reference point previously introduced (this would amount to a direct representation of the "succession effect" of the PS). Alternatively, we may posit that the PS represents the eventuality as perfective and located in the past, and derive the succession effect from this, whenever it is appropriate.

We will choose the latter option, as it seems to be a better representation of the core meaning of the PS, succession being in our view only a (albeit

quite frequent) side-effect. In fact, a good counter-example to the unconditional validity of the succession effect of the PS was given by Kamp and Rohrer, here slightly changed to

- (5) L'été de cette année-là vit plusieurs changements dans la vie de nos héros. François épousa Adèle, Jean partit pour le Brésil et Paul s'acheta une maison à la campagne. (4×PS)

The first sentence introduces an event which gets divided in the following sentence (in SDRT, this phenomenon is known as the rhetorical relation of elaboration; it can also be viewed as a change of granularity in the description of events²). How this first event is divided cannot be determined from those PS sentences alone. In a way the first sentence 'asks' for an enumeration afterwards, and so the next verb phrases enumerate the list of changes in the life of the 'heroes', but in the absence of adverbs or ordering conjunctions (like *puis*) we cannot give the precise temporal relationship between those events. Hence we have here two phenomena: the first sentence gets divided by others (in a way this could be seen as contradicting the perfectivity of the PS), and furthermore the following PS sentences do not impose a natural ordering on the events described by them. One of the causes of this lack of ordering is that the VPs have different subjects: *François*, *Jean* and *Paul*. We can reformulate example (5) by removing one of the subjects as in

- (6) L'été de cette année-là vit plusieurs changements dans la vie de nos héros. François épousa Adèle et partit pour le Brésil, Paul s'acheta une maison à la campagne. (4×PS)

In sentence (6) we have now a succession of two events *François marrying Adèle* and then leaving to *Brazil*. However we still cannot derive any ordering of those two VPs with the third. We should also note that the inverse temporal order seems to be called for in the following example of Gosselin [45, p.117]

- (7) Pierre brisa le vase. Il le laissa tomber. (PS ×2)

It seems we can derive the explanation reading, even without the use of an explanative conjunction like *car*, and this for two reasons: first, the achievement of the first sentence is irreversible in the sense that the object of the sentence is changed for good after the achievement (*briser*), second, the anaphoric pronoun *le* in the second sentence refers to the vase, not to the broken vase which is the result of the first sentence, hence we expect that the second sentence applies to the not-yet-broken vase. We can further notice that the first sentence presupposes an action on the part of the subject

²A good description of how granularity of events can be treated formally can be found in Thomason [120].

Pierre on the vase (directly or indirectly), which causes the breaking. Furthermore, the subjects of the two sentences agree, and the pronoun of the second sentence refers to the object of the first sentence; and obviously to drop something is a plausible cause of breaking this same thing. It then seems natural that the second sentence actually describes the action that leads to the breaking of the vase.³ It should also be noticed that the fact that the two sentences in (7) are separated by a period is of major importance. If it is not the case, as in

(8) Pierre brisa le vase et le laissa tomber. (2×PS)

there is no ambiguity about the ordering of the two events described by the sentences: the breaking happens before the falling. Furthermore, even when the sentences are separated by a period we may get the ordering expressed by (8). If we add a further sentence, as in

(9) a. Pierre brisa le vase avec un marteau. Il le laissa tomber et s'en alla.
b. Pierre brisa le vase avec un marteau. Il le laissa tomber. Il s'en alla sans regarder sa femme.

the narrative seems to force the events to be ordered corresponding to the sentences.

Let us now change the examples (4), (5) and (7) somewhat, to determine when and why narration (of consecutive events) occurs or on the contrary breaks down. In example (4) we have a simple succession of events affecting one subject, in (5) we have several events affecting different subjects and occurring in a certain period of time but not explicitly ordered with respect to each other, and finally in (7) we have two events affecting one subject and one object in inverse temporal order. Now consider the following variations.

(10) a. Pierre monta dans sa chambre et ferma la porte. (2×PS)
b. Pierre ferma la porte et monta dans sa chambre. (2×PS)
c. Pierre ferma la porte et Jean monta dans sa chambre. (2×PS)
d. # Pierre monta dans sa chambre, ferma la porte, alluma la radio et se leva. (4×PS)
e. Cet été-là, François épousa Adèle, Jean partit pour le Brésil et Paul s'acheta une maison à la campagne. (3×PS)
f. Cet été-là, François épousa Adèle et partit pour le Brésil et Paul s'acheta une maison à la campagne. (3×PS)

Examples (10-a) and (10-b) describe a succession of two events accomplished by a single subject: monter dans sa chambre (go upstairs in his room) and fermer la porte (close the door). In example (10-a) Pierre goes first in his room and then closes the door whereas in (10-b) he first closes the

³We have provided such an extensive discussion of example (7) because there appears to be a general agreement in the literature on the impossibility of the PS to give an inverse temporal reading.

door and then goes in his room. As those eventualities are seen as perfective (this is the aspectual effect of the PS), are ascribed to one subject (this is a syntactic property of the sentence) and can hardly be done simultaneously (this is part of the semantics of those eventualities), the only possibility is that those two events are consecutive. However, the claim that the PS *implies* succession must be revised. All we get is that in a discourse in which the PS describes eventualities which have few semantic connections (note that going upstairs doesn't presuppose closing the door and vice-versa) and in which there is a unique subject, the order of the events is isomorphic to the utterance order. What is heard (or read) first, happens first.

Here are some more examples to show that the two factors identified, semantic connections and uniqueness of subject, indeed influence the reading of a piece of discourse. The importance of uniqueness of subject can be seen in examples (10-c), (10-e) and (10-f). The only difference between (10-b) and (10-c) is that in the latter the second VP has a different subject than the first. The correct reading of this sentence is probably that of a succession but the possibility of simultaneity is not excluded, as in (10-b). This sentence can describe the simultaneous actions of two subjects but would be inadequate to describe the inverse order.

Examples (10-e) (a simplified version of (5)) and (10-f) differ in that François is now the subject of two events. Furthermore those two events are successive but still in no particular relation to the third event. In (10-e) all subjects differ and we have no special ordering between the events.

Sentence (10-d) is usually incorrect because in normal circumstances Pierre's going into his room and closing the door presupposes (semantically) that he remains standing⁴. Hence to determine the temporal relation of a new PS VP with respect to a given sequence of PS VPs, all having the same subject, the meaning of the new VP must be compared with the possible lexical (semantic) information conveyed by the preceding VPs.

The next example involves aspectual information. The reader may have noticed that the VPs in the preceding examples are either accomplishments or achievements. The PS can also be used with states or activities, however.

(11) Il fut président. (PS)

In this example we obtain an inchoative reading. Thus, the proper English translation is *He became president*, and not *He was president*. The stative VP is coerced by the PS into its initiating event.⁵

⁴This is not the case for the VP 'switch the radio on'. Therefore the following sentence is correct.

(i) Pierre alluma la radio et se leva. (2×PS)

⁵Notice that the combination PS + stative VP does not logically imply an inchoative reading.

2.2. Examples involving the Imparfait. It is illustrative to begin this Section by citing several comments on this tense from the literature.

De Swart says in [27, p. 57],

sentences in the Imparfait are traditionally taken to describe background information that does not move the story forward.

De Swart here follows Kamp's view, which is motivated by the study of tenses in narrative context and where the fact that the Imp does not move the narration forward is contrasted with the fact that the PS does.

Gosselin, in [45, p. 199], does not put the emphasis on moving the story line forward, but notices that

the Imp refers to a moment in the past during which the process is going on, without being precise about the temporal location of the beginning and the end of the process.⁶

Sten in [111] focusses on its use as "present in the past":

L'imparfait sert à indiquer une action qui serait du présent pour un observateur du passé,...", (the Imp serves to indicate an action which would be present for an observator in the past).

Finally, all authors stress the anaphoric nature of this tense, in the sense that it cannot be used by itself but only with reference to another sentence or with temporal adverbials.⁷ We may summarize these positions by saying that the Imparfait is an anaphoric, imperfective past tense. We will now introduce some examples of the use of the particular we will not give examples of the so-called narrative

Imparfait; however the reader should be aware that these examples only partially represent the possibilities of the Imparfait. In Imparfait, or habitual and iterative readings.

The anaphoric and imperfective nature of the Imp can be seen in the following example

- (12) a. # Il faisait chaud. (Imp)
 b. Il faisait chaud. Jean ôta sa veste. (Imp, PS)

That sentence (12-a) is not felicitous is explained in Kamp's theory by the fact that an Imp sentence such as (12-a) does not introduce its own reference

(i) Mitterrand fut président de 1981 à 1995. (PS)

Here, we do not obtain an inchoative reading but just a perfective eventuality.

⁶The whole passage [45, p. 199] is relevant to our concerns: "L'imparfait renvoie donc typiquement à un moment du passé pendant lequel le procès se déroule, sans préciser la situation temporelle du début et de la fin du procès. Ce temps apparaît non autonome (anaphorique) et situe le procès comme simultané par rapport à d'autres procès du contexte, et comme se déroulant en un même lieu.

⁷See for instance the quote from Gosselin [45] given above, and Kamp and Rohrer [59, p. 35].

point, while there is no previous "reference point" to anchor the sentence⁸. In sentence (12-b), the Imp sentence is "attached" to the reference point introduced by the PS sentence, and the imperfective aspect of the Imp is due to the fact that the PS event happens while the Imp eventuality holds. It is however not a general rule an Imp eventuality contains its reference point, as is shown by the following examples.

(13) Jean appuya sur l'interrupteur. La lumière l'éblouissait. (PS, Imp)

(14) Jean attrapa une contravention. Il roulait trop vite. (PS, Imp)

The Imp sentence in (13) is viewed as a consequence of the PS sentence; clearly the light cannot blind Jean before he switched it on. De Swart, in [27, p. 59-61], maintains that the reference point for the Imp sentence is not the event described by the PS sentence, but rather its consequent state (the light is switched on). On this analysis we have simultaneity between the Imp sentence and its reference point. On de Swart's approach, the decision whether the Imp overlaps with the PS reference point or with its consequent state is made on the basis of *rhetorical relations* between the sentences; this theory is what is known as SDRT. De Swart calls this particular rhetorical relation *temporal implication* and she provides an analogous explanation for (14), introducing the rhetorical relation of *temporal presupposition*. In example (14) the Imp sentence is understood as being the cause of getting a ticket, hence even though the Imp sentence is placed after the PS sentence the activity *driving too fast* takes place before getting a ticket⁹.

We believe that the so-called 'rhetorical relations' are best viewed as symptoms of an underlying discourse organization based on planning, and that in this area explanations of much greater generality are possible than those provided by SDRT. Below we present a fully computational semantics for French tenses, built upon the computational mechanism of constraint logic programming. In this setup, rhetorical relations will turn out to be derived constructs, abstracting certain features of the planning computations.

3. Formalizing the Passé Simple and Imparfait

We will now formalize the examples of Section 2 using the event calculus formalism. This entails providing sufficiently informative scenarios which capture the meanings of the lexical items, and defining integrity constraints setting the reference points for PS and Imp. We provide explicit

⁸Notice that the reference point does not have to be introduced by a PS sentence; it can also be a temporal adverbial, or even the subject of the sentence, as in the following examples

- (i) a. Mercredi, il pleuvait. Jeudi, il faisait soleil. (Imp, Imp)
- b. Le grand-père de Marie était noir. (Imp)

⁹Nowadays, radar surveillance means that getting a ticket no longer necessarily terminates driving too fast.

computations in constraint logic programming which show how the order of the events described by the sentences in the discourse is determined. The main theme will be the importance of the anaphoric contribution of the tenses.

3.1. Passé Simple: scenarios and integrity constraints.

DEFINITION 35. *The effect of the Passé Simple is to introduce an integrity constraint of the form*

$$?HoldsAt(f, R), Happens(e, R), R < now,$$

where e is the event type derived from the VP which occurs in the PS, and the fluent f represents the context in which the PS is interpreted. If the context is empty, for example if the sentence considered is the first sentence of the discourse, we leave out the *HoldsAt* clause.

One may observe immediately that this stipulation accounts for two features of the PS: it presents the eventuality as perfective, and it places the eventuality in the past of the utterance time.

We also need a meaning postulate for the conjunction *et*, which is not simply the Boolean conjunction. The following stipulation seems to capture what we need.

DEFINITION 36. *If the PS occurs in the form ‘S et PS-VP’, then the fluent f occurring in the integrity constraint for the PS refers to the state which results from the event described by S (and not from material that was processed earlier). We view the construction ‘ S_1, S_2 et S_3 ’ as an iterated form of *et*, that is, as ‘ $(S_1$ et $S_2)$ et S_3 ’. Sentences conjoined by *et* are thus bound together more tightly than sentences conjoined by a period.*

3.1.1. *Succession (non)effects.* Recall that we have argued in Section 2.1 for the succession effect in PS narratives as a (default) side-effect of the semantics of the PS. Consider the sentences

- (15) a. Pierre monta dans sa chambre et ferma la porte. (2×PS)
 b. Pierre ferma la porte et monta dans sa chambre. (2×PS)

On the proposed analysis, the implied succession in sentences (15-a) and (15-b) should derive from the ordering of the sentences, the identity of the subject in the conjuncts related by *et*, and the perfectivity of the PS. We propose the following derivation of this succession effect.

Scenario and integrity constraints for sentence (15-a) comprise at least

- (1) (a) *Initiates*(*go-upstairs*(x), *upstairs*[x], t)
 (b) *?Happens*(*go-upstairs*(*Pierre*), R), $R < now$ succeeds
 (2) (a) *Initiates*(*close*(x , y), *closed*[y], t)
 (b) *?HoldsAt*(*upstairs*[*Pierre*], R'),
Happens(*close*(*Pierre*, *door*), R'), $R' < now$ succeeds

The formulas collected in 1. represent the information in the scenario induced by the first VP¹⁰. The first formula gives lexical information, and the integrity constraint gives the contribution of the PS. For completeness we should have added a *HoldsAt* clause as well, whose fluent has information about the context; but this clause would be irrelevant to the computation. The minimal model for this scenario looks as follows. Until reference time R , the fluent *upstairs*[*Pierre*] does not hold (by negation as failure), at R the event *go-upstairs*(*Pierre*) happens and initiates the fluent *upstairs*[*Pierre*].

The next pair of formulas introduces the semantic contribution of the second conjunct of (15-a). The lexical information introduced in 2a is straightforward. The choice of the integrity constraint 2b requires some explanation. The *Happens* clause of the integrity constraint represents the effect of the PS (perfective event in the past of the speech time), and the *HoldsAt* clause represents the context in which the PS is interpreted. Since the two clauses in (15-a) are linked by *et*, the fluent in this *HoldsAt* clause must refer to the state resulting from 1. We show that this choice accounts for the default succession effect of a sequence of PS sentences ascribed to a single subject. In the minimal model we have $R < R'$, as is shown by the following derivation. The first few steps in the argument look like this:

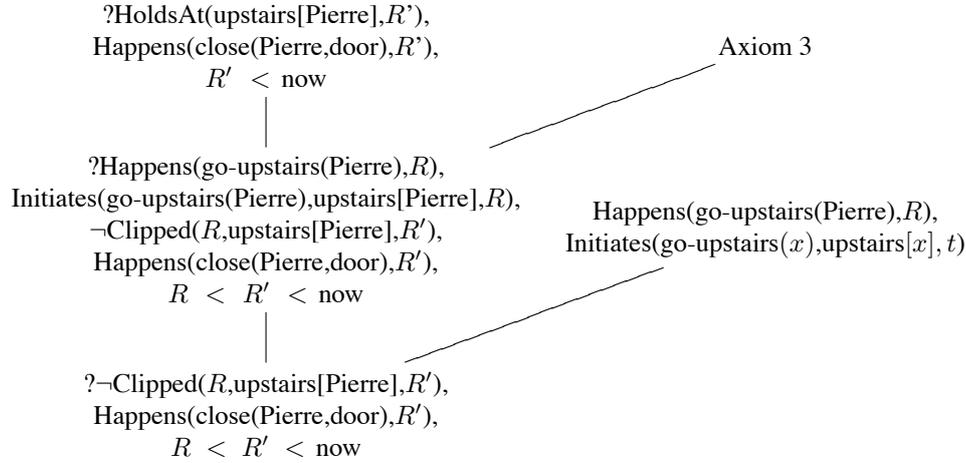


FIGURE 1. Effect of the second integrity constraint in (15-a).

The top node of this derivation contains the integrity constraint, i.e. a goal which is assumed to succeed. The derivation shows that the top goal can only succeed if the goal $?¬Clipped(R,upstairs[Pierre],R'), Happens(close(Pierre,door), R'), R < R' < now$ also succeeds. The subgoal $?¬Clipped(R,upstairs[Pierre],R')$ can be shown to succeed by negation as failure. This leaves us with the goal

$$?Happens(close(Pierre, door), R'), R < R' < now,$$

¹⁰Recall that *go-upstairs*(x) denotes an event type and *upstairs*[x] a fluent.

which means that any R' satisfying $Happens(close(Pierre,door),R')$ must also satisfy $R < R' < now$.

Having explained the main idea, we shall usually leave the last few steps, including the proof that the $?¬Clipped$ subgoal succeeds, to the reader.

Analogously, the scenario for sentence (15-b) looks as follows

- (1) (a) $Initiates(close(x,y), closed[y], t)$
 (b) $?Happens(close(Pierre, door), R), R < now$ succeeds
- (2) (a) $Initiates(go-upstairs(x), upstairs[x], t)$
 (b) $?HoldsAt(closed[door], R'), Happens(go-upstairs(Pierre), R'), R' < now$ succeeds

A derivation analogous to figure 1 shows that in the minimal model we must have $R < R'$. In both this case and the previous, the derivation does not branch, corresponding to the fact that sentences (15-a) and (15-b) have a single reading.

So far so good, but we also have to check whether the proposed integrity constraint does not overgenerate, that is, we have to look at examples where the succession does not hold. Let us first look at example (10-d), here adapted to

- (16) # Pierre monta dans sa chambre et se leva. (2×PS)

As we remarked above, sentence (16) is usually not felicitous because in normal circumstances the information conveyed by the second VP contradicts the lexical presupposition of the first. That is, you typically go upstairs walking, hence you need to be standing up; and if you are already standing up you cannot perform the action of getting up. The scenario and integrity constraints for this case have the following form

- (1) (a) $Initiates(go-upstairs(x), upstairs[x], t)$
 (b) $?Happens(go-upstairs(Pierre), R), R < now$ succeeds
- (2) (a) $HoldsAt(sitting[x], t) \rightarrow Initiates(get-up(x), upright[x], t)$
 (b) $?HoldsAt(upstairs[Pierre], R'), Happens(get-up(Pierre), R'), R' < now$ succeeds

Using only the material in 1. we obtain a minimal model where the fluent $upstairs[Pierre]$ is initiated at time R (hence this fluent does not hold before R). Viewed superficially, the material in 2. enforces that the event $get-up(Pierre)$ happens at time R' with $R' < now$. However, the integrity constraint in 2(b) is actually inconsistent with 2(a), as can be seen when we try to compute the query $?HoldsAt(upright[Pierre], t), R' < t$.

In the course of the derivation, the query $?HoldsAt(upright[Pierre], t)$ is transformed into $?HoldsAt(sitting[Pierre], R')$ which cannot lead to successful termination, as we do not have any information in the scenario pertaining to an event initiating this fluent (see figure 2). This means that also for t later than R' , $HoldsAt(upright[Pierre], t)$ is false. As a consequence, the goal 2(b) cannot succeed.

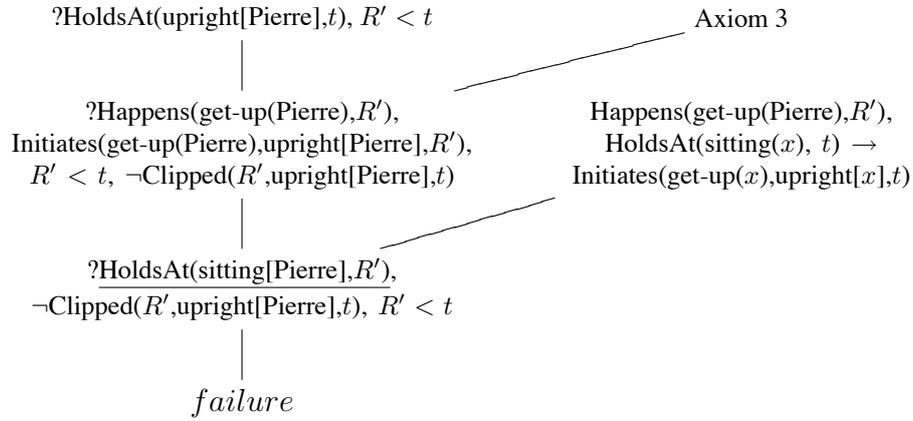


FIGURE 2. Conditions on the fluent *upright[Pierre]* in (16).

The following example presents a case where the order of the events described can actually be the inverse of the order of the sentences describing them.

(17) Pierre brisa le vase. Il le laissa tomber. (2×PS)

When introducing this example in Section 2.1 we noted that one may get the standard ordering back upon enlarging the discourse:

(18) a. Pierre brisa le vase avec un marteau. Il le laissa tomber et s'en alla.
 b. Pierre brisa le vase avec un marteau. Il le laissa tomber. Il s'en alla sans regarder sa femme.

Thus we must be able to explain the inversion of example (17) by a construction which is flexible enough to also accommodate examples (18-a) and (18-b).

It is important to mention at this stage that lexical expressions do not come with unique scenarios. A clause in a scenario can be seen as an activated part of semantic memory¹¹; which part is activated depends on all kinds of circumstantial factors. For this example we assume that the scenario for ‘break’ contains an open-ended set of clauses specifying possible causes of the breaking. We choose a simplified formulation here; e.g. 1b below could be derived in more elaborate scenario detailing the relationship between ‘drop’, ‘fall’ and impact on the ground. The simplified formulation is better suited, however, to illustrate the main points of the argument. Accordingly, we will take the scenario and integrity constraints to be of the following form, where we omit the *HoldsAt* components of the integrity constraints because they play no role in the derivation.

(1) (a) *Initiates*(*break*(*x*, *y*), *broken*[*y*], *t*)

¹¹As we have seen above, there is actually a close connection between logic programming with negation as failure and the spreading activation networks beloved by psycholinguists.

- (b) $Happens(drop(x, y), t - \epsilon) \rightarrow Happens(break(x, y), t)$
 (c) $Happens(smash(x, y), t) \rightarrow Happens(break(x, y), t)$
 \vdots
 (d) $?Happens(break(Pierre, vase), R), R < now$ succeeds
 (2) (a) $?Happens(drop(il, le), R'), R' < now$ succeeds¹²

A successful computation starting from the query

$$?Happens(break(Pierre, vase), R), R < now$$

is given in figure 3.

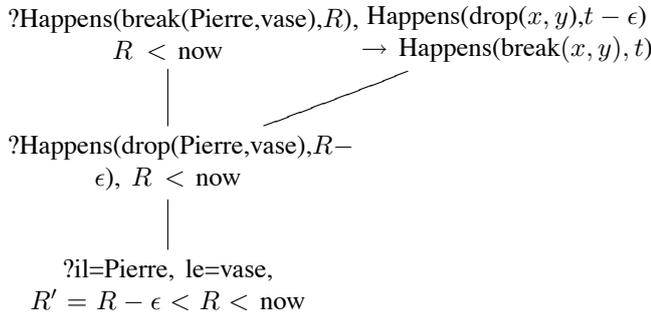


FIGURE 3. The effect of the two integrity constraints in example (17).

This computation explains the reversed order. Notice however that if we would bind the two sentences with an *et*, as in (8), the integrity constraint for the second sentence would be

$$?Happens(drop(il, le), R'), HoldsAt(broken[vase], R'), R' < now$$

By negation as failure, $Initially(broken[vase])$ is false, hence there must have been an event initiating the fluent $broken[vase]$. If ‘il’ is unified with Pierre and ‘le’ with the vase, this is impossible, because dropping the vase would have to take place before R' . If the fluent $broken[vase]$ goes proxy for the broken vase (as an object), it is possible to unify ‘le’ with $broken[vase]$, and get a coherent interpretation again. The examples (18-a) and (18-b) can be treated in the same manner.

Finally, we come to an example where the discourse does not determine the order of the events.

- (19) Cet été-là, François épousa Adèle, Jean partit pour le Brésil et Paul s’acheta une maison à la campagne. (3×PS)

The fact that we cannot order the enumerated events in sentence (19) is mainly due to the different subjects of the VPs. The temporal adverbial (Cet

¹²The anaphors ‘il’ and ‘le’ are really variables to be unified with concrete objects; we keep the words as handy mnemonics.

été-là) only places the events in a certain period of time, without implying anything about their order.

Scenario and integrity constraint might look as follows:

- (1) (a) *Initiates(begin, this-summer, t)*
(b) *Terminates(end, this-summer, t)*
- (2) (a) *Initiates(marry(x, y), married[x, y], t)*
(b) *?HoldsAt(this-summer, R₁), Happens(marry(François, Adèle), R₁), R₁ < now* succeeds
- (3) (a) *Initiates(leave-for(x, y), be-in[x, y], t)*
(b) *?HoldsAt(this-summer, R₂), Happens(leave-for(Jean, Brasil), R₂), R₂ < now* succeeds
- (4) (a) *Initiates(buy(x, y), have[x, y], t)*
(b) *?HoldsAt(this-summer, R₃), Happens(buy(Paul, countryhouse), R₃), R₃ < now* succeeds

What we obtain from the integrity constraints, by means of a derivation like the ones given above, is that there are times R_0 and R_4 such that *Happens(begin, R₀), Happens(end, R₄), R₀ < {R₁, R₂, R₃}* and *{R₁, R₂, R₃} ≤ R₄*. However, the order of R_1 , R_2 and R_3 cannot be determined.

3.1.2. *Inchoative use of the PS.* Consider again the example

- (20) Mitterrand fut président. (PS)

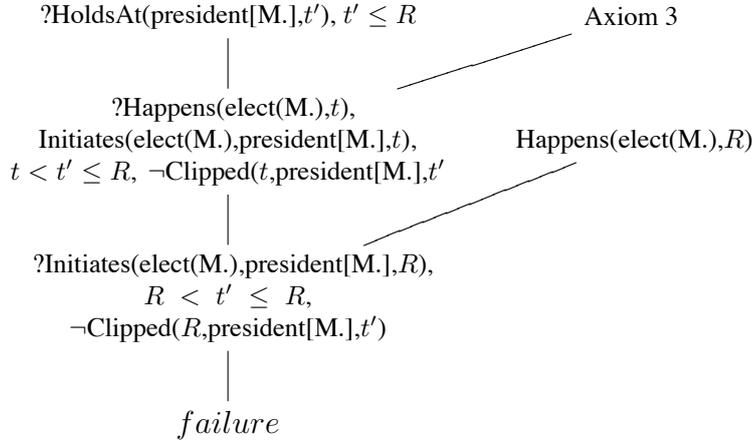
We have to derive formally that the PS applied to the stative expression ‘be president’, picks out the initiating event. Interestingly, when we are only given the fluent ‘be president’, there is no explicitly given event which warrants the application of the PS. Applying the PS means that a form of coercion is going on, in which the fluent is somehow transformed into an event. The proper way of doing this involves so-called hierarchical planning, as explained in Chapter 3. Since presidents are usually elected, the scenario for ‘be president’ will contain a statement such as 1a. This statement contains a reference to the event ‘elect’, which may thus figure in an integrity constraint. We thus get as scenario and integrity constraint

- (1) (a) *Initiates(elect(x), president[x], t)*
(b) *?Happens(elect(M.), R), R < now* succeeds

with corresponding derivation

As can be seen from figure 4, the fluent *president*[M.] does not hold before R . A similar derivation shows that it must hold after R .

3.2. Imparfait: scenarios and integrity constraints. The integrity constraint associated to the Imparfait must be very different from that associated to the Passé Simple, for example because an Imp sentence is not felicitous in isolation, unlike a PS sentence. An Imp sentence must be anchored by means of PS in the discourse. We therefore propose the following.

FIGURE 4. Fluent *president[M.]* before *R*.

DEFINITION 37. An *Imp VP* with an adjacent *PS VP* introduces an integrity constraint of the form

$?Happens(e, R), HoldsAt(f_1, R'), \dots, HoldsAt(f_n, R'), R < now, R' < now,$
 where e is some *PS event* of the discourse context (this sentence can precede or come after the *Imp sentence*), and f_1, \dots, f_n are the relevant fluents describing the *Imp verb phrase*.

The most relevant part of the integrity constraint for the *Imp* is the $HoldsAt(f, R')$ part. This part is what distinguishes the *PS* and the *Imp*: the *PS* introduces an integrity constraint of the form $Happens(e, R)$, possibly together with some other fluents that hold at R , while the integrity constraint associated to the *Imp* introduces a number of $HoldsAt(f, R')$ statements that are combined with the $Happens(e, R)$ statement of a *PS VP* in the discourse.

3.2.1. *Imparfait as background*. Consider the discourse

(21) Il faisait chaud. Jean ôta sa veste. (*Imp*, *PS*)

The scenario for these sentences must contain a fluent *warm*, and an event and a fluent for the achievement ‘take off one’s sweater’. For the latter we choose the event *take-off*, which terminates the fluent *wearing*; equivalently, we could have *take-off* initiate *not-wearing*. The integrity constraint anchors the fluent *warm*; note again that anchoring is only possible given a *PS VP*.

- (1) (a) $Terminates(take-off(x, y), wearing[x, y], t)$
- (b) $?HoldsAt(warm, R), HoldsAt(wearing[Jean, vest], R),$
 $Happens(take-off(Jean, vest), R), R < now$ succeeds

The following derivation (figure 5) shows that ‘Il faisait chaud’ really functions as a background.

The final query can succeed only if *warm* is true from the start. The next derivation (figure 6) shows the fate of the fluent $wearing[Jean, vest]$.

- (b) *Trajectory(light-on, t, blinded[x], d)*
 (c) *?Happens(push(Jean, y), R), HoldsAt(light-on, R'),*
HoldsAt(blinded[Jean], R'), R < now, R' < now succeeds

Figure 7 shows the derivation starting from the integrity constraint 2c. The substitution leading to success is indicated. The last query in the derivation can be made to succeed because the scenario makes no mention of the event of pushing the button to turn off the light, and we therefore obtain the conclusion $R < R' < now$.

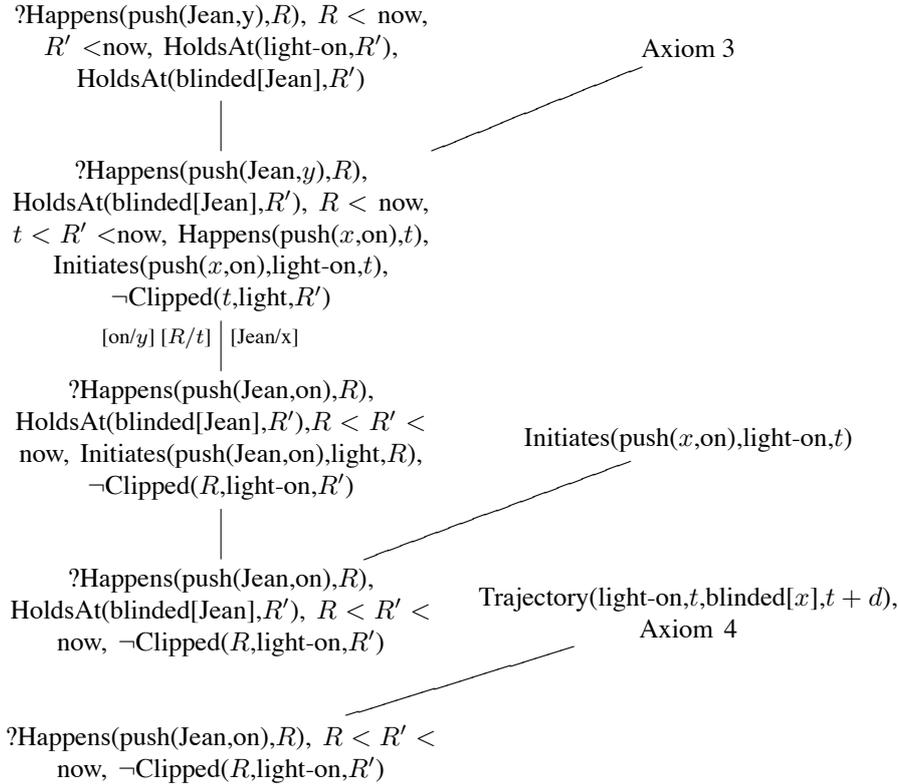


FIGURE 7. Integrity constraint in example (22).

3.2.3. *Imparfait in an explanatory context.* In the following discourse, the second sentence has the function of explaining the event described in the first sentence. The eventuality described in the second sentence should therefore be placed in its entirety before the event described in the first sentence. As will be clear by now, we do not want to have recourse to the rhetorical relation ‘explanation’ here – the intended order should fall out of a planning computation applied to the lexical material.

- (23) Jean attrapa une contravention. Il roulait trop vite. (PS, Imp)

The scenario and integrity constraints for this situation could be given by the following list. The first two statements have been included for convenience

only; it would make no difference if we pushed the beginning of the scene further in the past and introduced an event initiating driving.

- (1) (a) *Initially(driving[Jean])*
 (b) *Initially(speed[s])*
 (c) *Initiates(get(x, ticket), have[x, ticket], t)*
 (d) *?Happens(get(Jean, ticket), R), R < now* succeeds
- (2) (a) *Terminates(get(x, ticket), driving[x], t)*
 (b) *Terminates(get(x, ticket), speed[s], t)*
 (c) *Initiates(get(x, ticket), speed[0], t)*
 (d) *?Happens(get(Jean, ticket), R), HoldsAt(speed[s], R'), HoldsAt(driving[Jean], R'), s > limit, R < now, R' < now* succeeds

In the first step we start from the query in 2d, and we expand the derivation tree according to the different possibilities for the relation of R and R' ; we then recombine to get the possibilities $R' \leq R$ and $R < R'$. These possibilities are considered in the figures 9 and 10, respectively.

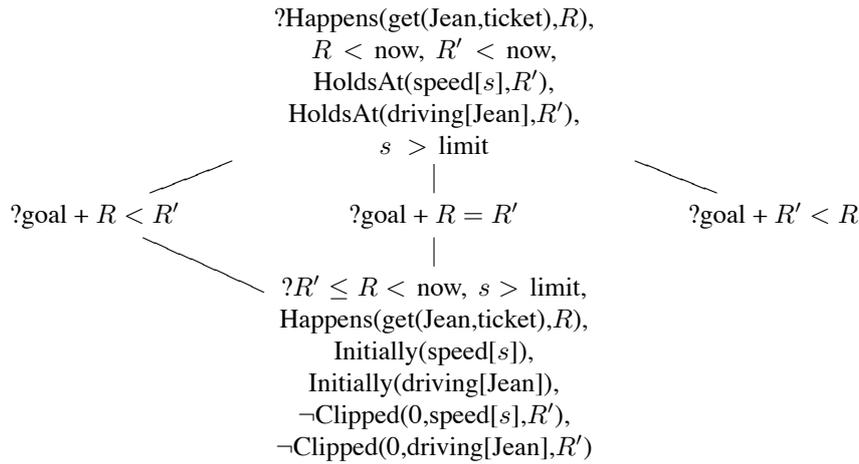
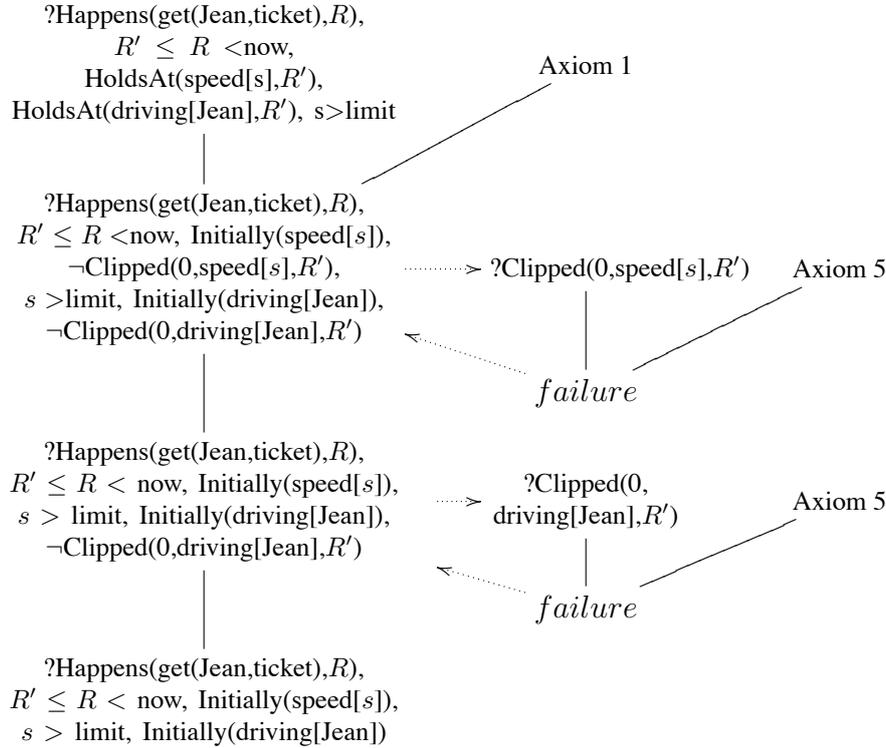


FIGURE 8. Integrity constraint in example (23).

The reader should notice that, for the sake of readability, in figures 9 and 10 we have processed the *Clipped* formulas in the same tree and have deleted them from the goal. The proper treatment of the integrity constraint would be, as described in Section 1, to first update the database with the *Happens* statement and then begin a new tree for the *Clipped* statement and check it for failure.

Derivation 9 terminates successfully with the constraint $R' \leq R < now$, because of part 1 of the scenario.

Now consider derivation 10 for the other possibility, $R < R'$. This derivation ends in failure, because the subderivation starting with the query *?Clipped(0, driving[Jean], R')* will end in success, given that getting a ticket at $R < R'$ ends in terminating the driving at that point.

FIGURE 9. Integrity constraint in example (23) with $R' \leq R$.

4. Coda

In Chapter 8 we formally introduced the English tenses as they apply to single sentences. Restricting attention to single sentences works to some extent for English, although on the analysis proposed here, the integrity constraint for the reference point must refer to the discourse of which the sentences forms part. We have discussed the French past tenses in some depth, because their very definition requires reference to other sentences. At this point we can do no better than quote from the eloquent ‘Apology and guide to the reader’ of Kamp and Rohrer [59]

... the mechanisms which natural language employ to refer to time cannot be properly understood by analyzing the properties of single sentences. Thus the methodology of modern generative grammar, which takes the single sentence as the basic unit of study is not, we believe, suited to this particular domain. Rather, a proper analysis of temporal reference must

(a) make explicit its *anaphoric* aspects – the systematic ways in which such devices of temporal reference as tenses and temporal adverbs rely for their interpretation on temporal elements contained in the antecedent discourse –

and

(b) discover the temporal organization of those conceptual structures which extended discourses produce in the human recipients who are able to interpret them.

This is precisely what we have attempted to do here.

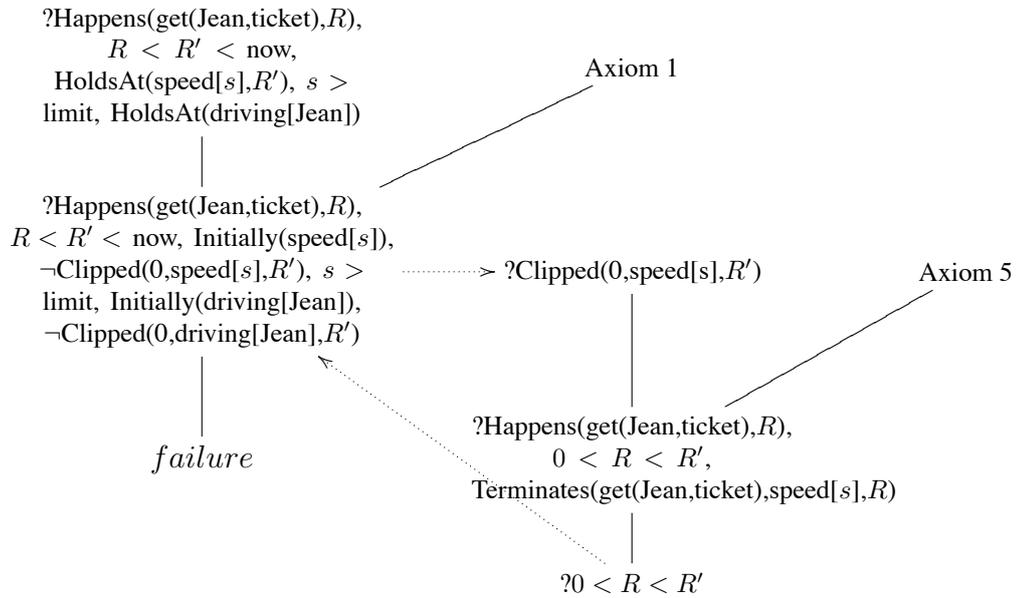


FIGURE 10. Integrity constraint in example (23) with $R < R'$.

5. Exercises

EXERCISE 11. In Section 5 of Chapter 2 the *Passé Antérieur* was discussed. Give a formal definition of the PA in terms of integrity constraints.

CHAPTER 10

Grammatical aspect

So far we have been concerned with semantic aspectual distinctions, irrespective of whether there is a morphological correlate to a given distinction. For example, the opposition perfective/imperfective is not grammaticalized in English, even though it plays an important role in the temporal semantics of English. We now turn to grammatical aspect proper.

1. The perfect

In a sense the perfect sits uncomfortably between the two stools of tense and aspect. It is not just concerned with ‘grammaticalized location in time’, since for instance the most important function of the present perfect is to stress the current relevance of some past situation or event, as exemplified by the grammaticality distribution in (1) (cf. Steedman [109, p. 898]).

- (1) a. *I have lost my watch but I have found it again.
b. I lost my watch but I (have) found it again.

On the other hand, it is also not concerned with aspect in the sense of ‘the internal temporal constitution of an event’. If anything, it seems to form a category of its own, concerned with the information value of events and situations. We shall nevertheless follow Comrie in assigning the perfect to the category of grammatical aspect. Figure 1 illustrates the role of the reference point *R* in the interpretation of the perfects.

The meaning of the perfect is however not exhausted by the relative positions of *E*, *R* and *S* as depicted in the diagrams in figure 1. Comrie [17, p. 56] distinguishes four typical uses of the perfect

- (1) perfect of result, as in
(2) John has arrived.

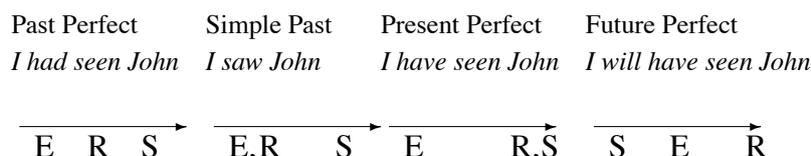


FIGURE 1. The reference point in the perfects

Here it is implied that the consequent state still holds *now*, so it is clear that the event *arrival* has current relevance.

(2) experiential perfect, as in

- (3) a. Bill has been to America.
b. Mary has run.

Here it is claimed that on at least one occasion Bill did go to America. Current relevance is somewhat less clear in this case, although it is taken to explain the oddity of

- (4) #Einstein has visited Princeton.

(3) perfect of persistent situation, as in

- (5) I've been waiting for hours.

It is a peculiarity of English to use the present perfect for this purpose, and not the present tense, as is the case in German

- (6) Ich warte schon drei Tage.

Indeed, what is peculiar in comparison to the previous cases, is that this case does not talk about a specific event (such as arrival, or departure), the current relevance of which is claimed. The existence of such an event is at most implied: e.g. waiting must have started at some point.

(4) perfect of recent past, as in

- (7) I have seen some old classmates recently.

Here, recency is taken to be a sufficient condition for current relevance. It is easy to see why, starting from this meaning, the present perfect has ousted the simple past in some languages, such as German and spoken French.

It can be seen, with some effort, that all these meanings of the present perfect derive from a basic meaning implying 'current relevance', provided the latter notion is generalized beyond 'the state f resulting from the event e mentioned in the sentence still holds'. Consider for example how case 2 can be viewed as a generalization of the basic meaning. In sentence (3-b) we are concerned with a non-telic activity, so there is no associated natural pair (e, f) such that the event e triggers the consequent state f by virtue of the meaning of the activity verb. That is, the lexical information for the activity 'run' will not contain a formula of the form $Initiates(e, f, t)$. Apart from lexical information, however, a scenario may also introduce contextual information, which may very well contain a formula $Initiates(e, f, t)$. This happens for instance if (3-a) is used in answer to the question 'Why is Bill wearing a Stetson all the time?'. Although the emphasis on the present relevance of the present perfect thus entails that it applies naturally only to

Aktionsarten where a consequent state is given, in the other cases the use of the present perfect leads to a process of coercion in which a consequent state is introduced in the scenario, along lines discussed in chapter 11 below. Since this coercion is so easily done in natural discourse that it goes almost unnoticed, we will not make our customary distinction between *Aktionsarten* here, and will give a general definition instead, based on the idea of current relevance.

1.1. Present perfect. There are two features of the present perfect to explain here. The first feature is the idea of current relevance. The second feature is Comrie's observation, cited in Chapter 8, that use of the present perfect, unlike that of the past tense, does not require common knowledge of the event time. In a lecture, Marie-Eve Ritz pointed out that this feature is increasingly exploited in Australian English, in the form of the 'police perfect', which the police use to report events which they have not themselves witnessed. Here is an example¹:

- (8) He apparently overtook on a blind bend on Kalamunda Road, he was heading East and he's then hit the kerb on the wrong side of the road and lost control of the motor cycle and crashed to the ground. He suffered severe head injuries and he's later died at Royal Perth Hospital.

It will be recalled from Chapter 8 that we first and foremost tried to find a temporal location for the reference point, in this case by means of an integrity constraint; the event time was then derived from the reference time. In the case of the past tense and the future tense that was of course easy, since they could be taken to coincide. The situation is different here, in that the reference point is given as *now*, and the event time has to be determined relative to *now* by means of the predicates *Happens* and *HoldsAt*. The current relevance of an event e is modelled by a fluent f such that the scenario under consideration contains a formula $Initiates(e, f, t)$. What the perfects share with the simple tenses is their definition by means of integrity constraints.

DEFINITION 38. *The present perfect can be applied if there are an event e , and a state f such that the scenario contains a formula $Initiates(e, f, t)$. The present perfect introduces the integrity constraint*

$$?HoldsAt(f, now) \text{ succeeds.}$$

Assume the full scenario is such that f is only initiated by e . Then the effect of the integrity constraint is that in a minimal model of this scenario there must have been a time $t < now$ such that $Happens(e, t)$; this can be seen by using the completion of axiom 3 of Chapter 4. Therefore the event time must lie in the past, but it need not be common knowledge: it is *inferred*

¹Marie-Eve A. Ritz and Dulcie M. Engel, 'Meaning, variation and change: the example of the English present perfect'; lecture at Linguistics Department, University of Amsterdam, February 14, 2003.

by the hearer using a minimal model which may well be different from the model the speaker has in mind. That is, the inferred event time need not be definable by a set of *Happens* and/or *HoldsAt* formulae, whereas by definition the reference time can always be characterized in this way. This gives the reference time a ‘public’ quality that the event time is lacking. In the past tense, where event time and reference time coincide, the event time also becomes public knowledge, but in the present perfect knowledge of the event time may well remain private. The development of the ‘police perfect’ is a natural consequence of this formal property of the perfect.

In this connection it is also of some interest to mention that in English it is seldom felicitous to add an explicit event time to a present perfect:

- (9) a. I got up at 5 a.m. this morning.
 b. *I have got up at 5 a.m. this morning.

Comrie [17, p. 54], discussing these data, writes ‘It is not clear that the mutual exclusiveness of the perfect and specification of the time of a situation is a necessary state of affairs in a language’, and goes on to cite the example of Spanish where precise specification is allowed. The analysis given here indicates that there is something pragmatically odd about a precise specification of the event time. The definite adverbial phrase *at 5 a.m. this morning* requires for its interpretation a suitable antecedent which is provided in (9-a) by the reference time defined by *Happens* and *HoldsAt* formulae as already indicated.

By contrast the perfect in (9-b) does not give an explicit definition of the event time. Hence the search for a suitable antecedent is much more complicated in (9-b) than in (9-a).

The Spanish examples mentioned in Comrie [17] may be similar to German examples using the perfect. At least in colloquial speech a semantic perfect does not exist any more in German. Therefore the German translation of (9-b) is fully grammatical in modern German.

- (10) Ich bin diesen Morgen um fünf Uhr aufgestanden.

1.2. Past perfect. This construction actually has two meanings, as exemplified by (cf. [17, p. 56]):

- (11) a. Bill had arrived at 6 p.m.; in fact he came in at 5 p.m.
 b. Bill had arrived at 6 p.m. and had left again at 7 p.m.; the inspector did not get there until 8 p.m.

In the first case, 6 p.m. functions as the reference point in the past, and it is stated that the consequences of his arrival, namely being there, still hold. This sense of the past perfect is represented in the picture above. In the second case we are only concerned with a past-in-the-past: 6 p.m. is now acting as event time, and we only express that Bill’s arrival preceded some

other past situation. The reference time here coincides with the event time, i.e. 6 p.m., so nothing new is involved here.

It is the first meaning of the past perfect that is of most interest to us here. The following definition captures the essentials.

DEFINITION 39. *The past perfect can be applied if there are an event e , and a state f such that the scenario contains a formula $Initiates(e, f, t)$. The past perfect introduces the integrity constraint*

$$?HoldsAt(f, R), R < now \text{ succeeds.}$$

This definition has the effect of making the integrity constraint

$$?Happens(e, t), t < R \text{ succeeds}$$

satisfiable. As in the case of the present perfect, the precise time of occurrence of e is a matter of private not public knowledge. The use of the past perfect requires that the reference time itself is introduced by a temporal adverbial or a *when*-phrase, as in

- (12) a. Bill had arrived *at 6 p.m.*
 b. Bill had arrived *by the time the match started.*
 c. Bill had arrived *when the mailman brought the package.*

This shows that the integrity constraint mentioned in definition 39 will generally have more components. For example, as indicated in Section 4.2 of Chapter 8, sentence (12-c) would correspond to an integrity constraint

$$?Happens(e', R), HoldsAt(f, R), R < now \text{ succeeds,}$$

where e' is the event 'mailman bringing the package'.

1.3. Future perfect. The future perfect can be analysed analogously, the only difference being that the integrity constraint now expresses the future location of the reference point:

DEFINITION 40. *The future perfect can be applied if there are an event e , and a state f such that the scenario contains a formula $Initiates(e, f, t)$. The future perfect introduces the integrity constraint*

$$?HoldsAt(f, R), R > now \text{ succeeds.}$$

Observe that, since the event time is inferred and not explicitly given, nothing is implied logically about the relative position of event time and speech time, i.e. *now* (cf. Comrie [18, p. 70–74]). For example, in

- (13) John will have finished his manuscript by tomorrow.

it is a conversational implicature, but not a logical consequence, that John has not yet finished his manuscript. This distinction is illustrated by the following dialogue

- (14) "Will John have finished his manuscript by tomorrow?" "Yes, in fact he finished it yesterday."

1.4. Exercises.

EXERCISE 12. Write the integrity constraint corresponding to the ‘past in the past’ interpretation of the past perfect.

2. The progressive

The following small narratives (lifted from Google) make perfect sense

- (15) a. By 1887, Adelia was building a house in Washington and had sold Belmont to a land development company. Later that year, while on a shopping trip to New York, Adelia contracted pneumonia and died in The Fifth Avenue Hotel.
- b. Carlos Thompson was building a house with neighbors one day in 1984 when a wooden beam fell and landed on his foot. Since that fateful day, Thompson’s life has never been the same.

Upon reflection, there is something paradoxical about both these narratives. Whereas it belongs to the meaning of the accomplishment *build a house* that the activity (‘build’) is directed toward the consequent state of a finished (‘built’) house, the actual occurrence of that consequent state can be denied without contradiction². So how can a seemingly essential component of the meaning be denied, without affecting the meaning itself? This is known as the ‘imperfective paradox’. The literature is replete with attempted resolutions of the paradox, ranging from explaining the problem away (cf. the recent Michaelis [78]) to various invocations of possible worlds (see Dowty [30], Landman [67] or de Swart [26]). Possible worlds solutions are based upon the idea that

The progressive picks out a stage of a process/event which, if it does not continue in the real world, has a reasonable chance of continuing in some other possible world [26, p. 355].

but differ in the (largely informal) descriptions of the possible worlds used. For example, [30] claims that the following are equivalent

- (1) ‘Mary is drawing a circle’ is true in the actual world
- (2) ‘Mary will have drawn a circle’ is true in all so-called ‘inertia worlds’, worlds which are identical with the present world until ‘now’, but then continue in a way most compatible with the history of the world until ‘now’.

Thus these approaches are intensional in the formal sense of using possible worlds. In fact, most authors (though not all; see below) would agree that the progressive creates an intensional context: even though Carlos Thompson may have stopped building at a stage when it was unclear whether he was building a house or a barn, still only one of

- (16) a. Carlos was building a house.

²The full stories make clear that the respective houses were never finished by the protagonists.

- b. Carlos was building a barn.

can be true of the situation described.

Our solution (first proposed in [47]) will use the event calculus, but before we go into this, we discuss Michaelis' attempt in [78] to explain the problem away; this will then show why an elaborate machinery is necessary.

Explicitly denying that the progressive creates an intensional context, Michaelis writes

Under the present proposal, the Progressive sentence *She is drawing a circle* denotes a state which is a subpart not of the accomplishment type *She- draw a circle* but of the activity type which is entailed by the causal representation of the accomplishment type. Since this activity can be identified with the preparatory activity that circle drawing entails, circle drawing can in principle be distinguished from square drawing etc. within the narrow window afforded by the Progressive construal [and] does not require access to culmination points either in this world or a possible world ... [78, p. 38].

We find this rather doubtful. Without access to a person's intention it may be very hard to tell initially whether she is drawing a circle or a square, building a barn or building a house. But that person's intention in performing an activity is characterised precisely by the associated consequent state, even though the latter cannot yet be inferred from the available data.

Here the event calculus comes to our rescue, because the notion of goal or intention is built in from the start. In the event calculus, an activity comes with a scenario which describes a plan for reaching the goal. However, unlike approaches such as Parsons' [87], where one quantifies existentially over events, the scenario is a universal theory and does not posit the occurrence of the intended consequences, i.e. the attainment of the goal. Even if the plan is appropriate for the goal, attaining the goal is guaranteed only in minimal models of the scenario combined with the axioms for the event calculus, in which no unforeseen obstacles occur. Thus, the meaning of an accomplishment (as embodied in the scenario) involves a culminating event *type* (which therefore must exist); but there are no existential claims about the corresponding event *token*³. Type and token are handled by different mechanisms.

We will now make the above considerations slightly more formal. Consider the sentence

- (17) Carlos is building a house.

In this example, lexical material ('build a house') is combined with the present progressive to create the sentence (17). Semantically, the lexical material is represented by a scenario, in fact the scenario introduced in Section 2.1 of Chapter 7, which is reproduced here for convenience:

³And similarly for the state consequent upon a culminating event type.

- (1) *Initially*(*house*(*a*))
- (2) *Initiates*(*start*, *build*, *t*)
- (3) *Initiates*(*finish*, *house*(*c*), *t*)
- (4) *Terminates*(*finish*, *build*, *t*)
- (5) *HoldsAt*(*build*, *t*) \wedge *HoldsAt*(*house*(*c*), *t*) \rightarrow *Happens*(*finish*, *t*)
- (6) *Releases*(*start*, *house*(*x*), *t*)
- (7) *HoldsAt*(*house*(*x*), *t*) \rightarrow *Trajectory*(*build*, *t*, *house*(*x* + *g*(*d*)), *d*)

The present progressive is applicable because the scenario features a dynamics. The temporal contribution of the present progressive is the integrity constraint

$$?HoldsAt(build, now) \text{ succeeds,}$$

where ‘build’ is the activity that drives the dynamics, in accordance with the crucial presupposition for the use of the progressive⁴. The following theorem then provides information on the default character of the progressive. A computational proof is given in Section 3.

THEOREM 6. *Let \mathcal{P} be the logic program consisting of EC and the scenario given in 2.1 of chapter 7. Suppose \mathcal{P} is extended to \mathcal{P}' so that the query $?HoldsAt(build, now)$ succeeds in \mathcal{P}' . Suppose $\lim_{d \rightarrow \infty} g(d) \geq c$. Then $comp(\mathcal{P}')$ has a unique model, and in this model there is a time $t \geq now$ for which $HoldsAt(house(c), t)$. By virtue of the stipulation that $House(house(c))$, there will be a house at time t .*

A by now familiar argument shows that the integrity constraint has the effect of introducing a time at which the start event happens. The effect of taking the completion is that in a model, only those events occur which are forced to happen due to the scenario; similarly, only those influences of events are considered which are explicitly mentioned in the scenario. In this particular case, since the scenario does not mention an event ‘Accident’ with its attendant consequences, the completion excludes this possibility; and similarly for other possible impediments to completing the construction. It is then a consequence of the general result corollary 1 of theorem 4 in chapter 5 that the completion actually has a unique model, thus proving theorem 6. One may reformulate the preceding theorem as a result on entailment between integrity constraints, following the ‘dynamic’ meaning of

⁴Darrin Hindsill drew our attention to the Papua language Kalam, where to express the mono-clausal accomplishment *I am building a house for you* three separate clauses are required :

- (i) kotp gy, np ñnp gspyn
house having-built-SS you intending-to-give-SS I-am-doing
I am building a house for you

Her ‘SS’ means ‘same subject’. Note that there are separate clauses for the finished house (in the future) and the building-in-progress. Languages such as Kalam, in which event structure is coded much more explicitly than in English, provide some evidence for the appropriateness of the above representation.

integrity constraints as explained in Section 1 of Chapter 8. Recall Dowty's intuitive explanation of the progressive as:

‘Carlos is building a house’ entails that ‘Carlos will have built a house’ in all inertia worlds.

Now read ‘minimal model’ for ‘inertia world’: note that whereas ‘inertia world’ is of necessity an informal concept, ‘minimal model’ is defined precisely and moreover computable. We get, using definition 40

COROLLARY 4. *Under the same assumptions as theorem 6: the query*

?HoldsAt(house(c), R), R > now

succeeds.

We leave the proof of the corollary as an exercise to the reader (cf. 13 below). A glance back at Section 1.3 shows that the query in corollary 4 is precisely the one occurring in the integrity constraint for the future perfect.

Notice, however, that the existence of a time at which the house is finished is only guaranteed in a minimal model. Thus, this inference is non-monotonic: if we obtain more information, the conclusion may fail. For example, Carlos' story as given in (15-b) expands the scenario with

(8) *Terminates(Accident, build, t)*

(9) the integrity constraint

?Happens(Accident, t), t < now, ¬HoldsAt(house(c), t) succeeds

Together, these imply that building will be clipped before the house is completed. However, there is no longer an imperfective paradox. The meaning of ‘build a house’ is the same in both (17) and (15-b), since essentially the same scenario is involved in both cases. It is true that *Terminates(Accident, build, t)* was not included among (1–7), but it would have made no difference had we done so, since *Accident* did not occur as argument of *Happens* there. The crucial point is that in both cases the successful completion of the building process is present as a *goal*, and not as an event *token*.

3. **A computational proof

In this Section we provide a logic programming proof showing the truth of theorem 6. This is included as an illustration of the computational content of the theory – it can be skipped without loss. The proof has the following structure. On the basis of the scenario and the axioms for the event calculus, we have to show that the query

?HoldsAt(house(c), t₁), t₁ ≥ now)

is satisfiable, given the integrity constraint that

?HoldsAt(build, now) succeeds.

We start with a derivation having *?HoldsAt(house(c), t₁), t₁ ≥ now* as the top query (figures 2 and 3) and we apply the integrity constraint when that derivation cannot be developed any further.

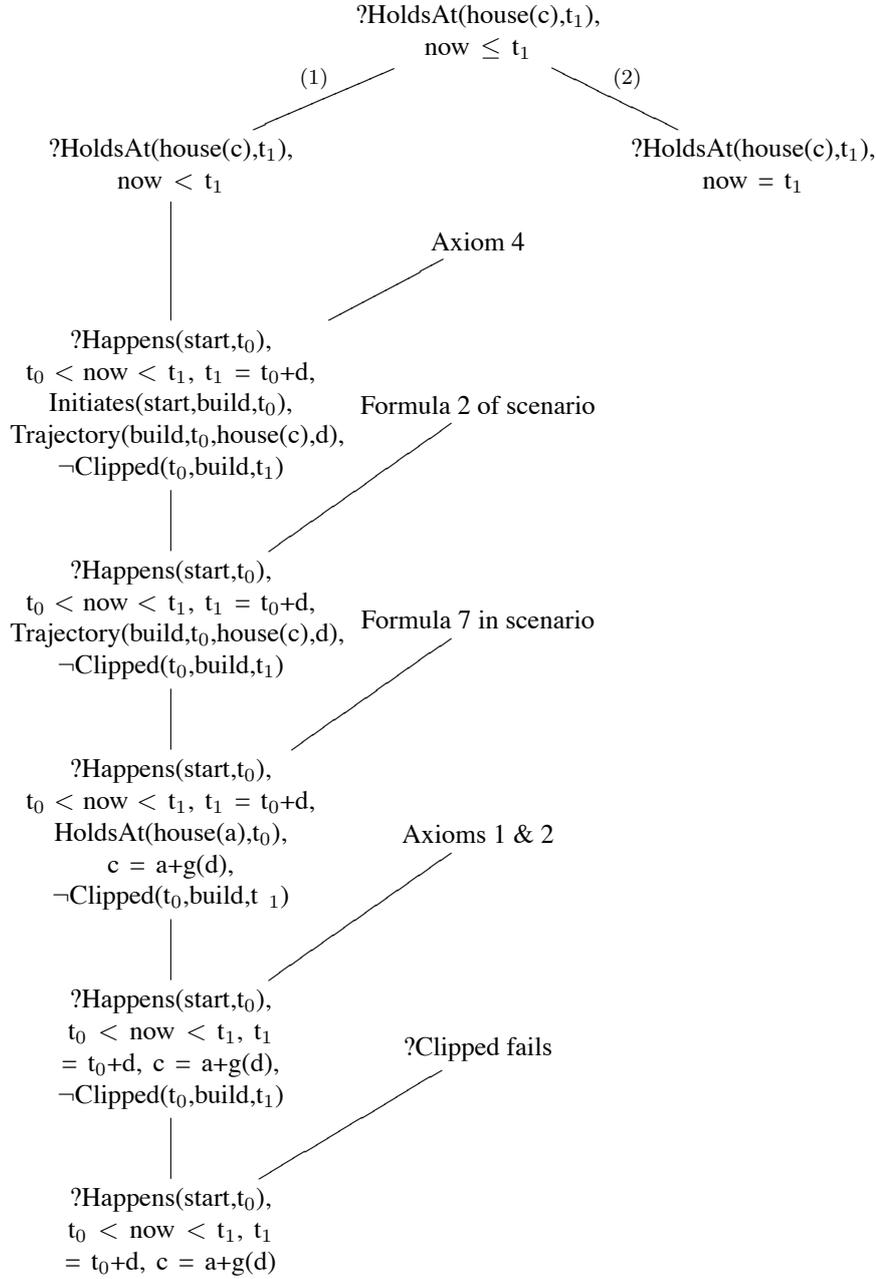


FIGURE 2. Proof of theorem 6: left branch

In both cases we must ensure that the query $?Happens(start, t_0)$, $t_0 < now$ succeeds. We know that $?HoldsAt(build, now)$ succeeds, and by applying axiom 3 and formula 2 of the scenario, the latter query reduces to the former. We are thus left with two constraints: $?t_0 < now < t_1, t_1 = t_0 + d, c = a + g(d)$ for the left branch and $?t_0 < now = t_0 + d, c = a + g(d)$ for the right branch. By the choice of g , both constraints are satisfiable.

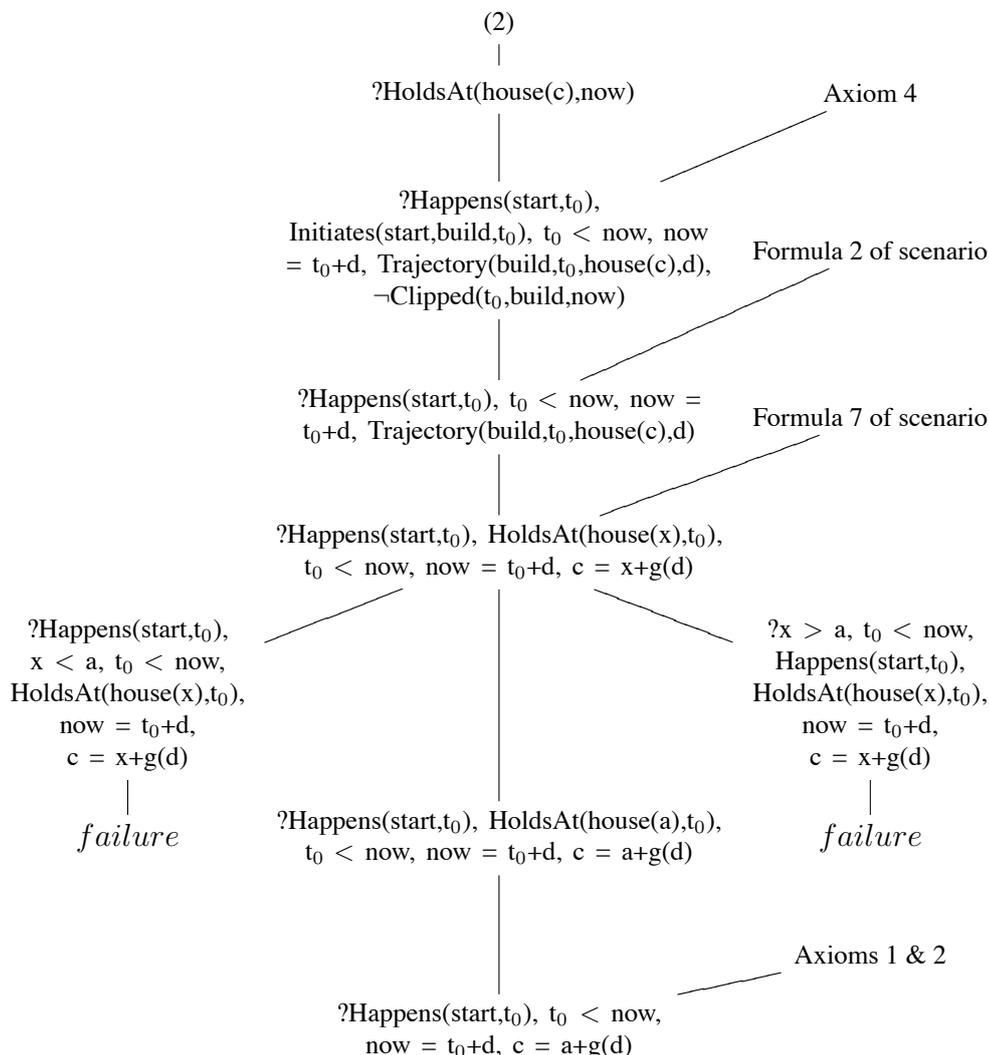


FIGURE 3. Proof of theorem 6: right branch

4. Comments on the literature

The literature on the progressive and the imperfective paradox is vast, and it is impossible to do justice to all of it here. We can only discuss a handful of papers whose examples have attained the status of benchmarks against which any proposed solution of the imperfective paradox must be tested.

4.1. Asher. We start with a few remarks comparing our proposal with Asher [2] (see also [4]), who was actually the first to provide a formalized treatment of the progressive within a nonmonotonic logic. The characteristic feature of Asher's theory is that he assimilates the progressive to generic expressions. Applied to the accomplishment 'cross the street', this means

that the following entailment holds by virtue of the meaning of the progressive

- (18) When you are crossing the street, you typically get to the other side eventually.

The conditional in sentence (18) is interpreted formally as a nonmonotonic implication $>$ ⁵, which satisfies $p, p > q \vdash q$, but not necessarily $p, r, p > q \vdash q$. This allows for canceling the implied consequent when the antecedent is expanded with, say, ‘and are hit by a truck’.

Asher’s procedure requires, first, that one is able to associate a typicality-judgment to each use of the progressive and, second, that these judgments can be ordered according to priority. That prioritized defaults are necessary can be seen from an example discussed in Naumann and Piñón ([85]; more on this paper below):

- (19) Rebecca was running across the minefield.

It now seems that the use of the progressive is not governed by the analogue of (18)

- (20) When you are running across a minefield, you typically get to the other side eventually.

the default assumption is rather

- (21) When you are running across a minefield, you typically don’t make it to the other side.

Asher solves the problem by assigning priorities to defaults, in such a way that specific defaults such as (21) get priority over more general defaults of type (20) (more general, because based only on the running across, without taking into account the object of the preposition). Not the least problem raised by Asher’s analysis is to say what ‘typically’ means: does it mean ‘usually’, or is it a conventional expectation? There is also a subtle difference between ‘typically’ and ‘in the absence of information to the contrary’, both of which are used by Asher as intuitive motivations. Formally, ‘typically’ is an expression that belongs to the object language, and hence can be modelled by a generalized quantifier or conditional, whereas the second expression denotes a concept of the meta-language, for which some form of default reasoning is a more appropriate formalization.

By dispensing with a genericity interpretation of the progressive, we do not need to provide an interpretation for ‘typically’. In our approach, we must provide scenarios and dynamic laws which, upon applying program completion, yield predictions which can then be tested against whatever typicality-judgments are available. However, the machinery can be put to

⁵Actually, in the context of predicate logic it is a binary generalised quantifier.

work even when these judgments are absent; it is then just a matter of establishing, what, if anything, is true in certain minimal models. Asher's approach is thus compatible with ours, but the claim is that here defaults such as (18) or (21) are derivable, not introduced as meaning postulates. Furthermore, it seems that the distinction between 'John is crossing the street' and 'Mary is drawing a circle' (in the latter case, but not the former, the direct object is a true incremental theme) is not easily explained in Asher's setup, since his language does not seem to be expressive enough to accommodate such distinctions.

4.2. Landman. This brings us to Landman's example ([67])

(22) God was creating a unicorn, when he changed his mind.

The scenario is that God, after much preparatory work, was just about ready to create a unicorn in one stroke (no partial unicorns here), when he changed his mind; in this scenario sentence (22) is true. The problem posed by the sentence is, how to interpret the quantifier 'a unicorn', since it clearly cannot quantify over unicorns in the real world. This seems to make (22) analogous to 'Mary tried to find a unicorn', so that the progressive creates an intensional context.

On the analysis presented here, 'create a unicorn' is an accomplishment on a par with 'build a house': 'create' is an activity which drives a dynamics, the natural culminating event of which is the sudden coming into existence of a unicorn. The progressive is intensional in so far as the culminating point need not be reached in all models. For instance in the minimal model for (22) the culminating event does not occur. Therefore we do not know that the fluent expressing the existence of a unicorn holds. The difference between (22) and say *crossing the street* is given by properties of the scenario function g describing the state of the partial object. In Section 3 of chapter 5 *distance* was assumed to be continuous, monotonically increasing function. The function g for (22) however is according to Landman's scenario a discontinuous one.

4.3. Naumann and Piñón. The paper [85] contains a number of interesting observations on the progressive, and we will discuss some of their examples here.

For a start, consider the sentences

- (23) a. Rebecca is drawing a square.
b. Rebecca is drawing.

uttered when Rebecca has just drawn a single straight edge (which could also form part of, say, a triangle). Clearly, sentence (23-b) is true in this situation, but Naumann and Piñón argue that, unless we make some assumptions concerning Rebecca's intentions, the truth value of (23-a) cannot be established. Accordingly, in their proposed semantics, intention (modelled as a primitive accessibility relation) plays an important part.

In our setup, truth is evaluated relative to a discourse model. That model should be such that the scenarios for all the lexical items occurring in the discourse are true there; moreover, the integrity constraints corresponding to the tenses must be satisfiable in the model. In both (23-a) and (23-b) the integrity constraint is of the form $?HoldsAt(draw, now)$ succeeds. The difference between the two cases is that in the situation sketched, the real world does not support a dynamics connecting ‘draw’ and ‘square’. This means that a precondition for the truth of (23-a) – the discourse model verifies the scenarios for the various lexical items – cannot be satisfied, and hence the truth value of (23-a) is not defined.

Next, consider

(24) Rebecca is swimming across the North Sea.

uttered when she is 100 meters off shore at Zandvoort Beach, heading west.

This is an interesting example, because it may be considered true or false, depending on whether or not Rebecca’s intention is taken into account.

Suppose first that the sentence is uttered by an observer who has no access to Rebecca’s intention, only to the objective dynamics. In this case one is likely to say that (24) is false. Formally, this is because the limit of the function g will be a number far smaller than the width of the North Sea. Therefore the scenario cannot be true in the discourse model, and this fact renders (24) formally false. More generally, we have that for a statement ‘ A is φ -ing’ to be true, the statement ‘ A has φ -ed’ should at least be possible.

On the other hand, if it really is Rebecca’s intention to swim across the North Sea, then sentence (24) is generally considered to be true, never mind the objective dynamics which make it unlikely that she will get to the other side. As long as her personal view of the dynamics does not make attaining the shore of Britain impossible, sentence (24) will come out true.

Lastly consider the sentence (see also Dowty [30])

(25) #The coin is coming up heads.

uttered after flipping a coin and before the coin has landed. In this context, the utterance of (25) seems to be infelicitous, and likewise when ‘is’ is replaced by ‘isn’t’.

Well, is it? Here’s a quote from a synopsis of Tom Stoppard’s play ‘Rosencrantz and Guildenstern are dead’⁶:

(26) Rosencrantz and Guildenstern are tossing coins, and the coin is coming up heads all the time. Guildenstern is considering reassessing the laws of probability.

The difference between (25) and (26), however, is the expression ‘all the time’, which coerces the expression ‘come up heads’ to an activity. So there

⁶Found on the website of the Arden Theatre in Philadelphia.

really is something to be explained: why the progressive does not seem to be applicable in case (25). Again, note that it must be something related to the use of the futurate progressive; it is not connected to the future tense *per se* as the following sentence shows

(27) (Google) The coin is going to come up heads.

The characteristic feature of the situation is of course that for all practical purposes, coin tossing is indeterministic. In the case at hand there does indeed exist a dynamic law, which deterministically relates an initiating event (flipping the coin with a definite speed and rotational motion), the action of gravity, a trajectory and an outcome (heads). This dynamical law could even be formulated in the event calculus, but that is not the point – which is that there is no dynamical law acting at the level of granularity that we are typically concerned with in ordinary life, where we can see, for example, that a coin is falling while rotating, without being able to fix the parameters for the motion in any great detail. This means that the *Aktionsart* of ‘come up heads’ is not, as one might think at first, an accomplishment, but rather an achievement. Hence the progressive is not applicable without coercing the expression ‘come up heads’ into either an activity or an accomplishment.

4.4. Bonomi. In [9], Bonomi has tried to isolate another aspect of the intensionality of the progressive in what he calls the *Multiple–Choice Paradox*, which he takes to be as central as the imperfective paradox. To solve the problem, he develops an intricate semantics for the progressive, which we shall not go into, since we believe that the paradox can be treated with the machinery developed here.

The Multiple–Choice Paradox can be illustrated by the following story:

Leo, who has just left Dijon in his car, has decided to spend the night in one of the following three cities: Besançon, Metz or Paris. He drives on the autoroute which runs in the direction of these cities (before branching into three different roads). In each of the cities he has reserved a room. However, before he has made up his mind, his car breaks down. Now suppose that Besançon, Metz and Paris are the only cities in France where there will be a concert of Baroque music that night. Then the sentence (28), uttered shortly before the car breaks down, appears to be true

(28) Leo is going to a French city where today there is a concert of Baroque music.

But given that the cities mentioned are the *only* cities where there is a concert of Baroque music that night, we should also have that the following sentence is true

(29) Leo is going to Besançon or Leo is going to Metz or Leo is going to Paris.

However, each of the sentences *Leo is going to Besançon*, *Leo is going to Metz* and *Leo is going to Paris* appears to be false, hence so is the disjunction. The paradox is then that the true sentence (28) logically implies the sentence (29) which is false.

The task at hand is thus to provide formal correlates of (28) and (29) which explains their logical relation. We give an informal description of the solution, and leave the formalization to the reader as exercise 15.

Sentence (28) involves a dynamics which relates a general activity *go* to the goal-state *be in a city where there is a concert of Baroque music*. The activity is general in the sense that no specific path to the goal has to be specified. The use of the present progressive means that

$$?HoldsAt(going, now) \text{ succeeds,}$$

where the fluent *going* is derived from the use of the progressive in (28).

In contrast, (29) involves three much more specific activities *going to Besançon* etc. which can be related to the former *going* by means of hierarchical planning as in Section 3 of Chapter 7. That is, we may have clauses

$$(\dagger)HoldsAt(going[x], t) \rightarrow HoldsAt(going, t),$$

for $x = b, m$ or p .

It follows using (\dagger) that if one of the queries $?HoldsAt(going[b], t)$ or $?HoldsAt(going[m], t)$ or $?HoldsAt(going[p], t)$ succeeds, $?HoldsAt(going, now)$ can be made to succeed as well. Similarly for the corresponding parametrized start-event $e[x]$, if either

$$?Happens(e[b], t), t < now$$

or

$$?Happens(e[m], t), t < now$$

or

$$?Happens(e[p], t), t < now$$

succeeds, then so does

$$?HoldsAt(going, now).$$

However, the latter query can be made to succeed without requiring success of the former queries if we do not avail ourselves of the clauses (\dagger) .

In terms of planning, using (\dagger) means that the planning has entered its final stage, where the destination has to be chosen. Conversely, not using (\dagger) means that the plan has not yet been decomposed into a more detailed plan. This would seem to be the case in the situation described by Bonomi. Therefore sentences (28) and (29) are logically unconnected in the situation described.

Now note that if the queries $?HoldsAt(going[x], now)$ (for $x = b, m$ or p) neither succeed nor fail, the truth of the sentences *Leo is going to Besançon* etc. is similarly undecided. In fact we would doubt that *Leo is going to Besançon* etc. are false, as Bonomi maintains. It is correct to say these sentences are not true, but to call them false would mean that the

corresponding queries fail. Since this is not the case, it seems more appropriate to assign an indeterminate truth values in these cases. The upshot of this discussion is that in the situation described, (28) is true, and (29) indeterminate.

5. Exercises

EXERCISE 13. *Prove corollary 4. (Hint: apply definition 40.)*

EXERCISE 14. *Discuss the difference between the sentences*

- (30) a. *#The coin is coming up heads.*
 b. *It is important for the catcher to know what the ball is going to do when it hits the fence.*

in view of the fact that both describe largely unpredictable physical phenomena.

EXERCISE 15. *Formalize Bonomi's 'Multiple Choice Paradox' and the proposed resolution.*

CHAPTER 11

Coercion

Aspectual properties or *Aktionsart* of verbs cannot be *lexical* properties of these verbs alone. For instance, *John drank* is certainly an activity and *John drank beer* is an activity as well. However, *John drank a glass of beer* is not an activity but an accomplishment. Therefore, the choice of the object NP (a mass term versus an indefinite NP) is a determining factor for assigning aspectual class. The next example from [109] shows that in general *Aktionsart* is not fully determined prior to the sentence level. The verb ‘arrive’ in

(1) Chapman arrived.

is an achievement, as can be seen from the ungrammaticality of

(2) *Chapman arrived all night.

But if we choose a bare plural subject instead of a proper name as subject NP, we get the grammatical sentence

(3) Visitors arrived all night.

In the last sentence, ‘arrive’ denotes an activity. Therefore the choice of the object and subject NPs can force verbs to be interpreted in different aspectual classes. However, in several quarters of the semantic world, Vendler’s classification of verbs into states, achievements, activities, accomplishments and points is often conceived of as pertaining to inherent properties of a verb (or, rather verb phrase). By contrast, various forms of cognitive grammar treat aspect as a way to *impose* temporal structure on events, or a way to conceptualize the temporal constitution of events. In the words of Croft [21, p. 70]:

The aspectual grammatical constructions determine in part the temporal structure of the event it describes via conceptualization.

The difference between these ways of treating aspect is subtle but important, as can be made clear by means of an example due to Croft (*op. cit.*). If the verb *love* is considered to be inherently stative, then

(4) *I am loving her

is ungrammatical, and cognitive linguists would agree; here the intended content has to be expressed by

(5) I love her.

But if (4) is provided with a context, it may become grammatical, as in

(6) I am loving her more and more every day, the more I get to know her.

This phenomenon is hard to explain if stativity is taken to be a property of the verb *love*, fixed in the lexicon. It seems more profitable to explain the contrast by saying that the events referred to in (5) and (6) are conceptualised differently, so that the event taken as an atom in (5) is considered to have stages and phases in (6). Thus, the phenomenon of *aspectual coercion* comes to the fore: the potential of grammatical constructions, such as the progressive, to ‘move’ a verb or verb phrase from one aspectual category to another.

The term *coercion* was introduced in [81]. Unfortunately, in the literature the term ‘coercion’ does not refer to a coherent class of phenomena. Consider example (7):

(7) Pollini played the sonata for two days.

Pollini played the sonata is an accomplishment and accomplishments are generally bad with *for*-adverbials; so one might think that (7) is not felicitous. Nevertheless sentence (7) can mean that Pollini played this sonata repeatedly within a timespan of two days. Thus the accomplishment *Pollini played the sonata* is coerced to an iterative reading. But here we cannot identify a linguistic constituent of the sentence which is responsible for this reinterpretation, as we did above in example (6). The expression *for two days* does not force an iterative reading in *John slept for two days* and neither does the accomplishment *Pollini played the sonata*.

Worse, even cases of metonymy are often considered as instances of coercion. Thus a sentence like *The pianists are on the top shelf* interpreted with respect to a scene within a CD-shop means that the CDs of pianists are on the top shelf. The NP *The pianists* is in this context reinterpreted as *CDs of pianists* on the basis of a threatening type conflict with the VP *be on the top shelf*.

In this chapter we will certainly not do justice to all phenomena which are grouped together under the linguistic term *coercion*, but we will concentrate on those which pertain to Vendler’s classification outlined above. Moreover, we will not comment on the vast literature on reinterpretation but refer the interested reader to a selection of more recent work¹.

Note that in general we cannot expect to arrive at the result of a coercive move in a compositional way². To see this, consider again example (7). As

¹See especially Moens [81], Steedman [109] and the unpublished [108], Pulman [92], Pustejovsky [93], Dolling [29], De Swart [26], and Egg [34].

²The reader will have noticed that we have paid scant attention to compositionality in this book. One reason is that it is impossible to do justice here to the recent wave of interest in compositionality inaugurated by Hodges [51]. Another reason is that the approach to

pointed out in the case of sentence (7), the linguistic material itself does not give any clue that an iterative reading is prominent here. We have to rely on extralinguistic knowledge such as the timespan it usually takes to play a sonata to derive this meaning. A theory of coercion effects should therefore allow to combine very different aspects of meaning within a unified formal framework.

We will now begin the technical development of this chapter. Here is a characteristic example. The default eventuality associated to, say, ‘reach the top’, is an achievement. As such, the application of the progressive is excluded, because the achievement lacks the dynamical component which is a prerequisite for the application of the progressive. Nevertheless, in some contexts the progressive is appropriate, as in

(8) They were reaching the top when a blizzard forced them to go back.

In this case the use of the progressive *coerces* the achievement ‘reach the top’ into an accomplishment. The purpose of the following Sections is to give a formal treatment of this operation, or rather, family of operations. We will distinguish three forms: additive coercion, subtractive coercion, and cross-coercion.

1. Additive coercion

This is the simplest kind of coercion, which consists in elaborating a scenario.

1.1. Activity \rightsquigarrow accomplishment. The verb ‘build’ is an activity, which is transformed into an accomplishment by adding a direct object such as ‘a house’. In the simple setup outlined above, the only property governing ‘build’ is a statement of the form ‘*Initiates(start, building, t)*’. The effect of adding the direct object *a house* is that this scenario is extended with a dynamics, which relates the activity *building* to the construction stages of the house, and a set of statements describing the behaviour of the culminating event. Hence the name ‘*additive coercion*’.

1.2. Achievement \rightsquigarrow accomplishment. This case is already slightly more complicated. Above we remarked that sentence (8) requires the transformation of ‘reach the top’ into an accomplishment. That is, since the progressive requires for its application the presence of a dynamics, this must

semantics chosen here goes counter to several formulations of compositionality such as for instance

- (i) The meaning of an expression should not depend on the context in which it occurs (Hintikka).
The semantic value of a constituent (phrase marker) does not depend on what it is embedded in (Higginbotham). It has been shown here that one can have a fully computational theory of meaning without resorting to such forms of compositionality.

first be added to the scenario. Now since an achievement does not contain an activity-component, the dynamics introduced into the scenario is of the following general form, with f_1 and f_2 still unknown *parameters*³:

- (1) $Initiates(e, f_1, t)$
- (2) $Releases(e, f_2, t)$
- (3) $HoldsAt(f_2(x), t) \rightarrow Trajectory(f_1, t, f_2(x'), d)$

and such that for a particular value c :

$$HoldsAt(f_1, t) \wedge HoldsAt(f_2(c), t) \rightarrow Happens(reach, t)$$

Furthermore, the use of the past progressive dictates that the query $?HoldsAt(f_1, t), t < now$ succeeds. The question is therefore, what values to substitute for the parameters. In this case, semantic memory will contain something like a clause of the form

$$HoldsAt(height(x), t) \rightarrow Trajectory(climb, t, height(x), d),$$

where *climb* is an activity and *height(x)* a parametrized state, representing the height gained during climbing. This then indicates that *climb* must be substituted for f_1 , and *height(x)* for f_2 , and we have the required addition to the scenario.

2. Subtractive coercion

We now discuss an instance of the converse transformation, in which parts of a scenario are deleted.

2.1. Accomplishment \rightsquigarrow activity. The most interesting case of this form of coercion occurs when we consider bare plurals: ‘drink a glass of wine’ is an accomplishment, but ‘drink wine’ is an activity (in the wide sense). What the (extended) activity and the accomplishment have in common is a dynamics, the crucial sentence of which has the following form

- (1) $HoldsAt(f_2(x), t) \rightarrow Trajectory(f_1, t, f_2(x + g(d)), d)$.

Here, f_1 is the activity, and f_2 is the changing partial object. Now activities and accomplishments are differentiated from each other by the fact that the scenario for an accomplishment requires sentences of the type

- (2) $Terminates(e(c), f_1, t)$
- (3) $HoldsAt(f_2(c), t) \wedge HoldsAt(f_1, t) \rightarrow Happens(e(c), t)$.

Here, $e(c)$ is a culminating event, dependent upon a constant c . An example of such a constant would be the quantity determined by a glass of wine; then $e(c)$ is the event ‘finish a glass of wine’. Looking at the last sentence, one sees that, in a minimal model, the culminating event will only be activated

³The difference between variables and parameters is that in a clause the former but not the latter must be read as universally quantified. An alternative way to represent the dynamics introduced by the progressive is via a suitable integrity constraint, but this is notationally somewhat inconvenient (see exercise 16).

if for some $x, d, x + g(d) = c$ can be derived, such that $Initially(f_2(x))$ is true.

Thus, the effect of ‘a glass of –’ would be to add (2) and (3) Conversely, taking away ‘a glass of –’ from ‘drink a glass of wine’ would lead to the deletion of sentences (2) and (3) Hence the name *subtractive* coercion.

3. Cross-coercion

We now discuss several difficult cases of coercion which cannot be viewed as adding or deleting sentences from a scenario. This kind of coercion occurs for instance when a state (which is of the form $(-, -, -, +)$) is coerced into an activity (of the form $(+, +, -, -)$) under the influence of the progressive. Of course a combination of the addition and subtraction operations would also yield this transformation, but presumably something else is going on.

3.1. State \rightsquigarrow activity. Croft [21] gives the following convincing examples:

- (9) She is resembling her mother more and more every day.
- (10) I am loving her more and more, the more I get to know her.

Consider example (9). Let f be a fluent denoting *resemble*, conceived of as a state. Then the sentence

- (11) She resembles her mother.

is represented as the integrity constraint ‘ $?HoldsAt(f, now)$ succeeds’.

The phrase ‘more and more’ introduces a new fluent $f'(x)$ denoting *resemblance to degree x* . Furthermore, the application of the progressive introduces a dynamics consisting of the general form

- (1) $Releases(e, f_2, t)$
- (2) $HoldsAt(f_2(x), t) \rightarrow Trajectory(f_1, t, f_2(x'), d)$

Here, the f_1, f_2, e are *parameters*⁴, which have to be unified with terms provided by the discourse, namely f and $f'(x)$, and a starting event e_0 which derives from the integrity constraint ‘ $?HoldsAt(f, now)$ succeeds’.

We claim that the observed coercion is due to the unification of the constant f with the parameter f_1 , thus forcing a state to become an activity. The use of the present progressive in (9) means that the query $?HoldsAt(f, now)$ must succeed. If we start a derivation with $?HoldsAt(f, now)$ as the top query, we soon see that parameters have to be replaced by constants in order for the derivation to proceed.

Suppose that f is substituted for f_2 in the *Releases* and *Trajectory* predicate, then $Clipped(s, f, r)$ will be true of f if $s < t_0 < r$. This means that the query $?¬Clipped(s, f, r)$ will fail for these s, r , and with this the query

⁴Here it is necessary to use the formal treatment of parametrized fluents as given in Section 2.0.1 of Chapter 6, since f_2 is a parameter standing for a parametrized fluent.

$?HoldsAt(f, now)$. By contrast if f is substituted for the first argument of the *Trajectory* predicate then $f'(x)$ can be substituted in the third argument and in *Releases* and $?HoldsAt(f, now)$ can be made to succeed by requiring that a *start* event occurs. This implies that the state *resemble* is coerced into an activity via unification. (See also exercise 17.)

It should be noted that some native speakers of English find this example more problematic than Croft appears to do. From the formal side we may observe that this example involves much more processing than all the previous examples; it is possible that this difference is related to the difference in acceptability judgments.

3.2. The structural versus phenomenal distinction. Comrie [17, p. 37] pointed to another case where the progressive can be used felicitously with a stative verb:

- (12) a. The Sphinx stands by the Nile.
b. Mr. Smith is standing by the Nile.

‘Stand’ is a stative verb, but the use of the progressive is allowed, even mandatory, in case (12-b).

This phenomenon was dubbed the ‘structural versus phenomenal’ distinction by Goldsmith and Woisetschlaeger [44, p. 84]. They pointed out that, contrary to what example (12) might suggest, temporal duration is not the issue per se:

- (13) a. The statue of Tom Paine stands at the corner of Kirkland and College (but everybody expects the new Administration to move it).
b. The statue of Tom Paine is standing at the corner of Kirkland and College (and nobody thinks the deadlocked City Council will ever find a proper place for it).

The durations need not differ in (13-a) and (13-b); what matters here is how the speaker construes the position of the statue: permanent, as in (13-a) (structural interpretation), or transitory, as in (13-b) (phenomenal interpretation). In the latter case use of the progressive is called for.

After the foregoing elaborate treatment of coercion from state to activity, the analysis of this phenomenon should cause no problem, so we will give an informal indication only. Consider example (12). The main point is this: whereas in example (13-a) ‘stand’ is coded by a parameter-free fluent f , in (13-b) it is coded by a parametrized fluent $f(x)$, representing the position of Mr. Smith. The fluent $f(x)$ may be constant for long stretches of time, but it needs an activity to drive it via a dynamics; this is precisely what the use of the progressive indicates. The analysis of (13) is left to the reader.

3.3. Point \rightsquigarrow activity. Here we must consider a transformation from a quadruple $(-, -, +, -)$ to $(+, -, -, -)$ (or perhaps $(+, +, -, -)$). Consider

- (14) a. The light flashed.
 b. The light was flashing all night.

‘Flash’ is a point; for this example we need therefore consider only the event *flash*. Sentence (14-a) can then be formalised as the integrity constraint

$$?Happens(\textit{flash}, t), t < \textit{now} \textit{ succeeds}.$$

In sentence (14-b) ‘flash’ occurs in the progressive, hence it must formally be represented by a fluent, or rather a pair of fluents – one an activity f_1 , the other a (possibly parametrized) state f_2 , related by a dynamics. At first f_1 and f_2 are parameters, which then have to be unified with material given by sentence (14-b). However, at first sight there seems to be *no* fluent with which the parameters can be unified, since the *lexical* material of (14-b) provides only an event type: the representation of the point ‘flash’. The coercion process requires therefore as a preliminary stage coercing an event type to a fluent. In [47] (see also Section 3.5 of Chapter 12 below) we have indicated (for the case of coerced nominals) a procedure which achieves just this. Namely, this form of coercion is represented by a mapping $\textit{flash} \mapsto Happens[\textit{flash}, \hat{t}]$, which maps an event type to a fluent⁵. By construction, in any given model the fluent has the same temporal profile as the event type, but it belongs to a different syntactic category and is now available for unification. We still need one to fill argument place however, because we have made an absolute syntactic distinction between activities and states in terms of the argument places that they can fill (see Chapter 7). A fluent may change syntactic category due to a change in the scenario, but we do not allow that a fluent represents simultaneously a state and an activity. Therefore $Happens[\textit{flash}, \hat{t}]$ unifies either with f_1 or with f_2 , but not with both. We then imagine that semantic memory is searched for an activity f , or rather a mechanism, which drives the flashing via dynamics (it is clear what this would be in the case of a lighthouse, say); and in the next step f is unified with f_1 and $Happens[\textit{flash}, \hat{t}]$ with f_2 . Sentence (14-b) is then represented by the integrity constraint⁶

$$?HoldsAt(f, s), s < \textit{now} \textit{ succeeds}.$$

3.4. Activity/accomplishment \rightsquigarrow state. To conclude the Section on cross-coercion we briefly discuss two examples about which much more could be said, in particular with regard to the syntax-semantics interface.

3.4.1. *Negation.* It has often been held (see for example Verkuijl [128]) that negating an activity results in a stative predicate. This intuition can to a certain extent be reproduced in the present framework. An activity (in the wide sense) consists of a fluent representing the activity proper, and a parametrized state. In a minimal model, the state changes only when the activity is ‘on’; outside of those intervals, the parametrized state is really

⁵At this point the material on coding in Chapter 6 is used essentially.

⁶Disregarding for the moment the temporal adverbial ‘all night’; these adverbials will be treated in Section 4.

static. It follows that the set of intervals complementary to that representing the activity indeed has a state-like character. It is a somewhat atypical state in that it is initiated and terminated by events, since of course a terminating event for an activity is an initiating event for its negation.

3.4.2. *Passive*. In German and Dutch there is a form of the passive, the ‘Zustandspassiv’ (indicated by a special auxiliary), which transforms an activity or an accomplishment into a (consequent) state.

- (15) a. Johann baut ein Haus. Es ist weiss.
 b. Das Haus ist von Johann gebaut. Es ist weiss.
 c. Das Haus wird von Johann gebaut. *Es ist weiss.

In (15-b), the auxiliary *sein* indicates that the house is finished; by contrast, in (15-c) the auxiliary *werden* refers to the process of building. This explains the * in (15-c).

In English the passive is apparently ambiguous between the two readings. Compare

- (16) a. John is building a shed in his garden. This causes his neighbours much distress [because of the tremendous noise].
 b. The shed that is built in John’s garden causes his neighbours much distress [because it spoils their view].
 c. A shed is built in John’s garden. This causes his neighbours much distress.

‘This’ in sentence (16-c) can refer both to an activity, as in (16-a), and the result of that activity, as in (16-b). If one conceives of the passive syntactically as movement of the object NP into subject position, an interesting tension between syntax and semantics comes to the fore. The NP *a house* in *build a house* is an incremental theme, and need not denote an object in the ontological sense. In our setup, the denotation of *a house* is as it were distributed over the changing partial object, the canonical terminating event, and the consequent state whose relations are governed by the scenario. Similarly, the verb *build* is not a two-place predicate, but a fluent with one (subject) parameter. So what happens semantically when the NP is moved into subject position?

One possibility is that nothing happens, in which case a passive sentence retains the process reading of the corresponding active sentence. Another possibility is that the NP is re-interpreted as a real object, to which an adjective can be applied. Indeed, in English grammar this form of the passive is known as ‘adjectival passive’. For example, *built* would now be an adjective, obtained from the verb *build* by existentially quantifying the subject position. But the upshot of this is that ‘a house is built’ now corresponds to the consequent state of the accomplishment, as indicated in Section 2.1.

4. Temporal adverbials: 'in' and 'for'

The adverbials 'in' and 'for' can to some extent be used as a test to differentiate between *Aktionsarten*. Accomplishments and achievements are generally bad with 'for', but good with 'in', whereas activities and states exhibit the opposite pattern. Even so, the computational import of 'in' as applied to an accomplishment differs from that as applied to an achievement. In the former case, one measures the length of the interval between start and culminating event, i.e. the duration of the activity implied by the accomplishment. In the latter case, 'in' is measured in the same way except that the activity now does not form part of the achievement. This use of 'in' is therefore an example of coercion, albeit a mild one. Furthermore, sometimes it is possible to use 'for' with an accomplishment as in

(17) Pollini played Opus 111 for two weeks.

which now means something like 'Pollini played Opus 111 on most evenings during a two-week tour', because we know that Opus 111 by itself takes about 25 minutes. Here we see that 'for' coerces the accomplishment to an iterative reading. Starting from the other end, we see that the use of 'for' with a state implies a mild form of coercion, introducing initiating and terminating events:

(18) He was CEO for eight years (when he was unceremoniously dumped).

The use of 'in' may coerce a state into an achievement, as in

(19) He was CEO in two years.

Similarly, an activity may be coerced into an accomplishment by means of 'in', as we can see from

(20) John sang in an hour.

This example needs some context however: one can imagine a singing class where each pupil has to sing *Die schöne Magelone*; (20) might then be a comment. But given the appropriate context, (20) is fine.

Our first task is to provide a formal meaning for 'in' and 'for'; we will then be in a position to formalize some of the coercion processes involving 'in' and 'for'.

As semantic representations of 'in' and 'for', we will augment the event calculus with primitive predicates $In(e, x)$ and $For(e, x)$, whose meaning will be defined by suitable logic programs. In both cases, e is an event type (but see below), and x is a real number representing an amount of time. The event type is to be derived from a given *Aktionsart* by means of hierarchical planning, as in Section 3 of Chapter 7. The hypothesis advocated here is that, by default, use of 'in' requires an accomplishment, and use of 'for'

an activity. If either is applied to a different *Aktionsart* missing bits in the scenario are filled in by means of coercion.

In order to give the formal definitions, we need the scenario for a ‘clock’ introduced in Section 2.4 of Chapter 7 and reformulated here for convenience. Our purpose is to measure the duration of an activity fluent f between the two designated events $start_f$ and $finish_f$. To this end, the scenario introduces two fluents, the activity-fluent $clock_f$, and the parametrized fluent $time_f(x)$ (where x ranges over \mathbb{R}) driven by the former via a dynamics.

DEFINITION 41. *The scenario for measuring the duration of the fluent f comprises the following statements.*

- (1) *Initially*($time_f(0)$)
- (2) *Releases*($start_f, time_f(0), t$)
- (3) *Initiates*($start_f, f, t$)
- (4) *Initiates*($start_f, clock_f, t$)
- (5) *HoldsAt*($time_f(x), t$) \rightarrow *Trajectory*($clock_f, t, time_f(x + d), d$)
- (6) *Terminates*($finish_f, f, t$)
- (7) *Terminates*($finish_f, clock_f, t$)

It should be noted that the clock stops ticking only after the designated culminating event $finish_f$ has been reached. We have said nothing about other terminating events, and it may very well be that f is stopped and resumed several times before $finish_f$ occurs. The clock keeps ticking during those interruptions. This observation will be of some importance below.

Suppose we are given an accomplishment with activity fluent f and culminating event $finish_f$. Let the event type e be derived from the accomplishment by means of hierarchical planning as in Section 3 of Chapter 7.

DEFINITION 42. *The predicate $In(e, x)$ is defined by the logic program*

- (5) $Happens(finish_f, t) \wedge HoldsAt(time_f(x), t) \rightarrow In(e, x)$

Furthermore, this is defined for activity fluents f only.

This definition (or rather its completion) has the effect of making $In(e, x)$ false for all x if $finish_f$ does not occur. Note that In measures the time between concrete $start_f$ and $finish_f$ events, interruptions included. Since In is therefore relative to particular initiating and terminating events, it is not entirely correct to make In depend explicitly on an event type only, where it actually should depend on an event token. In Chapter 12, particularly Section 4, we give a more elaborate discussion of the notion of event token for temporally extended events.

The program clause defining ‘for’ is based upon the same idea, but is somewhat simpler, since *For* can be made to apply directly to a fluent. The main difference between In and *For* is that the latter expression does not refer to the canonical terminating event $finish_f$, but has to be defined relative to a contextually determined terminating event, say *stop*; and furthermore that the fluent f can be a state as well.

DEFINITION 43. *The predicate $For(f, x)$ is defined by the logic program*

$$(6) \quad Happens(stop, t) \wedge HoldsAt(time_f(x), t) \rightarrow For(f, x)$$

Upon completion, this definition again has the effect of making $For(f, x)$ false for all x if $stop$ does not occur. Again, what is measured is the time elapsed between $start_f$ and $stop$; other terminating events may have occurred in the meantime. As with the definition of In , the notation $For(f, x)$ is sloppy in that it omits reference to the starting and stopping events.

Given these definitions, one may translate sentences involving the default use of 'in' and 'for' in the following way.

- (21) a. He crossed the street in two minutes.
 b. The query $?Happens(e, t), t \geq now$ fails, and $In(e, 2mn)$.

Here e is the event derived from the accomplishment 'cross the street' via hierarchical planning. Similarly for the activity 'write'

- (22) a. He was writing for an hour.
 b. The query $?HoldsAt(writing, t), t < now$ succeeds, and $For(writing, 1h)$.

Given the different arguments that In and For take, we can see immediately why the following sentences are not felicitous

- (23) a. *He wrote a letter for an hour.
 b. *He was writing a letter in an hour.

These sentences are however infelicitous for different reasons and to different degrees. The first sentence requires identification of the canonical culminating event $finish$ with a terminating event $stop$ as required by 'for'. This sentence seems to be infelicitous because a stronger statement is available. The second sentence requires the occurrence of the canonical culminating event whereas the progressive posits this occurrence only by default. This sentence therefore errs on the other side, by pretending to give more information than is available.

We now turn to coercion. Consider

- (24) Bill reached the top in five hours.

The adjunct 'in five hours' introduces its defining condition into the scenario, which, via the definition of the parametrized fluent $time(x)$, contains a reference to an activity leading up to the culminating event (cf. definition 41). The achievement 'reach the top' is therefore coerced into an accomplishment. Roughly the same process is at work in the case of

- (25) He was CEO in two years.

We saw the same process at work in Chapter 9, with the example 'Il fut président'. Being CEO is a state; but the adjunct 'in two years' now introduces both an activity fluent and a culminating event into the scenario;

the fluent ‘be CEO’ is unified with the state consequent upon the culminating event, which then functions as an event marking the beginning of being CEO.

The previous two examples are instances of what we called additive coercion. A really difficult case of coercion is sentence (17), where an accomplishment is coerced to an iterated activity. Here is an informal description of the computation involved. Since ‘for’ does not go with an accomplishment, it strips off all components of the quadruple except the first (activity) fluent. Unification of the event postulated in the definition of ‘for’ with the culminating event of the accomplishment is impossible because of the difference between 25 minutes and two weeks. We are left with ‘Pollini played for two weeks’ together with a scenario that says that playing is an activity fluent which drives the parametrized partial object fluent *Opus 111*. As remarked above, ‘Pollini played for two weeks’ does not imply that Pollini played continuously for two weeks, only that there is a time span of two weeks between two designated events *start* and *end*. Thus there may be several occasions of playing, and the scenario for playing (a concert) will ensure that these occasions are long enough to complete *Opus 111*. In this sense we get the iterated activity reading. However, the computation only shows that there must be two occasions of playing, one following *start* and one preceding *end*. The iterativity expressed by (17) is not captured thereby. The root of the trouble is that the definition of ‘for’ allows too much leeway: the interruptions may be out of all proportion to the actual activity. Since the result of the above computation therefore leaves something to be desired, we refrain from formalizing it here.

5. Coercion and intensionality

We close this chapter with some remarks of a philosophical nature. Frege introduced two concepts which are central to modern formal approaches to natural language semantics; i.e. the notion of *reference* (denotation, extension, Bedeutung) and *sense* (intension, Sinn) of proper names⁷. The sense of a proper name is *wherein the mode of presentation (of the denotation) is contained*; Frege speaks of *Art des Gegebenseins*. For Frege proper names include not only expressions such as *Peter*, *Shakespeare* but also definite descriptions like *the point of interSection of line l_1 and l_2* and furthermore sentences which are names for truth values. Sentences *denote* the *True* or the *False*; the *sense* of a sentence is the proposition (Gedanke) the sentence expresses. In the tradition of possible world semantics the proposition expressed by a sentence is modelled via the set of worlds in which the sentence is true. This strategy leads to well known problems with propositional attitudes and other intensional constructions in natural languages since it predicts for example that the sentences in (26) are equivalent.

⁷See especially [41] and the English translation [40].

- (26) a. Jacob knows that the square root of four equals two.
 b. Jacob knows that any group G is isomorphic to a transformation group.

Even an example as simple as (26) shows that the standard concept of proposition in possible world semantics is not a faithful reconstruction of Frege's notion of *sense* and *Art des Gegebenseins*. Frege developed his notion of sense for two related but conceptually different reasons. We already introduced the first one by considering propositional attitudes. The problem here is how to develop a general concept which can handle the semantics of Frege's *ungerade Rede*. The second problem is how to distinguish a statement like $a = a$ which is rather uninformative from the informative statement $a = b$; that is, how to account for the semantic difference between (27-a) and (27-b).

- (27) a. Scott is Scott.
 b. Scott is the author of *Waverly*.

Frege's intuitive concept of sense therefore was meant both to model information content and to provide denotations for intensional constructions. We adopt Frege's distinction between *sense* and *reference* with a decidedly psychological interpretation. That is, *Art des Gegebenseins* is taken to refer to the way an aspect of reality is structured by our cognitive system. Take as an example the fundamental notion of an 'event'. As we have seen, Vendler's classification of verbs into states, achievements, activities and accomplishments is often conceived of as pertaining to inherent properties of a verb (or rather, verb phrase). By contrast, this book, and various forms of cognitive grammar, treat aspect as a way to *impose* temporal structure on events, or as a way to conceptualize the temporal constitution of events. This means that an aspectual construction (such as the progressive) can sometimes override the Aktionsart of a verb. In such a view, *sense* takes primacy over *reference*. Hesperus and Phosphorus refer to the same object in the real world, but the names are different because that same object is given in two different ways. In the case of aspect, the distinction between sense and reference is radicalized in that, for instance, events have no canonical referents in the world; rather the reference is constructed from a sense. Evidence is provided by the process of coercion. The expression 'reach the top' is an achievement, and highlights a particular event⁸ and its aftermath. However, when used in combination with the progressive, the culminating event loses its salience and what is profiled instead is the run-up activity leading up to the culminating event. Native speakers of English do this automatically, and consistently. This suggests an algorithmic interpretation process whereby the denotation of an expression is constantly re-computed on the basis of incoming data (linguistic or otherwise) which modify the sense. Indeed, coercion may provide us with a clue to a more adequate formal treatment

⁸Which we can think of as being punctual in the sense of not having internal structure.

of intensionality, which goes back to Frege's informal description of sense as *Art des Gegebenseins*.

Moschovakis (see [82] and especially the recent [84]) develops a computational analysis of sense and reference which is certainly closer to Frege's intentions than is possible world semantics. Moschovakis' motivations are twofold. The first motivation is to give a rigorous definition of the concept *algorithm* (see for instance [83]) and thereby provide the basics for a mathematical theory of algorithms. The second motivation is of a philosophical and linguistic nature. It consists in providing a more adequate formal reconstruction of the Fregean notions *sense* and *reference* via the formalized concepts *algorithm* and *value of an algorithm*. In a nutshell, the idea is this:

The sense of an expression is the⁹ algorithm which computes its reference.

This idea fits rather well with the approach to semantics advocated here, if we read for 'algorithm': 'scenario' (i.e. a particular kind of constraint logic program) and for 'value of an algorithm': 'minimal model of the scenario' (which, as we have seen, can be computed from the scenario). That is, to each expression a scenario is associated as its sense, and resolution allows one to compute the reference of the expression in the minimal model¹⁰. The preceding formulation is not entirely correct however. We have seen repeatedly that the meaning of expressions is not given locally, as a fixed association between meanings and expressions. Rather, meanings are associated to finite sets of expressions by means of scenarios, which are the logical analogues of semantic networks. 'Sense' is therefore not attributed to single expressions, but to finite sets of expressions; and even then it is not a *property* of a given finite set of expressions, because that same set may be involved in several different scenarios. That is, the same set may have many different senses. Coercion plays an important role in navigating between these senses, as we have seen in great detail.

6. Exercises

EXERCISE 16. *Represent the introduction of a dynamics due to the occurrence of the progressive by means of an integrity constraint.*

EXERCISE 17. *Provide the formal details for the coercion involved in examples (9) and (10).*

EXERCISE 18. *Give a formal argument showing how the present progressive may be coerced into a futurate progressive (cf. Section 5.4).*

EXERCISE 19. *Formalize the coercion process going on in sentences (24) and (25).*

⁹In order to make 'the' meaningful one needs a normal form theorem for algorithms; see [83].

¹⁰This idea was first suggested in [125].

EXERCISE 20. *Discuss the meaning and relevance of the compositionality principle in view of coercion phenomena.*

CHAPTER 12

Nominalization

The ontology of the event calculus comprises both event types and fluents. This distinction proved to be very useful in formalizing the principle of inertia, because it allowed us to state formally that a property will continue to hold unless terminated by the occurrence of an event. The present chapter approaches this distinction from another angle by showing that it is ingrained in language. Nominalization is the process which turns verb phrases into nouns denoting eventualities. Interestingly, however, these nouns come in two forms, and if one looks carefully at the semantic properties of these forms it becomes apparent that they must denote event types and fluents as defined axiomatically by the event calculus. That is, the axioms of the event calculus are necessary to ensure that the two forms have the right logical properties.

1. Two types of English Gerunds

In English, nominalization either produces a so-called *derived nominal* (say of the kind ‘evolve – evolution’) or it involves the suffix *-ing*, whence the term ‘gerund’ for this case. It has been known for a long time¹ that English has at least two different types of gerunds: one more nominal in character, and the other one more verbal. An example for the nominal type is (1), and (2) exemplifies the verbal type.

- (1) beautiful singing of Cenerentola
- (2) singing Cenerentola beautifully

These two forms of gerunds differ both in the verbal contexts in which they can occur and in their internal syntactic structure. We now take up these points in turn.

1.1. Verbal Context. It was Zeno Vendler in [126] and [127] who pointed out that the different types of gerunds must actually denote in different categories, by investigating the verbal contexts which allow one or the other of the gerund types.

The general form of his argument can be illustrated by considering the two statements in (3):

- (3) a. John’s speech took place yesterday.

¹See for instance Poutsma [90] and Jespersen [56]

- b. John's speech was inconsistent.

Since the denotations of 'be inconsistent' and 'take place yesterday' are disjoint, it follows that the expression *John's speech* cannot denote a single entity. On the contrary, it must denote both an *event* which is characterized by John speaking and the *content* of John's speech.

According to Vendler something similar must hold with regard to gerunds. The two types of gerunds denote in different categories: the nominal gerund denotes in the category of events, while the verbal gerund denotes in what Vendler calls 'the category of facts, results or propositions'. However, in this case the data are much more complex than the simple example (3) above suggests. This complexity is due to two possible verbal contexts for gerunds, which were called 'loose' and 'narrow' by Vendler, and which are used to establish the claim that the respective gerunds must denote in different categories.

1.1.1. *Loose containers*. Some verbal contexts accept both kinds of gerunds as arguments; these were called *loose* containers by Vendler. Expressions like *surprised us*, *is unlikely*, *is improving* are examples of loose containers.

- (4)
- a. The beautiful singing of the aria surprised us.
 - b. John's not revealing the secret is unlikely.
 - c. The singing of the song is fun.
 - d. John's quickly cooking the dinner surprised us.
 - e. (Jespersen [56, p. 327]) They were surprised by the sudden coming in of a stranger.
 - f. They were surprised by a stranger coming in suddenly.
 - g. (Google) The band's playing of the song is improving and there is some very interesting playing in thirds that we could never quite work out.

For instance, (4-d) shows that *surprise* can take verbal gerunds, while (4-e) shows that it can take nominal gerunds as well.

1.1.2. *Narrow containers*. We next consider the other type of verbal context, dubbed *narrow* containers by Vendler. For instance, the VP *took place yesterday* is a verbal context which accepts nominal gerunds only. More examples of narrow containers are shown in (5).

- (5)
- a. *The soprano's singing the aria was slow.
 - b. The soprano's singing of the aria was slow.
 - c. John's revealing of the secret occurred at midnight.
 - d. *John's revealing the secret occurred at midnight.
 - e. *John's not revealing the secret occurred at midnight.
 - f. (Google) The video and the band's playing of the school's alma mater [*sic*] capped the evening.
 - g. (Google) Clearly the Passover slaying of Egypt's firstborn occurred at midnight on the 15th of Nisan.

- h. (Google) The contract provides that *the* transfer of the assets and *undertaking of the business* is deemed to have occurred at midnight on 31 August.

Since expressions like *took place yesterday, was slow, occurred at midnight* are naturally interpreted as predicates of *events*², Vendler concluded that nominal gerunds denote events, whereas verbal gerunds do not. Verbal gerunds denote in a different category, the category of facts, results, and propositions. Hence there is a kind of type mismatch. Vendler argues that a type mismatch indeed explains the acceptability pattern shown in (6).

- (6) a. The physician's revealing of the secret took place yesterday.
b. *The physician's revealing the secret took place yesterday.

1.1.3. *Coercion?* We have seen that there is an asymmetry in verbal contexts: loose containers take both kind of gerunds, whereas narrow containers take only nominal gerunds. Why is the nominal gerund acceptable as an argument of loose containers, as in (7)?

- (7) a. The physician's revealing of the secret is impossible.
b. The physician's revealing of the secret took place yesterday.

Again, since 'be impossible' and 'take place yesterday' are disjoint, the expression *the physician's revealing of the secret* cannot denote a single entity, irrespective of context³. We then have two possibilities to explain the behavior of nominal gerunds.

The first possibility is that the nominal gerund is always ambiguous between an event reading and a fact reading. The paraphrase (8-b) of *The president's revealing of a state secret* in (8-a) suggests that this is indeed sometimes the case.

- (8) a. The president's revealing of a state secret was a surprise.
b. That the president revealed a state secret was a surprise.

In certain contexts *that*-clauses may well be taken to denote facts, whence we get the ambiguity between event reading and fact reading. This strategy thus treats expressions like *The president's revealing of a state secret* in analogy with those of the form *John's speech* in (3).

There are however clear differences between the expressions *John's speech* and *The president's revealing of a state secret*. For instance there is no direct link between the two meanings of *John's speech* (event and content), whereas the paraphrase of *The president's revealing of a state secret* in (8-b) expresses the fact that an event of the type described by the

²When we turn to formalizing these notions, we will say rather that narrow containers turn an event *type* into an event *token*.

³Note that (7) cannot be reproduced using verbal instead of nominal gerunds, since in this case (7-b) would become *The physician's revealing the secret took place yesterday*, which is ungrammatical.

NP occurred. Therefore there is a systematic link between the two postulated meanings of *The president's revealing of a state secret*. Moreover the paraphrase usually adds information the NP by itself does not give, for instance temporal information via the past form of the verb *reveal*. A second, more appealing, option therefore is to assume that, in the context of a loose container, an eventive NP is reinterpreted as a factive NP, that is, as the statement that the event occurs, or has occurred, etc.. We think that this thesis is both closer to the facts and theoretically more challenging⁴.

Having introduced loose and narrow containers, we must note that narrow containers form only a very small class of verbal contexts, including, apart from those mentioned above, *begin, end, last*. This list is not complete⁵, but the number of loose containers is certainly much higher. Also the frequency of *of*-gerunds seems to be going down, although there still are clear examples like the following ones we found with Google.

- (9) Last, but not least, was the bass Pawel Izdebski, whose *beautiful singing of Ferrando's big aria* at the beginning of the concert made the audience prick up its ears.
- (10) She began in a dreamlike voice So war mein Mutter (which translates, "that's how my mother was"), and later *the actual singing of the aria begins*.

There may be several reasons for the current low frequency of *of*-gerunds. One is, presumably, that there are many derived nominals which express events, such as *detonation, blizzard, tempest, accident, refusal, performance, explosion*⁶. In addition, the coerced use of the eventive gerunds can be taken over by the verbal gerund, leaving the eventive gerund to occur only in the (relatively small number of) narrow containers.

To conclude this Section on verbal contexts, we discuss what may be termed logical aspects.

⁴Incidentally, as noted by Vendler, coercion triggered by verbal context occurs with lexical nominals like *snowman* or *Yeti* too. This is clearly shown by the following pair of sentences:

- (i) a. The Yeti is a fact.
b. The Yeti lives in caves.

Example (i-a) is interpreted as claiming that the existence of the abominable snowman is a fact. Certainly *the existence of the Yeti* is not one of the lexical meanings of *Yeti*. No such reinterpretation is necessary for (i-b). Cf. also the following example from Google:

- (ii) A literal translation of the German title is "Yeti - Legend and Reality," and that is what Messner reports on.

⁵In the above examples, we also encountered *cap the evening*.

⁶Of course expressions like *exploding* exist, but they are often used as attributive participles.

1.1.4. *Intensionality*. We first observe that an important difference between narrow and loose containers concerns intensionality. Narrow containers determine strictly extensional contexts, in contrast to loose containers which are usually highly intensional, as the following examples (adapted from Parsons [87]) clearly demonstrate. Assume that it is true that James Bond is the most famous spy. Then clearly if (11-a) is true then (11-b) is true too.

- (11) a. The beheading of the most famous spy took place yesterday.
b. The beheading of James Bond took place yesterday.

But by no means can we conclude from the truth of (12-a) that (12-b) holds as well.

- (12) a. The beheading of the most famous spy surprised us.
b. The beheading of James Bond surprised us.

1.1.5. *Negation of containers*. The following examples (not due to Vendler) are concerned with the interaction of verbal containers with negation. They will be important in what follows for the motivation of denotation types for the gerunds and their respective verbal contexts. Narrow containers can be negated, and they stay narrow under negation, as the following examples demonstrate.

- (13) a. The singing of the song didn't occur at noon.
b. (Google) The End of the World didn't occur at midnight, December 31 1999.
c. *John's kicking the cat didn't occur at noon.

1.2. Internal structure. The internal structure of nominal gerunds and verbal gerunds is rather different. The nominal gerund resembles a "real" English relational noun like *father of Mary*, whereas the other type of gerund exhibits the internal structure of an English VP or sentence. This has been well known for a long time, as many examples from the work of Poutsma and Jespersen will show.

1.2.1. *Perfect nominals*. The nominal gerunds form part of a class of nominals Vendler calls *perfect*, for reasons which will become clear shortly. This is a very small, homogeneous class which, apart from the gerunds, contains derived nominals expressing events. The verbal gerunds are part of a class of nominals Vendler calls *imperfect*. This class is huge and extremely heterogeneous.

The difference between perfect and imperfect nominals, and their most important properties are illustrated in (14) and (15). Perfect nominals, like those in (14), occur with determiners, can be modified by adjectives but not by adverbs, and cannot appear with grammatical aspect, and they cannot be modalized. Further, it is impossible to negate perfect nominals (with one

possible exception to be discussed below). To summarize, perfect nominals are nominalized forms which have lost their verbal characteristics and behave like “real” nouns. This is why Vendler dubbed them “perfect”⁷.

- (14)
- a. The singing of the song
 - b. The saving of us (Stevenson)
 - c. (Google) Deidre Haren begins the play with her beautiful singing of ‘A Poultry Tale’
 - d. stunningly beautiful singing of Cenerentola.
 - e. (Google) the Passover slaying of Egypt’s firstborn
 - f. *the Passover slaying Egypt’s firstborn
 - g. On account of his deliberate buying up of stocks
 - h. *quickly cooking of the dinner.
 - i. *having cooked of the dinner.
 - j. *being able to cook of the dinner.
 - k. *not revealing of the secret.
 - l. (Google) It may be more difficult to imagine the aria’s place in the drama or story of the whole opera when listening to *a recording of just that aria*.

1.2.2. *Imperfect nominals*. Imperfect nominals show the opposite behavior, as the examples in (15) demonstrate. They cannot occur with nominal determiners, they can be modified by adverbs⁸ but not by adjectives, they can occur with at least grammatical aspect (the present perfect), they can be modalized, and it is possible to negate them.

- (15)
- a. *The singing the song.
 - b. *beautiful singing the song.
 - c. (Google) He also plays Johnny Seoighe after *singing the song beautifully*.
 - d. quickly cooking the dinner.
 - e. (Jespersen [56, p. 322]) On account of deliberately buying up stocks
 - f. (Google) Mordechai Vannunu has spent the best part of the last fifteen years in solitary confinement in a cell in the desert for *having revealed the ‘secret’* of Israel’s ‘Jericho’ missiles.
 - g. (Google) Lisa gets Martin tucked into bed. Martin tells her he is sorry for *not being able to cook the dinner* he had planned for her.
 - h. (Google) ... *not revealing the secret* when you use it in any transform is a rather fundamental and well-known principle [in cryptography].

⁷Vendler’s use of the term “perfect nominal” is not completely precise. For instance it is not always clear whether the nominal *beautiful singing of the song* or the full NP *the beautiful singing of the song* is the perfect nominal. Often he refers to phrases like *beautiful singing of the song* as subjectless nominals, [126], p. 130. We shall henceforth use the term “perfect nominal” both for the respective nominal and for the NP which contains a perfect nominal.

⁸They therefore can occur with adverbial determiners like *always*.

Imperfect nominals can occur externally in noun phrase positions such as subject (in (15-h)) or as the object of a preposition (in (15-c), (15-f) and (15-g)), but as we have seen their internal structure strongly resembles the structure of the *VP* or the *S* they are derived from. This is, of course, the reason why Vendler called them “imperfect”.

Here are some more examples found on the web.

- (16) Stradivari died without *having revealed the secret* of his famous varnish that made his violins sing.
- (17) The Catholic Church has been criticised for *not revealing the extent* of its possible involvement or complicity in the Holocaust,
...
- (18) A person commits the offense of *using or revealing a trade secret* ...
- (19) *The physician's not only revealing the secret to the father, but perceiving it to be harmless* ...

The last two examples show that Boolean combinations (*and, or, not*) of imperfect nominals are acceptable. In contrast, although perfect nominals are closed under conjunction or disjunction (cf. example (5-h)), they are not closed under negation.

1.2.3. *Possessives*. Another difference in internal structure between perfect and imperfect nominals concerns the role of possessives. Vendler [127] demonstrates that the genitive in verbal gerunds is not a “real” genitive like *John's* in *John's house*. This is shown by the following examples:

- (20) a. John's house
b. The house of John
c. John's singing the song
d. *The singing the song of (by) John

Example (20-b) is a paraphrase of (20-a), but if we try to construct a paraphrase of (20-c) along these lines, we end up with the ungrammatical (20-d).

Compared with the genitive in imperfect nominals, the genitive of perfect nominals behaves like a “real” genitive. This is shown for instance by the following observation: it is possible to delete the genitive of embedded imperfect nominals if it is coreferential with the matrix subject. Deletion in the case of perfect nominals however leads to ungrammaticality⁹.

- (21) (Vendler [127, p. 50])
a. He shocked us by (his) telling a dirty joke.

⁹A related observation is made by Jespersen, who notes that the genitive and common case are not clearly distinguished in verbal gerunds.

With regard to the occurrence of genitives before a gerund it may be remarked that it is sometimes doubtful whether we have a genitive or a common case.

Jespersen [56, p. 324]

- b. He entertained us by *(his) singing of arias. (50)

There are even cases where the genitive is not possible at all. Jespersen gives the following example:

- (22) He insists on no one/*no one's knowing about the experiment.

The following examples would be rather bad with the genitive too.

- (23) a. They objected to Tom*(')s getting nothing and John*(')s everything.
 b. We speak of good people*(')s going to heaven, and wicked people*(')s to the Devil. (Defoe)
 Jespersen [56, p. 326]

We conclude from these considerations that there is no significant difference in meaning between verbal gerunds with genitive case and those with common case.

1.2.4. *Negation of nominals.* The data presented so far look relatively neat: we found evidence for two kinds of verbal containers and two kinds of nominals. The following examples, due to Cresswell [20] show that there is as yet no reason for complacency.

- (24) a. The arrival of the train surprised us.
 b. The non-arrival of the train surprised us.
 c. The arrival of the train occurred at noon.
 d. *The non-arrival of the train occurred at noon.
 e. The unexpected non-arrival of the train
 f. *The non-arrival of the train unexpectedly

The nominal *arrival of the train* seems a clear-cut example of a perfect nominal; but what are we to make of *non-arrival of the train*? It exhibits the internal structure of a perfect nominal and seems to share some external distribution properties of imperfect nominals (24-d). It is however not an imperfect nominal in Vendler's sense, because it can occur with nominal determiners (24-b) and adjectives (24-e) but not with adverbs (24-f).

Here are some examples found with Google, to show that Cresswell's examples are by no means contrived:

- (25) a. Second, there was no directive to report *the non-arrival of a combatant ship* [from a story about USS *Indianapolis*, torpedoed by a Japanese submarine]
 b. The non-departure of a boat or plane.

We will have much more to say about the meaning of negation in examples such as this; for now it suffices to say that negation here seems to be interpreted as producing an antonym¹⁰.

¹⁰Of course, examples with negation interpreted antonymically are quite common in natural languages. The following one is a quote from Davies' *The Isles*:

Here are two more examples where the negation in a gerund is interpreted antonymically (the first is due to R. Cooper, the second was found with Google).

- (26) a. Andrew's not stopping for the traffic light.
 b. "Vehicles not stopping for pedestrians in crosswalks is the number one complaint we receive regarding traffic safety" said Lieutenant Mark Gover.

It is not clear whether these constructions are allowed to appear as arguments of narrow container. Native speakers' judgments differ when asked about the acceptability of the following example:

- (27) ?Andrew's not stopping for the traffic light took place at noon.

If (27) is acceptable¹¹, then the statement that perfect nominals cannot be negated would have to be modified to: perfect nominals can be negated by antonymic negation, but not by classical negation. This would also explain the observed internal distribution. We shall return to this matter below.

1.2.5. *Nominals and determiners.* Verbal gerunds cannot occur with nominal determiners. In English (although not in German), they share this property with proper names since constructions like *the Peter*, *a Mary* are not acceptable. Schachter [99, p. 215] was apparently the first to suggest that some gerunds – his gerundive nominals – behave like names. This claim is further supported by the following observation due to Pullum [91]:

- (28) a. *his leaving her that you predicted.
 b. his revealing of the secret that you predicted.

That is, verbal gerunds are like proper names in that they do not tolerate restrictive relative clauses – in contrast to nominal gerunds.

1.2.6. *Pluralized nominals.* One further observation supporting Schachter's proposal is that Ing-of nominals can sometimes be pluralized but verbal gerunds definitely cannot. Example (29-a) is from Poutsma.

- (29) a. He ignored the sayings and doings of the ladies of his family [90, p. 113].
 b. blessings of the children.
 c. *blessings the children.

The plural form is in fact quite common with the following nominals: *drawings*, *savings* and *blessings*.

- (i) Such ideas are non-disprovable.
 [24], p. 132

The author clearly does not want to state that such ideas are provable.

¹¹Some of our informants think that (27) can be made acceptable with sufficient context. For instance, if Andrew's reckless driving caused an accident leading to a court case, then (27) could be uttered in court.

1.2.7. *Ellipsis*. Observations from Abney [1, p. 244] show that perfect and imperfect nominals also differ in their ability to participate in N-bar deletion. For instance, an ellipsis with an imperfect construction as in (30)(a) is bad, while it is possible with an ing-of gerund and a narrow container as is shown in (30)(b).

- (30) a. *John's fixing the sink was surprising, and Bill's was more so.
 b. John's fixing of the sink was skillful, and Bill's was more so.

Abney claims that the gerund *John's fixing of the sink* is ambiguous and can either refer to the manner in which John fixed the sink - called the Act-reading by Abney - or the fact that John fixed the sink (Fact-reading). N-bar deletion is only possible under the Act-reading.

- (31) a. John's fixing of the sink was skillful, and Bill's was more so.
 b. *John's fixing of the sink was surprising, and Bill's was more so.

Note that this is actually another empirical argument in favor of the coercion approach to the behavior of perfect nominals occurring as arguments of loose containers. If *John's fixing of the sink* were simply ambiguous an additional argument would be required to rule out (31-b). We thus do not agree with Abney's view that this nominal is ambiguous.

Unlike nominal gerunds, verbal gerunds do not have Act-readings. According to Vendler this is the reason why they cannot occur as arguments of narrow containers.

- (32) a. *John's fixing the sink was skillful.
 b. John's fixing the sink was surprising.

The contrast in (30) is now explained, because (30)(a) allows only a Fact-reading, but the gerund in (30)(b) has the required Act-reading. Therefore, Vendler's category distinction is also useful for offering an explanation for certain types of ellipsis.

1.2.8. *Iterated nominalization*. Finally we note the following examples of iterated nominalizations, a phenomenon not discussed by Vendler.

- (33) a. John's supporting his son's not going to church
 b. John's improving his singing
 c. John's watching the dog's playing
 d. My discovering her not leaving
 e. his discussion of John's revealing the secret

First observe that all examples are factive in the following sense: the phrases presuppose that the fact expressed by the embedded nominal is in fact true. For instance (33-a) presupposes that John's son is not going to church. Further the negation in *not going to church* clearly has antonymic force. It

means that John's son refrains from going to church. Such iterated constructions do occur in actual language use, as the following examples found with Google show:

- (34) a. ...the speeding up of the building of the houses ...
 b. ...speeding up the building of new ontologies ...
 c. This was the first I knew of *his objecting to my going to Nashville*.

1.3. Syntax. Although the syntax of gerunds is not treated in this book¹², we want at least mention the most comprehensive work on this topic. Abney [1] develops a detailed syntactic account of gerunds, which are a subclass of the perfect and imperfect nominals¹³. He distinguishes four types of gerunds:

- (35) a. Acc-ing: John being a spy.
 b. PRO-ing: singing loudly.
 c. Poss-ing: John's knowing the answer.
 d. Ing-of: singing of the song.

Assuming that PRO-ing is a special case of either Acc-ing or Poss-ing, there are three classes of gerunds, which differ with respect to their syntactic properties. For example, Abney shows that Acc-ing and Poss-ing constructions show differences with regard to agreement, long distance binding, pied piping, etc.. We do not think that these structural differences indicate a semantic distinction between Acc-ing and Poss-ing gerunds, leading to differences in external distribution¹⁴. Semantically, we are therefore left with two classes of gerunds¹⁵.

2. History of the English gerundive system

According to the standard theory the gerund in Old and beginning Middle English had purely nominal properties. The rise of the verbal gerund would be due to certain phonological and morphological features the gerund shared with the participle in some dialects. However, the more recent study Houston [53] shows that, although the data on which the traditional explanation is based are correct, the explanation itself is untenable. This is partly

¹²But see the exercises and [47].

¹³The class of imperfect nominals is a huge and structurally heterogeneous class including Poss-ing, Acc-ing gerunds (for which see below), absolute constructions, infinitives and even *that*-clauses, which are traditionally not thought of as *nominal* at all. But note that the concepts *perfect* and *imperfect* nominal are used by Vendler primarily to refer to sets of structural properties, which are assumed to be conditioned by two different *semantic categories*. Abney is only concerned with syntax.

¹⁴For a substantiation of this claim, see [47].

¹⁵Gerunds like *eating potatoes* are chiefly interpreted as habitual, but since in this book we will not develop a formal semantics for habitual or generic readings, they will be neglected here. See Portner [89] for more information about habitual readings.

due to the fact that frequency considerations were neglected in the traditional argumentation. The data in [53] clearly demonstrate that the reasons for the change of the English gerundive system were of a more *semantic* nature, and this is the reason why we include a discussion of this material here.

According to Houston, the similarity between the participle and the verbal gerund as the object of a preposition is based on a shared “backgrounding” function in discourse. A “backgrounding” discourse function provides additional information pertaining to eventualities introduced previously, while the function of “foregrounding” is to introduce new events and thereby move a narrative forward in chronological time, rather than to elaborate on established eventualities. We have seen in Chapter 9 while formalizing the Imparfait, that there are good reasons to identify background information formally with a fluent. Accordingly, by following Houston’s train of thought we may adduce evidence that verbal gerunds should be represented formally by fluents.

We will first discuss the traditional view, thereby introducing the relevant data. Then we will turn to Houston’s criticism of this view, and her own explanation for the change of the English gerundive system¹⁶.

2.1. Traditional accounts. In this Section we will first discuss the standard theory about the development of the English gerundive system as found in Poutsma [90] and then move on to the more recent paper Houston [53] in which the standard theory is criticized, and a different explanation for the historical development of the gerund is worked out. The discussion will mostly be concerned with verbal gerunds, because nominal (i.e. Ing-of) gerunds are less problematic from a grammatical point of view. Once a verb with an *ing* suffix may be viewed as a noun, there is no obstacle to viewing it as a relational noun such as ‘father of’.

The history of English is usually divided up into the following periods¹⁷:

Old English	500 – 1000
Middle English	1100 – 1500
Early Modern English	1500 – 1800
Modern English	1800 – present

The Old English period starts with the invasion of Britain by the Germanic tribes under the leadership of king Vortigern in the year 449. This period ends with the conquest of Britain by William, Duke of Normandy, in 1066. The Early Modern English period starts with the introduction of the printing press and the beginning of the Renaissance, and its end is marked by the independence of the American colonies.

¹⁶This historical Section owes much to input from Darrin Hindsill, whose MSc thesis [49] contains vastly more data than we can hope to cover here.

¹⁷A more extensive discussion of these periods and the principles used to distinguish them can be found in Fennell [39].

Although the thesis has been proposed that the gerund can be found as early as the beginning of the Old English period¹⁸, most scholars reject this thesis¹⁹ and propose early Middle English as the period²⁰ where the first gerunds can be found.

2.1.1. *Productive use of noun endings.* The traditional view of the development of the English gerundive system is based on the assumption that its source is in the Old English noun endings *ung* and *ing*, or in their inflected forms *unge*, *inge*. Some of the nouns from which the *ung*-derivations were formed were also used as weak verbs. This led to the assumption that the verbal forms are the stems of the derivations. This then produced similar forms from other weak verbs, a process which at the beginning of the sixteenth century allowed an *ing* or *ying*-form for practically any verb, although originally the formation of such *ing*-nouns was severely restricted. Examples of Old English verbs and the corresponding *ung*-nouns are given in (36).

- (36) a. *gieddean*
to speak formally, with alliteration
b. *gieddung*
saying
c. *lufianto*
love
d. *lufung*
act of loving
e. *geladian*
invite, summon
f. *geladung*
congregation
Houston [53, p. 174]

The traditional view thus rests on the assumption that a confusion between nouns and verbs is responsible for the formation of gerunds.

2.1.2. *Phonological roots.* This thesis is further elaborated by the observation that – at least in some dialects – the participle ending in the suffix *inde* came to sound similar to the *ing*-noun. This process is usually explained phonologically. The assumption is that a person speaking a Southern or some adjacent Midland dialect developed the habit of dropping the dental *d* after the nasal *n*, which changed *inde* to *inne*. This step was followed by the loss of the final *e*, which is just an instance of the general tendency to drop vowels in unstressed syllables. This then would be the source for confusion of nouns ending in *inge* with the participle ending in *inne*. Here are some examples.

- (37) [Southern dialect, about 1280]

¹⁸See Curme [22] for instance.

¹⁹An early critical discussion of [22] is contained in Einkenel [35] and [36].

²⁰Emonds [37] even doubts that Chaucer's English contains gerunds.

- a. al *fastinde* he lay | at þio holy mannes tounge
Fasting he lay at the tomb of that holy man
- b. Swete lorde . . . Ich am *cominde* to þine feste.
Sweet lord, I am coming to thy feast.
[90, p. 160]

The normal ending of the Midland dialects in Early Middle English was *ende*, but changed later to *inde*. It should be remarked, though, that in Northern dialects the ending was *and*, which maintained itself longer than the Midland *ende*. The nominal ending in Northern dialects was *ing* or *in*.

- (38) [Northern dialect 1303] Echone seyð to oþer jangland | þey toke neuer gode at Pers hand.
Each said chattering to the other they never took alms from Peter's hand.
[90, p. 161]

Around 1200 in Southern and Midland dialects the participle suffix *inde* began to change from the original *inde* to *inge* or *yng*, and therefore became identical to the ending of the verbal noun.

- (39) [~ 1250] Nū bōþe twō þēs swēte þinge | Crie hire mercī al *wēpinge*.
Now both these sweet things weeping cry to her to have mercy upon them.
[90, p. 162]

Since by the phonological process described above nouns ending in *ing* and participles became indistinguishable, the structural phenomena particular to participles could be shifted to the nominal domain. In this way a form of the gerund could arise which has the internal properties of verbs. It is furthermore hypothesized that this development was supported by the French 'en plus gerondif' construction which appears frequently in Middle English in constructions like (40).

- (40) Heo was a gast and *in feringe*.
He was aghast and in fearing.
[90, p. 162]

2.1.3. *Toward Acc-ing gerunds*. It seems that the first occurrences of *ing*-nominals with adverbials were caused by compounds such as *down-coming*, *downfalling* which were broken up into their components first and then the adverb was positioned postverbally leading to *coming down*, *falling down*. This occurred in the middle of the fourteenth century. However, Poutsma cites a much earlier example of a gerund with adverbial modification.

- (41) [~ 1275] þe appostels through *precheing lele* (loyally) | Gederd (= gathered) þam (dative) desciples fele (= many).
[90, p. 163]

It is assumed that a similar process was responsible for the government of direct objects by the nouns ending in *ing*. Thus compounds like *peace-making* or *book-selling* were presumably split up into their component parts and then rearranged. Gerunds governing direct objects are attested in the last quarter of the fourteenth century²¹.

- (42) [~ 1470] I suppose that he hath slayn her in *fulfyllynge his fowle lust of lecherye*.
[90, p. 164]

By contrast the common case of a noun before an *ing*-word is attested quite early. Poutsma cites examples from 1330 till 1340 and traces these constructions back to Old English imitations of Latin originals with participles.

Interestingly, it seems that for some time *ing*-nouns followed by *of* and those with a direct object (i.e. Acc-ing gerunds) were not clearly distinguished. Compare the following examples, in which the two forms occur side by side:

- (43) a. I had the misfortune to displease him by *unveiling of the future* and *revealing all the danger*.
b. and lykewise as *burnyng of thistles* and *diligent weding them oute*²²
[90, p. 165].

In our overview of the internal properties of verbal gerunds in Section 1 we noted that in modern English, verbal gerunds, unlike nominal gerunds, can be modified by voice and the present perfect. The first distinctions of voice and aspect appeared at the end of the sixteenth century. According to Poutsma no complex gerunds can be found before that time. For example, in the following sentence we find the expression *hurtynge* which in modern English would have to be replaced by *being hurt*.

- (44) [1545] A shootynge Gloue is chieflye for to save a mannes fyngers from *hurtynge*.
[90, p. 165].

The following examples are among the first where traces of voice (45) and aspect (46) appear.

- (45) [1585–1591]
a. by *being* unto God *united*
b. For *being* *preferr'd* so well
[90, p. 166].
- (46) [1580] Want of consideration is *not having demanded* thus much.
[90, p. 166].

²¹In the dissertation Irwin [54] some earlier examples of verbal nouns with direct objects were found but none before about 1350.

²²A possible difference is that in constructions with pronouns as objects the structure without *of* is preferred.

2.1.4. *A possible role for inflected infinitives.* Poutsma speculates that a second source of the gerund might be the inflected infinitive. Again the speculation is based on a possible confusion of the usual infinitive endings *ende* or *inde* with the nominal endings *enge* or *inge*. This confusion would then give rise to two infinitival forms like for instance *(to)binden* and *(to)bindenge* or *bindinge*. The first form was used when verbal functions were expressed, the second one for substantival functions. Poutsma cites plenty of examples where final infinitives and final gerunds seem to be confused, and also examples where a gerund would be expressed by an infinitive in modern English.

- (47) Behold what honest clothes you send forth to *bleaching*.
[90], p 168

In modern English *bleaching* would be changed to *to be bleached*.

To summarize: according to the traditional view, the main cause for the development of the English gerundive system is a phonological process, leading to the formal identity of the participle with the noun ending in *ing*. It is furthermore assumed that the ability of the verbal gerund to govern direct objects is due to splitting compounds and the rearrangement of their parts. Therefore phonological and morphological processes would be the source of the English gerundive system.

2.2. A semantic origin for gerunds: Houston's theory. The traditional view has been criticized by Ann Houston in [53]. She argues that the formal identity of two grammatical categories *alone* is not sufficient for confusing them, some overlap between syntactic function is also necessary. But in the period where the spelling of the participle changed, the syntactic distributions of participles and gerunds were essentially different. Houston cites Irwin's study [54] which shows that verbal nouns occurred primarily as subjects, objects, prepositional objects and genitive complements. By contrast present participles were used as nominal modifiers and parts of phrasal verbs. According to the data presented in [54] there is only a 5% change in the distribution of verbal nouns and only a 4% change in the distribution of the participles between the fourteenth and fifteenth centuries, which is the time period in which the most extensive change in the spelling of the participle occurred. This makes the "confusion" thesis rather implausible.

The second component of the traditional view, which sees the origin of Acc-ing gerunds in the splitting of compounds, is also criticized by Houston. She points out that the respective "splitting" constructions are rare in proportion to the frequency of verbal nouns. Moreover compounds like *good-doing* or *almes-giving* are found in texts that already contain verbal nouns governing direct objects. Houston's own data show a high frequency of alternations between verbal nouns followed by prepositional objects and direct objects, rather than alternation between compounds and their inverted parts. Houston illustrates the alternation she found with the following data.

Sentences (48-a), (48-b) and (48-d) involve an indirect object, whereas (48-c) and (48-e) use direct objects.

- (48) a. makyng of a gode emplastre
 b. makyng of the litel howse
 c. making a man slepe
 d. wakyng of hym
 e. contraryng your commandment
 [53, p. 181]

Although the data mentioned in Section 2.1 are accepted in Irwin's and Houston's work, their investigations on frequency of occurrence of the various forms show that the explanation for the rise of the verbal gerund offered by the traditional thesis is untenable.

2.2.1. *Discourse function of gerunds.* Houston's own explanation is not based on phonological or morphological similarities, but on a discourse function shared by gerunds and (appositive) participles. She therefore offers a functional explanation for the rise of the verbal gerund. Her starting point is the observation that as the verbal noun began to govern direct objects, this construction (i.e. the verbal noun) occurred at first primarily as the object of a preposition and only about 50 years later also in subject and object positions. This is statistically significant in the period ranging from 1450 till 1550. For the following one hundred years no statistically significant results could be exhibited. The following table summarizes Houston's data:

Date	Sub/Obj		Oblique		Total
	%	number	%	number	
c1350	0	0/42	2	1/41	85
c1400	0	0/22	0	0/40	62
c1450	3	1/39	3	4/127	166
c1500	0	0/20	4	1/26	46
c1550	4	1/28	31	13/42	66
c1600	54	15/24	64	40/62	90
c1650	38	5/13	60	38/63	76
Total					591

In this table, c1350 indicates the second half of the 14th century, c1400, the first half of the 15th. *Oblique* is short for prepositional object position.

The frequency of appositive participles with direct objects was already high during the Old English period, whereas verbal nouns with direct objects occurred rarely even in translations. This leads Houston to explore the following issue:

Even if there was never a widespread confusion between participles and verbal nouns, is it still possible that the verbal characteristics of the *appositive* participle constructions influenced the acquisition of verbal traits by the verbal noun? If so, how

could such an influence account for the initial lead of the prepositional object verbal nouns in this change?
[53, p. 182] [emphasis added]

Syntactically, appositive participles could occur either clause initially or at the end of a clause, as examples (49) and (50) illustrate.

- (49) *Going to preach*, H. Morley of my parish deliv'd mee a note of receipt of my procurations.
[53, p. 183]
- (50) I recommend me hartly onto yow, *thankyng yow* of all good brotherhood.
[53, p. 183]

This is a *syntactic* feature they often share with verbal nouns as objects of prepositions, and this syntactic correspondence might provide a springboard for further correspondences. In order to investigate this possibility, let us consider appositive participles in somewhat greater detail.

According to Callaway [13], there are three *uses* of the appositive participle, *attributive*, *adverbial*, and *coordinate*. We provide an example of each.

- (51) [attributive] Unto my brother George Cely merchande of the estapell *beyng at Calles*.
[53, p. 184]
- (52) [adverbial] Sir Samuel Baguel is lately slain there, *being stabd by Sir Laurence*²³.
[53, p. 187]
- (53) [coordinate] The Quene removed on Wensday toward Norfolk, *taking Dr. Cesars in her way*.
[53, p. 185]

For Houston, the important use is the adverbial one. The reason is that, according to Houston's data, the function which is most extensively shared by verbal nouns and appositive participles is the adverbial use. This use of appositive participles overlaps with the use of the verbal noun as an object of prepositions. It does not overlap with verbal nouns in subject and object positions. The next table presents an overview over the basic functions of verbal nouns and appositive participles:

	Att	Adv	Coord	Sub	Obj
Appositives	+	+	+	-	-
Prep. Obj. Verb. Nouns	+	+	-	-	-
Verbal Nouns	-	-	-	+	+

(Adapted from [53, p. 185])

Having thus established the importance of the adverbial use of appositive participles, Houston (taking her cue from Callaway [13]) distinguishes four

²³Example (52) is adverbial in the sense that the appositive participle stands in a causal relation to the main clause.

adverbial types: modal, temporal, causal, and goal. For each type examples of appositive participles and verbal gerunds are given.

- (54) [causal]
- a. (= (52)) [appositive] Sir Samuel Baguel is lately slain there, *being stabd by Sir Laurence*.
 - b. [verbal noun] God zelde yow for zoure labore *for gaderyng of my mony*.
[53, p. 187]
- (55) [temporal]
- a. [appositive] In the day, forsothe, folowyng, I *biholdyng the finger* I perceyued that the arsenek had wrouzt littel or nozt.
 - b. [verbal noun] Dr. Parkins, *at his first coming out of Denmarke*, made his braggs that he had bought . . .
[53, p. 187]
- (56) [modal]
- a. [appositive] (He) set upon him as he was coming out of his coach, *wounding him in three or four places*.
 - b. [verbal noun] Wee are very vigerous *in asserting our Religion* . . .
[53, p. 187,188]
- (57) [goal]
- a. [appositive] Lord yett in mercy shew mee favor in him *making him a comfort*.
 - b. [verbal noun] the generallitie of ye Privy Councell immediately move for ye *setting up of ye Militia here*.
[53, p. 188]

Houston furthermore notices that verbal gerunds already had adverbial function before they appeared with direct objects, as illustrated by the following example:

- (58) . . . this boon hath in sum maner of men a smal seen in *foldyng of the forehead* . . .
[53, p. 185]

On the basis of the data reviewed above, Houston suggests that appositive participles and verbal nouns occurring as the object of a preposition share a common *backgrounding* discourse function. This contrasts with verbal nouns in subject or object position, which have a *foregrounding* discourse function. According to Hopper [52] “foregrounding” is characterized by reference to events which belong to the skeletal structure of the discourse, events moving the discourse forward in chronological time. “Backgrounding” by contrast refers to events which provide supportive material and do not move the discourse forward in chronological time. In this sense both appositive participles and verbal nouns as objects of preposition serve a backgrounding function. They provide additional information about time,

manner, means etc.. This contrasts sharply with the function of verbal nouns in subject or object positions; Houston claims, on the basis of the data reviewed above, that in these positions verbal gerunds occur primarily in descriptions of events that are of central interest and therefore serve a foregrounding function, as in

- (59) Hee denies *the bringing the army to London*²⁴.
[53, p. 189]

So which came first, appositive participles or verbal nouns? It seems unlikely that the verbal noun influenced appositives with respect to governing direct objects, because appositive participles governed direct objects much earlier and with greater frequency than verbal nouns. On the contrary, the shared backgrounding discourse function of appositive participles and verbal nouns as objects of preposition explains much better why more and more verbal traits (such as governing direct objects) occurred in the latter construction. This may then have led to a transfer of the semantic representation of appositive participles to verbal gerunds as the object of a preposition. In the context of the event calculus, background can be represented as a fluent, as we have seen while discussing the Imparfait in Chapter 9. Also, a glance at the examples in (54), (55), (56) and (57) shows that it might very well be possible that the verbal gerund introduces a fluent, serving as a backdrop for the events introduced in the main clauses.

We have to note here that the concepts that the concepts *backgrounding* and *foregrounding* are not always used coherently in the linguistic literature. Sometimes²⁵ passives, topicalizations and left-dislocations are called foregrounding operations. For example, *passive* is described as backgrounding the agent and foregrounding the patient; *foregrounding* in this context means “draw our attention to”, rather than “introducing a new discourse referent”. Indeed, according to Hopper’s concept of foregrounding a typical expression used for foregrounding is an indefinite noun phrase. Such an expression usually introduces a new discourse referent which moves the event forward in chronological time. By contrast, the second concept of foregrounding points to definites, including demonstratives, as typical expressions used in foregrounding operations. The second concept of foregrounding may be relevant for examples involving both anaphora and kataphora such as (60) and (61).

- (60) Cromwell’s beheading the king is still surprising for us, although it occurred more than 300 years ago.
- (61) Although it occurred more than 300 years ago, Cromwell’s beheading the king is still surprising for us.

²⁴Observe that here a determiner is used with an imperfect nominal. Although this is now ungrammatical, it used to be quite common. See Hindsill [49] for further details.

²⁵See for instance Keenan [61].

In these examples, a pronoun referring to an imperfect nominal may nevertheless occur as an argument of a narrow container. It thus seems that foregrounding may operate on imperfect nominals to make them available as *event types* to be referred to by pronouns in preceding or subsequent sentences. We have unfortunately no theory to offer here.

3. Nominalizations formalized I: Denotation types

Before we show how to develop a formal account for the observed data, we review some basic philosophical issues.

3.1. Introduction. In Cocchiarella [16] predicate nominalizations, considered as transformations of predicates and predicate phrases into nouns or noun phrases, are investigated with regard to the traditional philosophical positions realism, conceptualism and nominalism. The general theory proposed in [16] is meant to apply not only to the examples we are considering in this book, but also cases of adjective nominalization such as *pious – piety*, *wise – wisdom* and various kinds of lexical nominalizations. However, Cocchiarella clearly states that his aim is not a description of distinctions relevant for linguistics, such as different types of nominalizations. His intention is to investigate a logic of nominalized predicates within a general theory of predication.

Here we will only discuss some general philosophical views of nominalized predicates and how the approach advocated in this book relates to two of these positions²⁶.

For Cocchiarella, Platonism is a form of logical realism. On the Platonistic view a nominalized predicate refers as a singular term to the same property or relation which is denoted by the predicate. This is the position taken up by Parsons with regard to “eventive” gerunds.

The underlying event analysis provides the means for a neat solution by proposing that nominal gerunds contribute the *very same* predicates to logical form as the verbs on which they are based.

[87, p. 17]

Not all forms of logical realism share this position. Frege, for instance was a logical realist, but he assumed that although nominalized predicates of natural language are singular terms these terms nevertheless never refer to what predicates in their role as predicates denote. Frege held that nominalized predicates as singular terms refer to certain individuals correlated with the properties and relations indicated by the predicates. Frege called these individuals *concept-correlates*. Cocchiarella characterizes a view of nominalized predicates as *Fregean* if it satisfies the following two postulates:

- nominalized predicates cannot refer as singular terms to the same universals which these predicates stand for in their role as predicates.

²⁶We will not discuss nominalism.

- nominalized predicates refer instead to certain individuals which are somehow correlated with these universals.

In this sense also a conceptualist theory of nominalized predicates can be Fregean, since concepts as cognitive entities or structures are unsaturated intelligible universals. It is consistent with conceptualism to hold that nominalizations are singular terms which denote as well as to assume that nominalizations are denotationless singular terms.

A conceptualist might even adopt a mixed strategy, holding that some nominalized predicates denote concept–correlates whereas others, as a matter of conceptual necessity (as in the case of Russell’s paradox) must be denotationless.
Cocchiarella [16, p. 167]

We will here assume a conceptualist Fregean view of nominalizations, but we will add one important ingredient. There is not one type of concept correlate but there are two, which we will now define and then explain with some simple examples.

3.2. Putting Feferman to work. The task at hand is to provide formal procedures which, starting from a verb phrase, produce the denotations of perfect and imperfect nominals, subject to the boundary conditions that the logical properties highlighted in Section 1 become provable, and also, that there is a relation between the syntax of nominalization and the semantic nominalization procedures. From this point on, acquaintance with the material on Feferman’s calculus explained in Chapter 6 is essential. To keep things simple, we will at first abstract from the *Aktionsart* of the VP that is nominalized. Section 4 shows how to handle this additional complication. Even so, we acknowledge that we can barely scratch the surface here. Nominalization is a very complicated phenomenon, especially in its imperfect guise, and to come up with a complete theory would require for instance making finer distinctions among imperfect nominals.

As in the previous chapters, our starting point is a verb (i.e. a predicate) with a temporal parameter. The Feferman calculus then provides two natural ways of turning this verb into an object, and this will give us a good first approximation to a formalized theory of nominalization.

The first possibility is that, before the verb is turned into an object, the temporal parameter in the verb is suppressed by means of existential quantification.. Coding then produces an entity which is in some sense atemporal.

DEFINITION 44. *If $\varphi(\bar{x}, t_1 \dots t_n)$ is a formula, the event type generated by φ will be $\exists t_1 \dots t_n. \varphi(\bar{x}, t_1 \dots t_n)[\bar{x}]$.*

The form involving several variables for time is useful to derive a lattice structure on event types, as will be seen below. When no confusion can arise, we shall usually write φ instead of $\varphi(\bar{x}, t_1 \dots t_n)$. Event types

constructed in this way may occur as event-arguments in applications of the event calculus.

The second possibility is that a verb is mapped onto the (intensional) set of instants at which it holds, using Feferman's abstraction notation.

DEFINITION 45. *The imperfect nominal derived from an expression $\varphi(\bar{x}, t)$ is the fluent²⁷ $\varphi[\bar{x}, \hat{t}]$.*

Note that this definition does not say that imperfect nominals denote *propositions*. But the right hand side in the following biconditional may denote a proposition:

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

Therefore the denotations of imperfect nominals are not propositions but are systematically related to propositions.

Definitions 44 and 45 provide the two Fregean concept correlates which we will use in this chapter. We will illustrate these definitions by applying them to the perfect nominal *John's burning of the house* and the imperfect nominal *John's burning the house*. Concrete lexical content will be added in Section 4. We assume that a predicate $\text{burn}(x, y, t)$ is given. The predicate $\text{burn}(x, y, t)$ means that x burns y at t . The perfect nominal *John's burning of the house* will then be translated as:

$$(62) \quad \exists t. \text{burn}(x, y, t)[j, h]$$

This is an example of the first type of concept–correlate, an individual which is an event type. The term in (62) can therefore occur in those arguments of predicates of the event calculus which are reserved for event types, especially *Happens*. For instance $\text{Happens}(\exists t. \text{burn}(x, y, t)[j, h], s)$ is a well-formed formula of the event calculus.

Consider next the imperfect nominal *John's burning the house*. According to definition 45 this nominal translates as:

$$(63) \quad \text{burn}(x, y, t)[j, h, \hat{t}]$$

Note that this term indicates a function of t , with parameters x and y .

Clearly $\varphi[\bar{x}, \hat{t}]$ may be substituted for fluent–arguments in the event calculus. For instance, since *HoldsAt* is a special case of Feferman's T_1 we have

$$\text{HoldsAt}(\text{burn}[j, h, \hat{t}], s) \leftrightarrow \text{burn}(j, h, s).$$

This is a concrete example which shows how imperfect nominals are related to propositions via the *HoldsAt*-predicate.

²⁷We implicitly assume α -conversion here, so that bound variables can be replaced *salva veritate*. Even so, these objects are strongly intensional.

Notation: Henceforth we will often use the notation $\exists t. burn[j, h, t]$ to abbreviate terms like those in (62) and similarly for imperfect nominals, i.e. we will write $burn[j, h, \hat{t}]$ for $burn(x, y, t)[j, h, \hat{t}]$

In the next Section we will use the two concept–correlates to fix denotation types for perfect and imperfect nominals and the two types of containers and then show how this allows us to deduce some simple predictions.

3.3. Denotation types for nominals and containers. The following table summarizes our assumptions about the denotation types of perfect and imperfect nominals, narrow and loose container and (binary) determiners.

Perfect nominal	set of event types
Imperfect nominal	fluent
Narrow container	set of event tokens; i.e. a subset of <i>Happens</i>
Wide container	set of fluents ²⁸
Binary determiner	see Section 3.4

These stipulations lead to an immediate consequence. Perfect nominals cannot be internally modified by tense and aspect since the temporal parameter is bound by existential quantification. Imperfect nominals by contrast can be so modified since they are of the form $\varphi[\bar{x}, \hat{t}]$ with the temporal parameter abstracted. Trivial as this explanation is, it nevertheless exhibits one important point. It is expressed with reference to denotation types only, the concrete lexical content of perfect or imperfect nominals is completely irrelevant for the above observation²⁹. The same holds for the explanations proposed below for more involved phenomena. Lexical content becomes important in Section 4 when we consider in greater formal detail how VPs can be transformed into event types or fluents.

The assumptions in the above table allow an immediate explanation for the contrast in (64), and hence account for Vendler’s main observation.

- (64) a. Your breaking the record was a surprise.
 b. *Your breaking the record took place at ten.

In (64-a) the expression *be a surprise* denotes a set of fluents and the fluent $break[x, record, \hat{t}]$ may well be an element of this set³⁰. By contrast the expression *took place at ten* denotes a set of event tokens which does not tolerate fluent elements. Therefore the unacceptability of (64-b) is due to a type conflict. Note especially that an expression like

$$Happens(break[x, record, \hat{t}], t)$$

is not well formed.

²⁸By this we mean the following: since a wide container is a verb, it has a parameter t for time. In principle, the set of fluents depends upon t .

²⁹For this brief discussion it suffices to conceive of event tokens as pairs (e, t) such that $Happens(e, t)$. Temporally extended events need a slightly more sophisticated treatment, which will be given in Section 3.4.

³⁰We will worry about tense in Section 3.5.

3.4. Determiners. We now turn to the interaction of Ing–of gerunds with determiners. Consider (65), slightly adapted from an example in Google:

- (65) (During the morning rehearsals,) every singing of ‘A Poultry Tale’ lasted five minutes.

The computational approach chosen here means that the treatment of determiners must be different from the customary one³¹, which considers e.g. a binary determiner as a relation between sets³². We can allow only those relations which are somehow computable in our constraint logic programming framework. For instance, the existential quantifier, conceived as a binary determiner $\exists x(A(x), B(x))$ is interpreted as the integrity constraint ‘ $?A(x), B(x)$ succeeds’. Likewise the universal quantifier, in its role as the binary determiner $\forall x(A(x), B(x))$ must be interpreted as the integrity constraint ‘ $?A(x), \neg B(x)$ fails’. One may wonder what to make of the quintessential binary determiner ‘most’, defined by:

$mostx(A(x), B(x))$ iff more than half of the A ’s are B ’s.

This requires a mechanism for counting the number of satisfying instances, so that one can compare the cardinality of the set of satisfying instances of $?A(x)$ with that of $?A(x), B(x)$. This can indeed be done by looking at the successful branches of the respective derivation trees, but here we will concentrate on first order definable quantifiers.

Another important difference between the present approach to determiners and the customary one is that here tense is built into the determiner. Traditionally, a sentence like

- (66) Every student passed the exam.

is analyzed as a relation between sets, disregarding the past tense. Here, tense is taken into account via the *Happens* predicate occurring in the restrictor, which relates event types occurring in the restrictor to event tokens in the nuclear scope. In this Section we consider only the case where the nuclear scope is a narrow container; the case of a wide container is left as an exercise, which needs the material of Section 3.5. Since fluents cannot occur as arguments of the *Happens* predicate, this strategy of incorporating tense immediately explains why (67) is unacceptable.

- (67) *Every singing ‘A Poultry Tale’

For simplicity we assume here that the event type *sing* is represented as the term $\exists t.sing[x, y, t]$ denoting a function which maps two individuals, a subject x and a song y , to an event type. Applying this function to the NP ‘A Poultry Tale’, represented by the constant p , yields the one-parameter event type $\exists t.sing[x, p, t]$, with a free parameter for the subject. This is

³¹And also different from the treatment given in [47].

³²For more detailed information about the standard set theoretic semantics of determiners see Westerståhl [130].

a considerable simplification, since we know that ‘sing ‘A Poultry Tale’ ’ is actually an accomplishment. A more elaborate analysis will be given in Section 4. The VP *lasted five minutes* applies to event tokens; the next task is to associate a token to the type. This is slightly nontrivial since any instantiation of the event type under consideration is temporally extended. Here we can use both the structure of minimal models and the available coding machinery to good effect. Intuitively, an event token corresponding to e is a *maximal* interval $(u, v]$ such that

$$u < s \leq v \leftrightarrow \text{Happens}(e, s).$$

Using coding techniques as explained in Chapter 6 the half-open interval can be represented by a single term. The left-to-right direction of the preceding equation can be written as a program clause. The right-to-left direction cannot be so written, since constraints are not allowed in the head of a clause; here we must use an integrity constraint to enforce maximality of the interval.

DEFINITION 46. *Let u, v be terms defining real numbers. The interval $(u, v]$ is an event token of the event type e if*

- (1) $u < s \leq v \rightarrow \text{Happens}(e, s)$
- (2) *for all terms t with $t > v$, the query $?u < s < t, \neg \text{Happens}(e, s)$ succeeds; and similarly for terms t with $t \leq u$.*

If $(u, v]$ is an event token of the event type e , we also write $\text{Happens}(e, (u, v])$.

The precise formalisation of example (65) without the phrase in brackets is now as follows. Abbreviate $\exists t. \text{sing}[x, p, t]$ to $e(x)$, then (65) is represented by the integrity constraint

$$?\text{Happens}(e(x), (u, v]), v < \text{now}, v - u \neq 5\text{min} \text{ fails.}$$

The form of the integrity constraint is dictated by the meaning of the universal quantifier $\forall x(A(x), B(x))$ as ‘ $?A(x), \neg B(x)$ fails’. The predicate *Happens* defines the restrictor, which is a set of event tokens derived from a parametrized event type $e(x)$. The next two conjuncts define the nuclear scope, in the case the past tense form of the narrow container ‘lasts five minutes’. Adding the phrase ‘During the morning rehearsals’ means that the set of event tokens $(u, v]$ is further restricted by a predicate *morning session*.

Determiners like *the* or *John’s* can be treated similarly. For example, the computational meaning of *thex*($A(x), B(x)$) is that $?A(x), B(x)$ must have a unique satisfying instance. Note that in contrast to Poss-ing gerunds the possessive *John’s* will be analyzed as a determiner, i.e. as the universal quantifier restricted to the set of actions that have John as an agent. In Section 3.5 we will show that sometimes determiners relate event tokens also to certain types of fluents. This will allow us to account for Vendler’s observation that in the context of loose containers perfect nominals tend to be interpreted as imperfect.

3.5. Coercion of nominals, and the role of tense. Vendler observed that in the context of wide containers, perfect nominals tend to be interpreted as being imperfect: a sentence like

(68) The collapse of the Germans is unlikely.

seems to have as a possible interpretation

(69) That the Germans will collapse is unlikely.

In this case the future tense does not seem the only possibility; one can imagine an officer, who upon receiving a report from the front refuses to believe what he reads, and utters (68), this time meaning

(70) That the Germans collapsed is unlikely.

Both examples suggest that the denotation of a perfect nominal given in Definition 44 is reinterpreted when it occurs in the context of a wide container, although the process is perhaps not completely deterministic. To formalize this process of reinterpretation, we therefore need a semantic representation for ‘that’-clauses. As a first approximation, we conceive of a ‘that’-clause as saying that an event occurs (possibly with other tenses or aspects instead of the present tense). This motivates the following construction

DEFINITION 47. *Let e be an event type, then there exists a canonical fluent f associated to e defined by $f = \text{Happens}[e, \hat{t}]$. We will refer to this fluent as $\text{that}(e)$. We also define tensed variants of $\text{that}(e)$ as follows*

- (1) $\text{that}_{Pa}(e) = (\text{Happens}(e, t) \wedge t < R)[\hat{R}]$
- (2) $\text{that}_{Fu}(e) = (\text{Happens}(e, t) \wedge t > R)[\hat{R}]$

Observe that

$\text{HoldsAt}(\text{that}_{Pa}(e), \text{now})$ iff $?\text{Happens}(e, t), t < \text{now}$ succeeds,

so that the complementizer translates an integrity constraint into a sentence, as it should.

Armed with this definition, let us consider how to construct semantic representations for (68) and (69). Let e be short for the event type ‘the collapse of the Germans’, and let *unlikely* be the fluent representing the wide container ‘be unlikely’; this fluent may itself take events and fluents as arguments. Then (68) is represented by the integrity constraint

$$?\text{HoldsAt}(\text{unlikely}(e), \text{now}) \text{ succeeds.}$$

Here is an informal description of what goes on in an example such as (69) and (70). We assume that the sentence is built from the sentence ‘The Germans will collapse’, the complementizer ‘that’, and the wide container ‘be unlikely’, used in the present tense. As we have seen in Chapter 8, the representation of ‘The Germans will collapse’, is the integrity constraint

(where e is as above)

$?Happens(e, t), t > now$ succeeds.

This integrity constraint can be internalized using the $that_{Fu}$ complementizer, and we get for (69)

$?HoldsAt(unlikely(that_{Fu}(e)), now)$ succeeds.

Similarly in the case of (70), the integrity constraint must read

$?HoldsAt(unlikely(that_{Pa}(e)), now)$ succeeds.

The wide container ‘unlikely’ creates an intensional context in the sense that the fluent argument of ‘unlikely’ need not be true. The opposite occurs for ‘surprise’, which is governed by a meaning postulate of the form

$HoldsAt(surprise(g), s) \rightarrow HoldsAt(g, s)$.

Now substitute $that_{Pa}(e)$ for g and one then easily checks that the integrity constraint corresponding to (71)

(71) The beheading of the king surprised us.

namely (for $e =$ ‘the beheading of the king’)

$?HoldsAt(surprise(that_{Pa}(e)), t), t < now$ succeeds,

entails that e happened before now.

Examples (68) and (71) shows that certain verbal contexts enforce an imperfect reading of a perfect nominal. Conversely, Chapter 11 contains many examples which demonstrate that the choice of an embedded NP can coerce the reading of the VP which contains that NP. Therefore, the nominal and the verbal systems of natural language seem to be semantically interdependent due to coercion.

3.6. Intensionality of nominals. An important issue that now has to be addressed is that of extensionality versus intensionality of fluents and event types. It seems advantageous to take fluents as intensional entities. If we say

(72) Mary predicted the king’s beheading.

then, even in the case that the king is actually identical to the red-haired spy, we still do not want to infer from this that

(73) Mary predicted the red-haired spy’s beheading.

This can easily be modeled in the Feferman calculus. Even when the formulas $\varphi(t, x)$ and $\psi(t, x)$ are logically equivalent, the terms $\varphi[\hat{t}, x]$ and $\psi[\hat{t}, x]$ are different, and the calculus contains no axiom of extensionality which can force equality of the sets these terms represent. Here is another example of the same phenomenon: if one doesn’t know that Bill is John’s friend,

the following two sentences involving imperfect nominals can be true simultaneously

- (74) a. John's greeting Bill surprises me.
b. John's greeting his friend does not surprise me.

A last example is one discussed by Zucchi [135, p. 185]. Suppose Gianni was going by train from Milan to Florence, but due to a strike of the railroad workers, he only got as far as Piacenza. On a Parsons-type approach to events, there is the following problem. Let e be the trip that Gianni took on this occasion and t the time at which he reached Piacenza. Event e does not culminate at t , since e is an unfinished trip to Florence, and Gianni is at Piacenza. But e is also a trip to Piacenza, which does culminate at t . On the present analysis there is no problem at all, since the trips to Florence and Piacenza would be represented by different fluents, which simply happen to share their space-time behavior from Milan to Piacenza. The predicate *Terminates* can well be true of one, but not of the other fluent. That is, if f is the fluent corresponding to a trip to Florence, and g the fluent corresponding to a trip to Piacenza, a the event of reaching Piacenza, b the strike, then the scenario would feature the conditions *Terminates*(a, g, t) and *Terminates*(b, f, t); computing the completion would then have the effect of enforcing \neg *Terminates*(a, f, t), as required.

The canonical fluent associated to event types e used for interpreting coercion (*that*(e) and its tensed variants) is also important for the intensionality that some containers enforce. Compare sentences (75) and (76)

- (75) The beheading of the tallest spy occurred at noon.
(76) Mary predicted the beheading of the tallest spy.

Even when *the king = the tallest spy*, (76) does not imply

- (77) Mary predicted the beheading of the king.

whereas we of course do have

- (78) The beheading of the king occurred at noon.

This can now be explained, if we assume that in the context of the wide container 'predict' the event type e is replaced by *that*_{Fu}(e) (because of the meaning of 'predict', *that*_{Pa}(e) is excluded here). Let $e(k)$ be short for 'the beheading of the king', and likewise $e(s)$ for 'the beheading of the tallest spy'. After coercion, sentence (76) is represented by the integrity constraint

$$?HoldsAt(predict(that_{Fu}(e(s))), t), t < now \text{ succeeds,}$$

whereas (77) is represented by

$$?HoldsAt(predict(that_{Fu}(e(k))), t), t < now \text{ succeeds.}$$

Since the fluents $that_{Fu}(e(k))$ and $that_{Fu}(e(s))$ are intensionally different even if $k = s$, the two integrity constraints are independent.

Now look at sentences (75) and (78), and their semantic representations

$$?Happens(e(s), t), noon(t), t < now \text{ succeeds}$$

and

$$?Happens(e(k), t), noon(t), t < now \text{ succeeds.}$$

In this case $s = k$ has the consequence that the two integrity constraints are simultaneously (un)satisfiable.

3.7. Present perfect in imperfect nominals. We have seen in Section 1 that imperfect nominals allow some form of aspectual modification, namely the application of the present perfect, as in

(79) He admits *having revealed the secret*.

The semantic representation for (79) should be such that it implies

(80) He has revealed the secret.

and is equivalent to

(81) He admits that he has revealed the secret.

Sentences (80) and (81) provide the clue to the proper representation. We first formalize the present perfect along the lines indicated in Chapter 10. Let f be the state resulting from the act e of revealing the secret; the scenario must therefore contain the statement $Initiates(e, f, t)$. With this notation, (79) is represented by the integrity constraint

$$?HoldsAt(f, now) \text{ succeeds.}$$

Adapting the definition of *that* to this context, we obtain $that_{PP}(f) = HoldsAt[f, \hat{R}]$. Sentence (79) is then represented by the integrity constraint

$$?HoldsAt(admit(that_{PP}(f)), now) \text{ succeeds.}$$

The verb ‘admit’ satisfies a meaning postulate which can be rendered as (for g an arbitrary fluent)

$$HoldsAt(admit(g), s) \rightarrow HoldsAt(g, s).$$

Substituting $that_{PP}(f)$ for g then yields the implication from (79) to (80).

3.8. Lattice structure of the set of fluents. We have seen in Section 1 that imperfect nominals can be combined by means of conjunction, disjunction and negation. Formally, this requires defining these operations on the *terms* interpreting imperfect nominals. This does not present a problem, since these terms, at least when L_0 -definable³³, form a Boolean algebra. Thus,

³³ L_0 -definability was introduced in Chapter 6.

LEMMA 5. *If f_1, f_2 are L_0 -definable fluents, we have*

- (1) $\text{HoldsAt}(f_1 \wedge f_2, t) \leftrightarrow \text{HoldsAt}(f_1, t) \wedge \text{HoldsAt}(f_2, t)$ and similarly for \vee ;
 (2) $\neg\text{HoldsAt}(f_1, t) \leftrightarrow \text{HoldsAt}(\neg f_1, t)$.

Sometimes, however, imperfect nominals cannot be interpreted as L_0 -definable fluents, because they somehow already involve a truth predicate. This can occur for essentially three reasons. The first has to do with the *coercion* of perfect nominals in the context of a wide container. This was explained more fully in Section 3.5: as an argument of the verbal context *is unlikely* the perfect nominal *the collapse of the Germans* is interpreted as the imperfect nominal *that the Germans will collapse*. Formally, an event type is mapped onto the fluent $\text{that}(e) = \text{Happens}[e, \hat{t}]$, or one of its tensed variants.

This construction may lead to fluents which themselves involve reference to the *HoldsAt* predicate. For instance, if one first defines an event type e by means of hierarchical planning, and then forms that_e , the resulting fluent, when interpreted in the minimal model, will implicitly refer to *HoldsAt*. The truth theory developed in Chapter 6 shows that part 2 of Lemma 5 no longer holds. In particular, occurrences of $\neg\text{HoldsAt}$ then have to be replaced by their positive counterparts $\overline{\text{HoldsAt}}$. In linguistic terms, this means that negation in such contexts has antonymic force. And indeed, although opinions differ as to whether Cooper's example

- (82) Andrew's not stopping before the traffic light took place at noon.

is quite grammatical, no such problems seem to arise when the container is wide:

- (83) Andrew's not stopping before the traffic light caused a commotion.

except that now 'not stopping' appears to have the meaning of an antonym to 'stopping', in line with the above analysis. Thus we see that we really need the full strength of the Feferman calculus, and cannot content ourselves with a truth predicate that operates on L_0 -formulas only, as is customary in the treatments of the event calculus current in artificial intelligence (if done formally at all).

This point is corroborated when we look at iterated nominalizations. Consider

- (84) a. John supports his son's not going to church.
 b. John's supporting his son's not going to church causes me much chagrin.

The implication of the use of *support* is that John's son is actually not going to church, so *support* satisfies a meaning postulate of the form

$$\text{support}(x, g, t) \rightarrow \text{HoldsAt}(g, t).$$

If f is the fluent *John's son going to church*, then we get as a special case

$$\text{support}(x, \neg f, t) \rightarrow \neg \text{HoldsAt}(f, t).$$

(84-a) is formalized by the integrity constraint

$$?\text{HoldsAt}(\text{support}[j, \neg f, \hat{t}], \text{now}) \text{ succeeds.}$$

Nominalizing this sentence yields a fluent of the form $\text{that}(g)$, whence (84-b) is represented by the integrity constraint

$$?\text{HoldsAt}(\text{chagrin}(\text{that}(g)), \text{now}) \text{ succeeds.}$$

If one unpacks the latter constraint, one sees that there is implicitly a negative occurrence of *HoldsAt* inside *HoldsAt*, which forces the negation to be nonclassical.

3.9. Lattice structure of the set of event types. By Definition 44 of event types as terms of the form $\exists t_1 \dots t_n. \varphi(t_1 \dots t_n, \bar{x})[\bar{x}]$, closure of the set of event types under \vee and \wedge is immediate. But since *Happens* is not a truth predicate, we have to augment scenarios with some additional formulas to ensure that *Happens* behaves properly with respect to these operations. The following clauses are just special cases of what was called a *definition* in Chapter 4 (see definition 9).

DEFINITION 48. *The operations \wedge and \vee on event types are defined by:*
 $(\wedge) \text{Happens}(e, t) \wedge \text{Happens}(e', t) \rightarrow \text{Happens}(e \wedge e', t)$
 $(\vee) \text{Happens}(e, t) \vee \text{Happens}(e', t) \rightarrow \text{Happens}(e \vee e', t)$

Once these definitions are added, it is clear that the lattice structure of perfect nominals is mirrored in that of the event types.

This lattice structure is of interest in view of the observation of Bach [6] and others (e.g. Link [73], Krifka [66], Lasersohn [69], Eckhardt [33]), that there exists a close parallel between the pair mass/count nouns on the one hand, and the pair processes/events on the other. Bach puts this in the form of the following equation³⁴:

$$\text{events:processes} :: \text{things:stuff}$$

Now just as there exists a mapping which associates to things the stuff they are made of, there should exist a mapping which associates to an event type a process, so that, e.g., a running event is mapped onto the ‘stuff’ it consists of, namely the activity running. This mapping should commute with conjunction and should respect temporal relationships such as ‘overlaps’³⁵. Now clearly our setup yields such a mapping for free, namely the mapping

$$e \mapsto \text{that}(e) = \text{Happens}[e, \hat{t}].$$

³⁴For which we have seen some psychological evidence in Section 1 of Chapter 2.

³⁵A remark is in order here: we require commutation with conjunction whereas Bach and Link require commutation with *disjunction*. This is because we have a different view of plural events: whereas Link (*op. cit.*, p. 247) considers *John and Bill hit each other* to consist of the *sum* of the events *John hit Bill* and *Bill hit John*, we believe it might as well be described as a conjunction.

Once definition 48 is added, this mapping is a homomorphism with respect to the *HoldsAt*-predicate; that is, the following biconditionals are true in minimal models:

$$\begin{aligned} \text{HoldsAt}(\text{that}(e \wedge e'), s) &\leftrightarrow \text{Happens}(e \wedge e', s) \leftrightarrow \text{Happens}(e, s) \wedge \\ &\text{Happens}(e', s) \leftrightarrow \text{HoldsAt}(\text{that}(e), s) \wedge \text{HoldsAt}(\text{that}(e'), s) \end{aligned}$$

This mapping is many-to-one in the sense that event types may be mapped onto fluents which are extensionally equivalent (i.e. as functions of time), even though the event types themselves are different. Bach's examples are

(85) Jones poison the populace.

(86) Jones pour poison into the water main.

in the situation where Jones intentionally pours poison in the water main (to get rid of bedfish) without having the intention to poison the populace.

3.9.1. *Negation of event types.* It appears that the negation³⁶ of an event type can only marginally be an event type itself, as (perhaps) in Cooper's example (82). If it yields an event type at all, negation seems to produce an antonym rather than a classical negation. This observation has been made several times in a different context, that of perception verb complements. Higginbotham's example ([48]) is

(87) John saw Mary not smoke.

Insofar as this sentence read with narrow scope for *not* is grammatical, the *not* seems to turn *smoke* into an antonym, meaning something like 'refrain from smoking' (with its attendant pained grimaces).

This situation can also occur in our context. In general, however, there seems to exist some evidence that negation preferably turns an event type into a stative predicate³⁷. We shall first show that there is a way to introduce a nonclassical negation-like operation on event types, and then proceed to give an interpretation of negation which coerces event types into fluents.

Let the operation \sim denote a negation on event types. We may try to introduce \sim definitionally by the clause

$$(7) \quad \neg \text{Happens}(e, s) \rightarrow \text{Happens}(\sim e, s),$$

which upon completion yields

$$(8) \quad \neg \text{Happens}(e, t) \leftrightarrow \text{Happens}(\sim e, t).$$

However, from this one cannot conclude that \sim is anything like classical negation, because in the context of logic programming the negation on the left hand side of 7 and 8 is defined by negation as failure. It follows that there is no guarantee that

$$(9) \quad \forall t (\text{Happens}(e, t) \vee \text{Happens}(\sim e, t)),$$

³⁶An extensive discussion of negation related to nominalization is contained in Asher [3].

³⁷In Chapter 11 we have seen that in some sense negation turns an activity into a state.

so that $\sim e$ will in general be an antonym of e rather than a true classical negation of e .

It is also possible to define another kind of negation on event types, which transforms event types into fluents.

DEFINITION 49. *The fluent negation $\approx e$ of an event type e is defined by $\approx e := \neg \text{Happens}[e, \hat{t}] = \neg \text{that}(e) = \text{that}(\approx e)$.*

This type of negation will be important for coercion phenomena.

While equation 7 provides the bare outlines of a possible form of event negation, in practice a ‘localized’ version of 7 is more useful. To this end, consider first the expression ‘the arrival of the train’, represented by the event type e . In this case, e is literally the endpoint of a trajectory: that is, there are two fluents, an activity f_1 and a parametrized fluent f_2 connected by a dynamics, such that e satisfies a clause of the form

$$\text{HoldsAt}(f_1, t) \wedge \text{HoldsAt}(f_2(c), t) \rightarrow \text{Happens}(e, t).$$

In this case, the negation $\sim e$ is plausibly taken to be localized to the trajectory, and not global as in definition 7. For instance, one could say that as the long as the train is on its way but has not reached its destination, *non-arrival of the train* happens. Or rather, since the train may have been forced to stop in the middle of nowhere³⁸, non-arrival must be take relative to a time-window in which the train could have arrived, given its intended dynamics. Using the concepts of Section 4 of Chapter 11 we formulate this as

$$\neg \text{HoldsAt}(f_2(c), t) \wedge \text{HoldsAt}(\text{time}_{f_1}(a), t) \rightarrow \text{Happens}(\sim e, t).$$

To understand this formula, it is necessary to recall from Section 4 of Chapter 11 that the clock used in defining the fluent *time* stops ticking only when the culminating event has been reached, not when an arbitrary stop-event occurs. Thus, if a train is stopped without reaching its destination, the corresponding clock keeps on ticking forever, but in order to define the event $\sim e$ we need a bound, here represented by the constant a . The result is that, after completion, $\sim e$ happens when the train has started on its way, and for a limited time after it has been stopped before reaching its destination. Outside this interval, $\sim e$ does not happen. Of course, neither does e , so in this sense $\sim e$ is strongly antonymic.

Armed with these concepts, let us now analyze example (25) of Section 1, and some variants thereof:

- (88)
- a. The non-arrival of USS *Indianapolis* at Leyte caused consternation.
 - b. *The non-arrival of USS *Indianapolis* at Leyte unexpectedly . . .
 - c. The unexpected non-arrival of USS *Indianapolis* at Leyte caused consternation.
 - d. The fact that USS *Indianapolis* did not arrive at Leyte caused consternation.

³⁸Leaves on the track are the Dutch railways’ favourite excuse.

- e. USS *Indianapolis*' not arriving at Leyte caused consternation.
- f. USS *Indianapolis*' not arriving at Leyte quickly/?unexpectedly caused consternation.
- g. *The non-arrival of USS *Indianapolis* at Leyte occurred at noon, July 31, 1945.
- h. Every non-arrival of a ship causes consternation.
- i. (Google) Second, there was no directive to report the non-arrival of a combatant ship.

As we have seen in Section 1 the problem posed by data such as this is that the denotation of *non-arrival of USS Indianapolis* seems to be neither a fluent nor an event type. Sentences (88-b) and (88-c) seem to indicate that this denotation should be an event type, which must therefore be of the form $\sim e$, defined analogously to 'non-arrival of the train'. This is consistent with sentences (88-a), and (88-h) once one recalls that in the context of a wide container such as 'cause consternation', an event type is reinterpreted as a fluent. Cresswell argued that (his analogues of) (88-d) and (88-e) are good paraphrases of (88-a), which again seems to indicate that coercion from an event type to a fluent takes place. Google has several examples from Senate hearings and court-martial proceedings along the lines of (88-i), but their import depends on whether 'report' is used as a narrow container here; if it is not, *non-arrival of a combatant ship* could be both a fluent and an event type. The really problematic case is therefore (88-g); if *non-arrival of USS Indianapolis* is indeed an event type, why is it not compatible with a narrow container such as 'occur at noon'? Well, consider the following example

- (89) The non-arrival of USS *Indianapolis* at Leyte that I was telling you about didn't occur in 1942, but in 1945, just before the end of the war.

This sentence seems to be rather intelligible; so the unacceptability of (88-g) may be due to a finer distinction among narrow containers. In fact, 'event *d* occurs at *i*' seems to have the meaning that the temporal extent of *d* must be included in *i*. With the given definition of $\sim e$ this is impossible in the case of (88-g), but true in the case of (89).

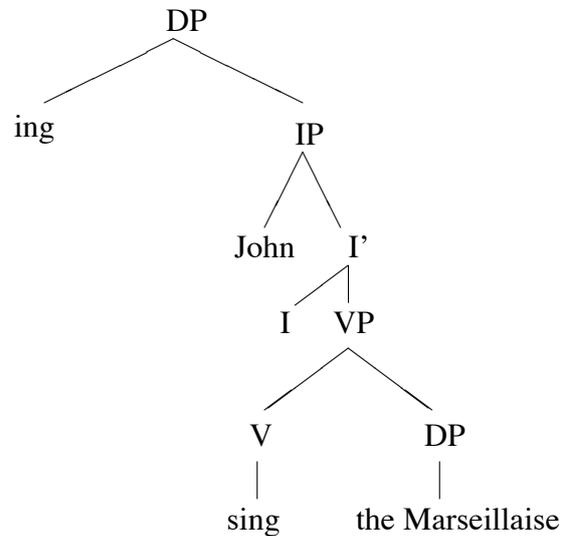
3.10. Exercises.

EXERCISE 21. Provide a formal definition of the determiner every for the case that the nuclear scope is a wide container.

EXERCISE 22. Consider the following examples from [14].

- (90) a. John runs.
 b. *John to run.
 c. John tries to run.
 d. *John tries runs.

Derive formal representations for these examples which explain their acceptability distribution.

FIGURE 1. Acc-*ing*

The following two exercises concern the syntax–semantic interface. The first is related to Abney’s analysis ([1]) the second to Pullum’s ([91]). We will first give a very brief introduction to Abney’s analysis.

Abney’s account is based on a conservative extension of classical \bar{X} -theory. It is conservative in the sense that it does not eliminate any inferences of \bar{X} -theory on the phrasal level. Abney’s approach differs from the classical theory only insofar as he assumes that the function of the affix *-ing* is to convert a verbal category into a nominal one. The essence of his analysis is then that the differences in the structures of the various types of English gerunds reduce to the question where in the projection path of the verb this conversion takes place. It is presumed that *-ing* can only be adjoined to the lexical category V and to maximal projections; i.e. VP and IP³⁹. If *-ing* is sister of IP the resulting structure is that of Acc-*ing*.

In case *-ing* is sister of the VP-node, we get in a similar way the structure of the Poss-*ing* gerund.

The third possibility is that *-ing* is sister to the lexical category V. In this case we have the structure of the Ing-*of* phrases.

It should be noted that *-ing* does nothing but convert a verbal projection into a nominal one. This abstract morphological element does not have a syntax of its own because it does not project any structure. This is the reason why Abney’s system is a conservative extension of classical \bar{X} -theory.

EXERCISE 23. *Derive a strictly compositional interpretation for Acc-Ing, Poss-Ing and Ing-of gerunds assuming Abney’s syntactic structures.*

³⁹A structure like $[_{CP}C[_{DP} -ing[_{IP} \dots]]]$ is excluded because it violates the selection properties of *C*.

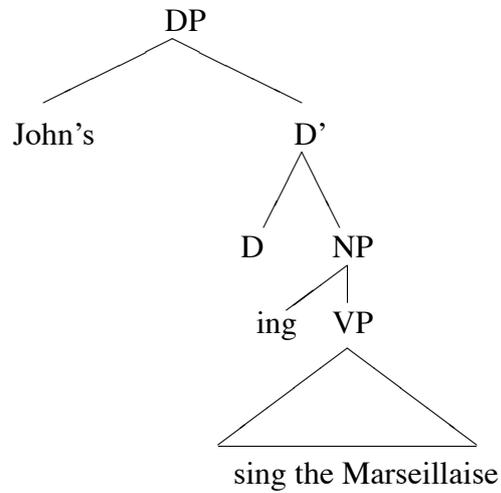


FIGURE 2. Poss-ing

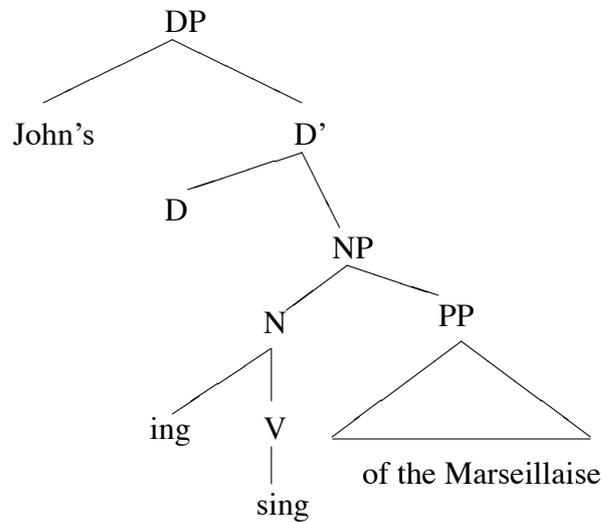


FIGURE 3. Ing-of

Hint: The semantic effect of -ing in Ing-of gerund is slightly different from that of -ing in Poss-ing and Acc-ing gerunds.

EXERCISE 24. *In Pullum [91] the following analysis of nominal gerund phrases (Abney's Poss-ing gerunds) is proposed. The feature [VFORM : prp] says that the verb is in its participle form. Similarly the feature [POSS :*

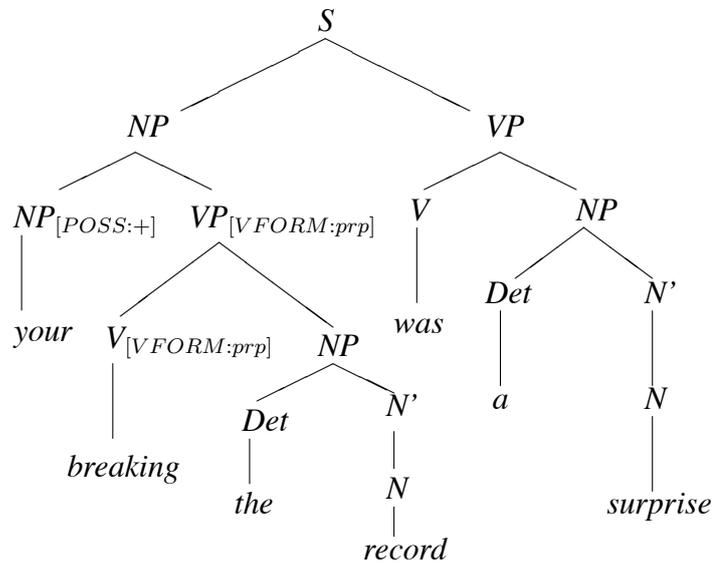


FIGURE 4. Pullum's NGP

+] indicates that the NP is a possessor NP. These forms are assumed to be derived in the morphological component of grammar and therefore only the results of such processes occur in syntax.

Derive a compositional interpretation for Pullum's syntactic analysis.

EXERCISE 25. Consider example (91). Show formally that sentence (91-c) follows from the premises (91-b) and (91-a).

- (91) a. The physician's revealing of the secret is unlikely.
 b. The physician's revealing of the secret took place yesterday.
 c. Something unlikely took place yesterday.

Hint: The verbal context *is unlikely* is a loose container. Therefore the perfect nominal in sentence (91-a) is coerced to an imperfect reading.

4. Nominalizations formalized II: Lexical meaning

In Section 3.2 we assumed that perfect nominals correspond to event types, and imperfect nominals to fluents, by applying the two forms of abstraction given by Feferman's coding machinery. For expository purposes we made a few shortcuts; for example in formalizing the perfect nominal *singing of 'A Poultry Tale'* the song was simply treated as an object of the verb. This was of course done against our better judgment, since we know from Chapter 7 that *sing 'A Poultry Tale'* is an accomplishment, so that 'A Poultry Tale' does not simply fill an argument slot of *sing*, but is related to the activity via a dynamics. A similar remark applies to the imperfect nominal *singing 'A Poultry Tale'*.

More generally, we have so far discussed only logical properties common to all (im)perfect nominals, and we have not considered the lexical meaning of the verb phrases from which the nominals derive. Now lexical content is determined by a scenario, and together with integrity constraints and the scenarios corresponding to other expressions, this determines a minimal model, in which the temporal profile of the (im)perfect nominal can be computed. This temporal profile may interact in complicated ways with the verbal context. The following two Sections intend to show how the abstract structural representations for perfect and imperfect nominals given in Section 3.2 can be enriched to take account of lexical content and *Aktionsart*.

4.1. Perfect nominals. The discussion will be focussed on the following example

- (92) During the morning rehearsals, every singing of ‘A Poultry Tale’ was interrupted.

The tricky point here is that the verb *interrupt* is a narrow container. For example, the following sentence is ungrammatical

- (93) *During the morning rehearsals, every singing ‘A Poultry Tale’ was interrupted.

Similarly, sentence (94-a) in contrast to the sentence (94-b) containing a perfect nominal is ungrammatical⁴⁰.

- (94) a. *John’s cooking the dinner meticulously was interrupted by a phone call.
b. John’s meticulous cooking of the dinner was interrupted by a phone call.

From a philosophical point of view this makes sense as well. Facts, results, propositions cannot be interrupted, but events or actions can. But the fact that this pattern obtains, and not the reverse, causes considerable difficulties. One might at first think that the event type corresponding to *singing of ‘A Poultry Tale’* can be introduced by means of *hierarchical planning* introduced in Chapter 3, on *Aktionsart*.

DEFINITION 50. *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 $Happens(start_f, s) \wedge s < r < t \wedge HoldsAt(f, r) \wedge Happens(finish_f, t) \rightarrow Happens(e, r)$*

The definition implies that in order for e to be nontrivial, the culminating event $finish_f$ must have occurred (and likewise for $start_f$). This formulation worked quite well when formalizing the past tense, where one needs to

⁴⁰Thanks to Orrin Percus for an insightful discussion of these examples.

make a distinction between ‘John crossed the street’, and ‘John was crossing the street’. We would like to use it here in the following manner: as in Section 3.4 one first constructs the event type $\exists t.sing[x, p, t]$ and then determines its temporal profile by means of definition 50, where one takes for the scenario that indicated for an accomplishment. Unfortunately, example (92) shows that hierarchical planning cannot be used for perfect nominalization in quite this way, because the culminating event need not have been reached.

To arrive at a better formalization, recall what a scenario for the accomplishment *sing* ‘A Poultry Tale’ looks like. There must be a fluent $sing(x)$ (where x denotes subject position), a parametrized fluent $p(y)$ (which denotes a stage of the song ‘A Poultry Tale’, indexed by the real number y), an initiating event type $start_{sing(x)}$, and a culminating event type $finish_{sing(x)}$. What sentence (92) shows is that the event type $\exists t.sing(x, p)$ associated to the perfect nominal must be defined without using $finish_{sing(x)}$. It would have been pleasant if the events used for defining tense were the same as those used in perfect nominalization, but apparently there is a slight difference.

We then get instead of definition 50

DEFINITION 51. *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 $Happens(start_f, s) \wedge s < r < t \wedge HoldsAt(f, r) \rightarrow Happens(e, r)$*

This construction allows the event to be interrupted before the culmination point is reached.

As before we take for the event e the Feferman code $\exists t.sing[x, p, t]$. As in definition 46 we associate event tokens (u, v) to $\exists t.sing[x, p, t]$. For simplicity, assume a purely temporal meaning of ‘be interrupted’ as ‘not reach its natural culmination point’. Recalling the definition of the universal quantifier, we may then formalize the relevant part of (92) for instance as the integrity constraint

$?Happens(\exists t.sing[x, p, t], (u, v)), Happens(finish_{sing(x)}, v), v < now$ fails.

Observe that this integrity constraint would make no sense if the formula $Happens(\exists t.sing[x, p, t], (u, v))$ would be determined in accordance with definition 50.

4.2. Imperfect nominals. We now turn to the formalization of imperfect nominals, taking into account the lexical content and the *Aktionsart* of the VP one starts from. In this case, nominalizing states and activities (in the strict sense) is straightforward, since they correspond to single fluents. Now consider the accomplishment *sing* ‘A Poultry Tale’, and the associated imperfect nominal *Deborah’s singing* ‘A Poultry Tale’. We have to find a fluent that corresponds to the latter expression. In one sense this is easy:

use the Feferman code $sing[d, p, \hat{t}]$. The more difficult problem is to ensure that this fluent gets the right temporal profile in minimal models; and this entails somehow connecting the fluent to the scenario for the accomplishment, which we introduced above in Section 4.1. As an example of the problems one runs into here: the temporal profile $sing[d, p, \hat{t}]$ can be different from that of the activity $sing$ itself: imagine that said Deborah alternates singing ‘A Poultry Tale’ and ‘Casta Diva’, then we are only interested in the intervals in which she sings ‘A Poultry Tale’. This means that, in the notation introduced in Section 4.1, the parametrized fluent $p(y)$ must somehow play a role in determining the temporal profile of $sing[d, p, \hat{t}]$.

One way to think of the required construction is via the concept of *incremental theme* introduced by Dowty [32], which in the case at hand is a relation between the fluents $sing$ and $p(y)$. The concept “incremental theme” applies to telic predicates, i.e. accomplishments and achievements. The incremental theme relates the activity parts of such predicates to their result parts. Now $p(y)$ is a fluent which is nondecreasing in the sense of the integrity constraint

$$? HoldsAt(p(y), s), HoldsAt(p(y'), t), s < t, y' < y \text{ fails.}$$

It is equally clear that the intervals during which Deborah sings ‘A Poultry Tale’ correspond to intervals on which $p(y)$ is strictly increasing. If p were a simple function of t and not a fluent, it would be clear (at least to a mathematically inclined reader) what to do: take those intervals at which the derivative of p is strictly positive. The reader must take our word for it that this idea can be adapted to the present context, and that we can formally define a fluent *increasing* which takes a parametrized fluent as an argument and holds on those intervals on which, intuitively speaking, the parametrized fluent increases strictly⁴¹.

In the context of the scenario for $sing$ ‘A Poultry Tale’ the temporal profile of the fluent $sing[d, p, \hat{t}]$ is then determined by the clause

$$HoldsAt(increasing(p), s) \rightarrow HoldsAt(sing[d, p, \hat{t}], s).$$

We have implicitly extended the concept of *definition* from event types to fluents here, but there is no harm in doing so. This concludes our discussion on how to incorporate lexical content in nominalization.

4.3. Exercises.

EXERCISE 26. We used hierarchical planning to provide lexical content for perfect nominals in Section 4.1. Write programs which define the events $start_f$ and $finish_f$ in terms of the basic predicates of the event calculus.

Hint: $start_f$ should be an event which when happening initiates fluent f (similarly for $finish_f$).

⁴¹Moschovakis [84, p. 12] does something similar to formalize the sentence ‘The temperature rises’.

Derived nominals such as *arrival of the train*, *destruction of the city* show a much less systematic behavior than Ing-of gerunds⁴². This was one of the reasons why Chomsky excluded them from a syntactic analysis in his *Remarks on nominalizations* ([15]). Often it is quite idiosyncratic how the meaning of the nominalization is related to the meaning of the verbs it is derived from. For example there seems to be no significant general pattern that forms the basis of nominalizations like *construction* in the *Anglo-Saxon genitive construction* and *revolution* in the *French revolution*. The relation between *construct* and *construction* and *revolve* and *revolution* in these cases clearly differs considerably⁴³. But although many derived nominals are highly ambiguous some of them have the eventive reading described for Ing-of gerunds among their meanings. For example *destruction of the city* has both a resultative meaning and an eventive reading. This aspect of the meaning of *destruction of the city* will therefore be analyzed in the following way:

(95) $\exists t. \text{destroy}[x, c, t]$

Other types of nominalizations however don't have any of the readings discussed here, for instance *referee* or *amusement*⁴⁴.

In Paragraph 48 of his *Elements of Symbolic Logic* [94], Reichenbach correctly observes that the following sentences have the same truth conditions

(96) Amundsen flew to the North Pole in May 1926.

(97) A flight by Amundsen to the North Pole took place in May 1926.

Here, *flight* is the nominal derived from *fly*. Sentence (96) is an example of *thing splitting*, whereas sentence (97) is an example of *event splitting*⁴⁵.

EXERCISE 27. *Formalize (96) and (97) and prove that there are equivalent.*

⁴²Comrie and Thompson [19] is a survey of lexical nominalization patterns in the languages of the world.

⁴³See Scalise [98] for a more thorough discussion of this topic.

⁴⁴See Spencer [105] for an overview of theories dealing with these kinds of nominalization.

⁴⁵Reichenbach uses the term *splitting* because he thinks that the predicate-subject form of a sentence splits the situation it describes into a part corresponding to the predicate and a thing-part corresponding to the subject.

Appendix: the basics of logic programming

First-order predicate logic, especially when combined with coding tricks such as detailed in Chapter 6, is a very expressive language, but as a consequence it suffers from undecidability. ‘Logic programming’ refers to a family of programming languages, including Prolog and Constraint Logic Programming, which exploit the existence of tractable fragments of predicate logic. Very roughly speaking, in logic programming one considers only so-called *program clauses*, that is, formulas of the form $\varphi \rightarrow A$, where φ can be arbitrary, but A must be atomic. This restriction allows very efficient proof search, since here a single derivation rule (called *resolution*) suffices. The language still retains remarkable expressive power, since all computable functions can be defined using formulas of this form. Furthermore, derivations in logic programming are simultaneously computations, in the following sense: if one derives that a formula $\psi(x)$ is satisfiable given a set of clauses, the derivation actually produces a computable witness for x .

However, it would be misleading to emphasize only logic programming’s computational efficiency as a cleverly chosen fragment of predicate logic. For our purposes, the semantics of logic programming is also highly relevant, both with regard to the type of models and the meaning of the logical operators.

A comparison with predicate logic may help to explain the first point. For the usual derivation systems, such as natural deduction \mathcal{ND} , we have the following

THEOREM 7. *If for a set of sentences Γ and a sentence ψ , $\Gamma \models \psi$, then $\Gamma \vdash_{\mathcal{ND}} \psi$.*

COROLLARY 5. *If a set of sentences Σ is consistent in \mathcal{ND} , then it has a model.*

However, the model witnessing consistency is constructed using set theoretic techniques, and need not be computable. But if Σ consists of program clauses, then it does have computable models, which can be constructed directly from the input Σ . This is important for us, because we believe that discourse understanding proceeds via the construction of models for the discourse, and so the process had better be uniformly and efficiently computable. Moreover, as explained in Stenning and van Lambalgen [115], it is in principle possible to perform these computations on neural nets. We will also see that these models are *partial* in the sense that for some predicates

A and tuple of elements \bar{d} , it may be undecided whether or not $A(\bar{d})$, and hence the meaning of the logical operators changes as well. More precisely, it will be seen that \wedge, \vee, \neg are truthfunctional in three-valued logic, whereas \rightarrow receives a different meaning, no longer truthfunctional. It is claimed in Stenning and van Lambalgen [115] that the \rightarrow of logic programming is an excellent candidate for formalizing conditionals expressing defaults, and we have put this machinery to work here because we believe that most if not all meaning-relationships have default character.

1. Logic programming for propositional logic

On the syntactic side, the basic format of logic programming is the following: given a logic program P , determine whether an atomic formula A can be derived from P using a single rule called *resolution*. We first discuss what a program is in a simple case, and we explain what resolution means here.

1.1. Positive programs.

DEFINITION 52. A positive clause is a formula of the form $p_1, \dots, p_n \rightarrow q$, where the q, p_i are propositional variables; the antecedent may be empty. In this formula, q is called the head, and p_1, \dots, p_n the body of the clause. A positive program is a finite set of positive clauses.

The second important ingredient of logic programming is the *query*.

DEFINITION 53. A query is a finite (possibly empty) sequence of atomic formulas denoted as $?p_1, \dots, p_m$. Alternatively, a query is called a goal. The empty query, canonically denoted by \square , is interpreted as \perp , i.e. a contradiction.

Operationally, one should think of a query $?q$ as the assumption of the formula $\neg q$, the first step in proving q from P using a reductio ad absurdum argument. In other words, one tries to show that¹ $P, \neg q \models \perp$. In this context, one rule suffices for this, a rule which reduces a goal to subgoals.

DEFINITION 54. Unit-resolution is a derivation rule which takes as input a program clause $p_1, \dots, p_n \rightarrow q$ and a query $?q$ and produces the query $?p_1, \dots, p_n$.

The name *unit*-resolution derives from the fact that one of the inputs is an atomic formula. Since we will be concerned with this case only, we refer to the derivation rule simply as *resolution*. A derivation starting from a query $?A$ can be pictured as a tree as in figure 1.

Resolution is complete for the chosen fragment in the following sense:

THEOREM 8. Let P be a positive program, A an atomic formula. Then $P \models A$ if and only if the empty query can be derived from $?A$ using P . If the right hand side of the equivalence holds, we say that the derivation is successful.

¹We use \top for an arbitrary tautology, and \perp for an arbitrary contradiction.

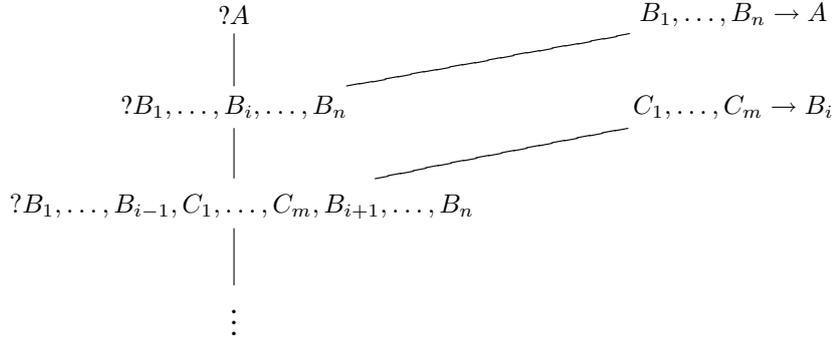


FIGURE 1. An illustration of a derivation with unit resolution.

1.1.1. *Closed world reasoning.* Let P be a positive program, and suppose that A is an atomic formula such that there is no successful derivation from $?A$. As a concrete case, suppose that P is a public transportation database, and that the query $?A$ is the question whether there is a train from Amsterdam to Tübingen on February 17, 2004, leaving around 9.00am. In this case the query is not successful, and we want to conclude from this that there is no such train. This is an instance of *closed world reasoning*: if given the data there is no reason to assume that A is true, one may assume it is false. The completeness theorem does not sanction this inference, since it only gives $P \not\models A$, instead of the stronger $P \models \neg A$. To get the stronger inference, we must restrict the class of models considered.

1.1.2. *Semantics for closed world reasoning.* We assume that propositions are either true (1) or false (0), but the required semantics is nevertheless nonclassical. The only models to be considered are those of the following form

DEFINITION 55. Let P be a positive program on a finite set of proposition letters L . An assignment \mathcal{M} of truthvalues $\{0, 1\}$ to L is a model of P if for $q \in L$,

- (1) $\mathcal{M}(q) = 1$ if there is a clause $p_1, \dots, p_n \rightarrow q$ in P such that for all i , $\mathcal{M}(p_i) = 1$
- (2) $\mathcal{M}(q) = 0$ if for all clauses $p_1, \dots, p_n \rightarrow q$ in P there is some p_i for which $\mathcal{M}(p_i) = 0$.

The definition entails that for q not occurring as the head of a clause, $\mathcal{M}(q) = 0$. More generally, the model \mathcal{M} is minimal in the sense that a proposition not forced to be true by the program is false in \mathcal{M} . Thus, supposing that there is no successful derivation of $?A$ from P , the completeness theorem gives $P \not\models A$, which means that on the models \mathcal{M} considered here, A is actually false.

A very important feature of models of a positive logic program P , as defined above, is that they can be constructed iteratively. More formally, they are given by the fixed points of a monotone operator:

DEFINITION 56. *The operator T_P associated to P transforms an assignment \mathcal{V} (identified with the set of proposition letters made true by \mathcal{V}) into a model $T_P(\mathcal{V})$ according to the following stipulations: if u is a proposition letter,*

- (1) $T_P(\mathcal{V})(u) = 1$ if there exists a set of proposition letters C , made true by \mathcal{V} , such that $\bigwedge C \rightarrow u \in P$
- (2) $T_P(\mathcal{M})(u) = 0$ otherwise.

DEFINITION 57. *An ordering \subseteq on assignments \mathcal{V}, \mathcal{W} is given by: $\mathcal{V} \subseteq \mathcal{W}$ if all proposition letters true in \mathcal{V} are true in \mathcal{W} .*

LEMMA 6. *If P is a positive logic program, T_P is monotone in the sense that $\mathcal{V} \subseteq \mathcal{W}$ implies $T_P(\mathcal{V}) \subseteq T_P(\mathcal{W})$.*

Monotonicity would fail if a body of a clause in P contains a negated atom $\neg q$ and also a clause $\neg q \rightarrow s$: one can then set up things in such a way that s is true at first, and becomes false later.

Monotonicity is important because it implies the existence of so called fixed points of the operator T_P .

DEFINITION 58. *A fixed point of T_P is an assignment \mathcal{V} such that $T_P(\mathcal{V}) = \mathcal{V}$.*

LEMMA 7. *If T_P is monotone, it has a least and a greatest fixed point.*

LEMMA 8. *A fixed point is model in the sense of definition 55. Every such model is also a fixed point.*

1.1.3. *Reformulation as a classical semantics.* The semantics given in the preceding Section is nonclassical in that only a very special subclass of the structures satisfying P are considered. Fortunately there is a trick which allows us to reintroduce a fully classical semantics for the case of positive programs. The trick consists in applying a syntactic operation to P .

DEFINITION 59. *Let P be a positive program.*

a. *The completion of a positive program P is given by the following procedure:*

- (1) take all clauses $\varphi_i \rightarrow q$ whose head is q and form the expression $\bigvee_i \varphi_i \rightarrow q$
- (2) if q does not occur as a head, introduce the clause $\perp \rightarrow q$
- (3) replace the implications (\rightarrow) by bi-implications (\leftrightarrow) (here, \leftrightarrow is semantically interpreted by $\mathcal{V}(\psi \leftrightarrow \varphi) = 1$ iff $\mathcal{V}(\psi) = \mathcal{V}(\varphi)$, and 0 otherwise)
- (4) take the conjunction of the (finitely many) sentences thus obtained; this gives the completion of P , which will be denoted by $\text{comp}(P)$.

b. *If P is a positive logic program, define the non-monotonic consequence relation \approx by*

$$P \approx \varphi \text{ iff } \text{comp}(P) \models \varphi.$$

If $P \models \varphi$, we say that φ follows from P by closed world reasoning. The process of completion is also referred to as minimisation.

LEMMA 9. Let P be a positive program, and $\text{comp}(P)$ its completion. Then $\mathcal{M} \models \text{comp}(P)$ if and only if \mathcal{M} is a model in the sense of definition 55.

1.2. Definite programs and negation as failure. Positive programs, when aided by the closed world assumption, allow one to derive negative conclusions. They do not yet allow one to handle negative conditions. For this one needs an extension of the preceding definitions which permits the occurrence of negation in the body of a clause.

DEFINITION 60. A body is a formula of the form $L_1 \wedge \dots \wedge L_m$, where each L_i is an atomic formula or a negation of an atomic formula (including \top or \perp). (Such L_i are called literals.) A (definite) clause is a formula of the form $\varphi \rightarrow q$, where φ is a body.

DEFINITION 61. A general query is a finite sequence $?L_1, \dots, L_m$, where L_i is a literal.

Occurrences of negation are handled in a derivation by means of an additional rule known as ‘negation as (finite) failure’. Suppose we are given a definite program P consisting of definite clauses, and a general query $?A_1, \dots, A_n, \neg B_1, \dots, \neg B_k$, where the A_i, B_j are positive. We have to define the action of P on the query, and to investigate what it means semantically. A formally precise definition of negation as failure would be too complex, so we give a definition that works for our case. An excellent reference for a fuller treatment is [28].

As before, the goal of a derivation is to derive the empty clause. Thus, we have to have a mechanism for erasing a literal from a query. For positive literals we can still use unit resolution. For a negative literal $\neg B_j$, we start a derivation beginning with the query B_j . If the resulting resolution tree is finite, but does not have the empty clause at one of its end nodes, then we say that the query B_j fails finitely. In this case, we may erase $\neg B_j$ from the query $?A_1, \dots, A_n, \neg B_1, \dots, \neg B_k$. If the resulting resolution tree does have the empty clause at one of its end nodes, then the query B_j is successful. In this case the query $\neg B_j$ fails, and therefore also the query $?A_1, \dots, A_n, \neg B_1, \dots, \neg B_k$. The attentive reader will have noticed that the above attempt at a definition really hides an inductive characterisation, since the derivation tree starting from the query B_j may itself involve applications of negation as failure. This may lead to loops in the derivation, for example when $P = \{\neg A \rightarrow A\}$ and the query is $?A$.

1.2.1. *Semantics for negation as failure.* As observed above, extending the definition of the operator T_P with the classical definition of negation would destroy its monotonicity, necessary for the incremental approach to the least fixed point. One solution is to replace the classical two-valued logic by a particular form of many-valued logic, Kleene’s three-valued logic.

		p	q	$p \wedge q$	p	q	$p \vee q$
		1	1	1	1	1	1
		0	0	0	0	0	0
p	$\neg p$	u	u	u	u	u	u
1	0	1	0	0	1	0	1
0	1	1	u	u	1	u	1
u	u	0	1	0	0	1	1
		0	u	0	0	u	u
		u	1	u	u	1	1
		u	0	0	u	0	u

FIGURE 2. Three-valued connectives

This logic has truth values $\{u, 0, 1\}$ with the partial order $u \leq 0$ and $u \leq 1$. Here, u is *not* a degree of truth, but rather means that the truth value is, so far, undecided. Thus, u is not a value in between 0 and 1, as it would be in Łukasiewicz' three-valued logic. The chosen ordering reflects the intuition that u can 'evolve' toward 0 or 1 as a result of computation—running the program finds determinate values for variables whenever possible, though some indeterminate values may remain. The truth tables given in figure 2 (taken from [62]) are then immediate, as readers should satisfy themselves.

In addition we define an equivalence \leftrightarrow by assigning 1 to $\varphi \leftrightarrow \psi$ if φ, ψ have the same truth value (in $\{u, 0, 1\}$), and 0 otherwise.

Suppose we are given a *definite* logic program P . We show how to construct models for such programs, as fixed points of a three-valued consequence operator \mathcal{T}_P^3 . We will drop the superscript when there is no danger of confusion with its two-valued relative defined above.

DEFINITION 62. *A three-valued model is an assignment of the truth values $u, 0, 1$ to the set of proposition letters. If the assignment does not use the value u , the model is called two-valued. If \mathcal{M}, \mathcal{N} are models, the relation $\mathcal{M} \leq \mathcal{N}$ means that the truth value of a proposition letter p in \mathcal{M} is less than or equal to the truth value of p in \mathcal{N} in the canonical ordering on $u, 0, 1$.*

DEFINITION 63. *Let P be a definite program.*

- a. The operator \mathcal{T}_P applied to formulas constructed using only \neg, \wedge and \vee is determined by the above truth tables.*
- b. Given a three-valued model \mathcal{M} , $\mathcal{T}_P(\mathcal{M})$ is the model determined by*
 - (a) $\mathcal{T}_P(\mathcal{M})(q) = 1$ iff there is a clause $\varphi \rightarrow q$ such that $\mathcal{M} \models \varphi$*
 - (b) $\mathcal{T}_P(\mathcal{M})(q) = 0$ iff for all clauses $\varphi \rightarrow q$ in P , $\mathcal{M} \models \neg\varphi$*

LEMMA 10. *If P is a definite logic program, \mathcal{T}_P is monotone in the sense that $\mathcal{M} \leq \mathcal{N}$ implies $\mathcal{T}_P(\mathcal{M}) \leq \mathcal{T}_P(\mathcal{N})$.*

LEMMA 11. *Let P be a program.*

- a. \mathcal{M} is a model of the $\text{comp}(P)$ iff it is a fixed point of T_P .
- b. The least fixed point of T_P exists and is reached in finitely many steps ($n + 1$ if the program consists of n clauses). The least fixed point of T_P^3 will be called the minimal model of P .

In this context, the non-monotonic consequence relation $P \approx_3 \varphi$ (see definition 59) is given by ‘ $\text{comp}(P) \models_3 \varphi$ ’, or in words: all (three-valued) models of $\text{comp}(P)$ satisfy φ . If $P \approx_3 \varphi$, we say that φ follows from P by negation as failure. As before, this relation is indeed non-monotonic, since in general, if P' extends P , the class of models of $\text{comp}(P')$ will be larger than that of $\text{comp}(P)$. Note that the relation \approx is completely determined by what happens on the least fixed point. Larger fixed points differ in that some values u in the least fixed point have been changed to 0 or 1 in the larger fixed point; but by the monotonicity property (with respect to truth values) of Kleene’s logic this has no effect on the output unit pairs, in the sense that an output value 1 cannot be changed into 0 (or conversely).

The soundness and completeness theorem appropriate for this context then reads

THEOREM 9. *Let a definite program P be given, and let A be an atomic formula.*

- (1) *There is a successful derivation starting from $?A$ if and only if $P \approx_3 A$.*
- (2) *The query $?A$ fails finitely if and only if $P \approx_3 \neg A$.*

1.2.2. *A further extension.* Above we have defined definite clauses starting from *bodies* defined as $L_1 \wedge \dots \wedge L_m$. A glance at the operational definition of negation as failure shows that more complex formulas can be handled as well.

DEFINITION 64. a. *A complex body is defined recursively as follows.*

- (1) *An atomic formula is a complex body.*
- (2) *If $\varphi_1, \dots, \varphi_k$ are complex bodies, so is $\varphi_1 \wedge \dots \wedge \varphi_k$.*
- (3) *If φ is a complex body, so is $\neg\varphi$.*
- b. *A normal clause is a formula $\varphi \rightarrow A$, where A is atomic and φ is a complex body.*
- c. *A normal program is a finite set of normal clauses.*

The definition implies that \forall may occur in the body of a program clause if it is taken to be defined via \wedge and \neg .

2. Logic programming for predicate logic

We now consider the case where the underlying language is first order predicate logic. We will explain logic programming for predicate logic only in so far as is necessary to understand the derivations given in the body of the book. Since the semantics of constraint logic programming,

our preferred tool, is different from standard logic programming, we will not bother to give the semantics of the latter here.

The logical form of first order logic programs is the same as that for propositional logic, whether we consider positive or definite programs. Literals will of course contain variables, and we assume that each clause is implicitly universally quantified, although we will in general omit the quantifiers. The presence of variables and individual constants requires one however to redefine the resolution rule². Here are two examples to illustrate the issues. Let P be the positive program $\{B(x) \rightarrow A(x), B(a)\}$. Suppose first the query is $?A(a)$. In this case we should have a successful derivation, because indeed $A(a)$ follows from $\forall x(B(x) \rightarrow A(x)) \wedge B(a)$. Intuitively, such a successful derivation can be constructed by specializing x to a . Recall that $B(x) \rightarrow A(x)$ is read as universally quantified, so that specialization to given value is valid. Syntactically, specialization takes the form of substituting a for x . In figure 3 below, the required substitution is indicated in the query. The derivation presented in figure 3 is an application of the

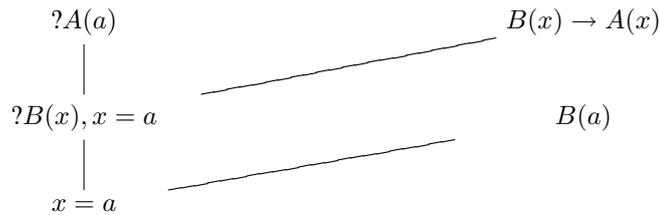


FIGURE 3. Resolution for predicate logic

following form of resolution.

DEFINITION 65. *Let P be a definite program.*

- a. *A query is a formula of the form $?L_1, \dots, L_k, e_1, \dots, e_m$ where the L_i are literals and the e_j are equations between terms.*
- b. *Let $?L_1, \dots, L_k, e_1, \dots, e_m$ be a query, and suppose L_i is a positive literal, $L_i = L_i(t_1, \dots, t_l)$, such that $K_1 \wedge \dots \wedge K_n \rightarrow L_i(x_1, \dots, x_l)$ is in P . Via resolution, one may then derive the new query*
 $?L_1, \dots, L_{i-1}, K_1, \dots, K_n, L_{i+1}, \dots, L_k, x_1 = t_1, \dots, x_l = t_l, e_1, \dots, e_m$.
- c. *A derivation is successful if its last line consists of equations between terms only.*

Let y_1, \dots, y_v be the variables occurring in the top query $? \varphi$. If the derivation from $? \varphi$ is successful, the equations $y_1 = s_1, \dots, y_v = s_v$ derived from the bottom query are jointly called the computed answer substitution for y_1, \dots, y_v in $? \varphi$.

²Our presentation is slightly nonstandard, and is motivated by constraint logic programming as used in the body of the book.

Figure 4 illustrates these notions. The bottom query consists of the equations $x = y, x = a$. We are interested in the computed answer substitution for y . The equations imply $y = a$, which is the required substitution. Definition 65 hides a subtlety, especially in its definition of successful

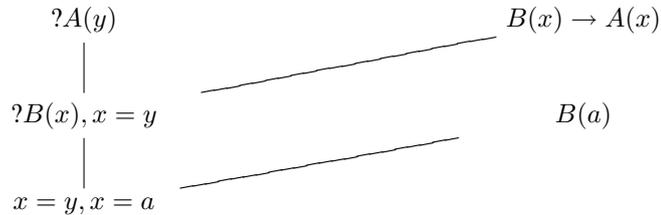


FIGURE 4. Extracting a witness

derivation. One has to exercise some care here, because of the occurrence of the subsidiary derivations (using negation as finite failure) for queries of the form $\neg B_k$. What is intended here, is that one only considers substitutions that arise from resolution applied to a positive literal in a query. The idea is that such a substitution is used to instantiate B_k , and that the instance thus obtained is tested for failure. Further instantiations used in that test are not taken into account.

For our purposes, the most important metatheorem is soundness:

THEOREM 10. *Let the definite program P be given. Suppose a derivation from $?\varphi(y_1, \dots, y_v)$ ends successfully with a set of equations which jointly imply $y_1 = s_1, \dots, y_v = s_v$. Then $P \models_3 \varphi(s_1, \dots, s_v)$.*

Bibliography

- [1] S. Abney. *The English Noun Phrase in its Sentential Aspect*. PhD thesis, Massachusetts Institute of Technology, 1987.
- [2] N. Asher. A default, truth conditional semantics for the progressive. *Linguistics and Philosophy*, 15:463–508, 1992.
- [3] N. Asher. *Reference to Abstract Objects in Discourse*. Kluwer, Dordrecht, 1993.
- [4] N. Asher and A. Lascarides. *Logics of conversation*. Cambridge University Press, Cambridge, 2003.
- [5] A. d' Avila Garcez, K.B. Broda, and D. Gabbay. *Neural-symbolic learning systems: foundations and applications*. Springer, London, 2002.
- [6] E. Bach. The algebra of events. *Linguistics and Philosophy*, 9:5–16, 1986.
- [7] R.A. Berman and D.I. Slobin (eds.). *Relating events in narrative*. Lawrence Erlbaum, Hillsdale, N.J., 1994.
- [8] R.A. Block. *Cognitive models of psychological time*. Lawrence Erlbaum, 1990.
- [9] A. Bonomi. The progressive and the structure of events. *Journal of Semantics*, 14:173–205, 1997.
- [10] G. Boolos and R.E. Jeffrey. *Computability and Logic*. Cambridge University Press, Cambridge, 1989.
- [11] J.D. Bransford, J.R. Banks, and J.J. Franks. Sentence memory: a constructive versus an interpretive approach. *Cognitive Psychology*, 3:193–209, 1972.
- [12] J. Bybee, R. Perkins, and W. Pagliuca. *The evolution of grammar*. Chicago University Press, Chicago, Ill., 1994.
- [13] M. Callaway. The appositive participles in anglo-saxon. *Publications of the Modern Language Association of America*, 16, 1901.
- [14] G. Chierchia. *Topics in the Syntax and Semantics of Infinitives and Gerunds*. Garland Publishing, Inc., New York, 1988.
- [15] N. Chomsky. Remarks on nominalization. In A. Jacobs and P. Rosenbaum, editors, *Readings in Transformational Grammar*, pages 184–221. Ginn and co., Waltham (Mass), 1970.
- [16] N. Cocchiarella. *Logical Investigations of Predication Theory and the Problem of Universals*. Bibliopolis, Napoli, 1986.
- [17] B. Comrie. *Aspect*. Cambridge University Press, Cambridge, 1976.
- [18] B. Comrie. *Tense*. Cambridge University Press, Cambridge, 1985.
- [19] B. Comrie and S. Thompson. Lexical nominalization. In T. Shopen, editor, *Language Typology and Syntactic Description, Vol. III, Grammatical Categories and the Lexicon*. Cambridge University Press, Cambridge, 1985.
- [20] M. Cresswell. Interval semantics for some event expressions. In R. Bäuerle, U. Egli, and A. von Stechow, editors, *Semantics from Different Points of*

- View. Springer, Berlin, 1979.
- [21] W Croft. The structure of events. In M. Tomasello, editor, *The new psychology of language*, chapter 3. Lawrence Erlbaum Associates, Mahwah, NJ, 1998.
- [22] G.O. Curme. History of the english gerund. *Englische Studien*, 65, 1912.
- [23] D. Davidson. The logical form of action sentences. In N. Rescher, editor, *The Logic of Decision and Action*. University of Pittsburgh Press, Pittsburgh, 1967.
- [24] N. Davies. *The Isles: A History*. Oxford University Press, New York, 1999.
- [25] H. de Swart. Aspect shift and coercion. *Natural language and linguistic theory*, 16:347–385, 1998.
- [26] H. de Swart. Aspect shift and coercion. *Natural Language and Linguistic Theory*, 16:347–385, 1998.
- [27] H. de Swart and F. Corblin (eds). *Handbook of French Semantics*. CSLI Publications, February 2002. NWO-CNRS PICS project.
- [28] K. Doets. *From logic to logic programming*. The M.I.T. Press, Cambridge, MA, 1994.
- [29] J. Dölling. Polysemy and sort coercion in semantic representation. In P. Bosch and P. Gerstl, editors, *Discourse and lexical meaning*. Arbeitspapiere des SFB 340, Nr. 30, Stuttgart/Tübingen, 1993.
- [30] D. Dowty. *Word meaning and Montague grammar*. Reidel, Dordrecht, 1979.
- [32] D. Dowty. Thematic proto-roles and argument selection. *Language*, 67:547–619, 1991.
- [33] R. Eckardt, editor. *Adverbs, Events, and Other Things*. Niemeyer, Tübingen, 1998.
- [34] M. Egg. Flexible semantic construction: the case of reinterpretation. Habilitationsschrift, 2000.
- [35] E. Einenkel. Zur geschichte des englischen gerundiums. *Anglia*, 37, 1913.
- [36] E. Einenkel. Die entwicklung des englischen participium praesentia auf –ing. *Anglia*, 38, 1914.
- [37] J. Emonds. The derived nominals, gerunds, and participles in chaucer’s english. In B. Kachru, R. Lees, Y. Malkiel, A. Pietrangeli, and S. Saporta, editors, *Issues in Linguistics*, pages 186–198. University of Illinois, Champaign, 1973.
- [38] S. Feferman. Toward useful type-free theories i. *The Journal of Symbolic Logic*, 49:75–111, 1984.
- [39] B. Fennell. *A History of English*. Blackwell, Oxford, 2001.
- [40] G. Frege. On sense and denotation. In P. Geach and M. Black, editors, *Translations from the Philosophical Writings of Gottlob Frege*. Blackwell, Oxford, 1952.
- [41] G. Frege. Sinn und Bedeutung. In G. Patzig, editor, *G. Frege: Funktion, Begriff, Bedeutung. Fünf logische Studien*. Vandenhoeck, Göttingen, 1962.
- [42] M. C. Geis and A. M. Zwicky. On invited inferences. *Linguistic Enquiry*, 2:561–566, 1971.
- [43] J. Gibson. *The ecological approach to visual perception*. Houghton-Mifflin Co., Boston, 1979.
- [44] J. Goldsmith and E. Woisetschlaeger. The logic of the english progressive. *Linguistic Inquiry*, 13:79–89, 1982.

- [45] L. Gosselin. *Sémantique de la temporalité en français*. Champs Linguistiques. Editions Duculot, 1996.
- [46] P.M. Greenfield. Language, tools and the brain: the ontogeny and phylogeny of hierarchically organized sequential behavior. *Behavioral and brain sciences*, 14:531–595, 1991.
- [47] F. Hamm and M. van Lambalgen. Event calculus, nominalization, and the progressive. *Linguistics and Philosophy*, 26:381–458, 2003.
- [48] J. Higginbotham. The logic of perceptual reports: An extensional alternative to situation semantics. *Journal of Philosophy*, LXXX:100–127, 1983.
- [49] D. Hindsill. On the turning verbs into nouns. Msc thesis, ILLC, Amsterdam, 2001. Available electronically via <http://www.illc.uva.nl>.
- [50] W. Hodges. *Model theory*. Cambridge University Press, Cambridge, 1993.
- [51] W. Hodges. Formal features of compositionality. *Journal of Logic, Language, and Information*, 10:7–28, 2001.
- [52] P. Hopper. Aspect and foregrounding in discourse. In T. Givón, editor, *Syntax and Semantics: Discourse and Syntax*. Academic Press, New York, 1979.
- [53] A. Houston. The english gerund: Syntactic change and discourse function. In R. Fasold and D. Schiffrin, editors, *Language Change and Variation*. Benjamins, Amsterdam, 1989.
- [54] B. Irwin. *The Development of the ING ending of the Verbal Noun and the Present Participle from c.700 – c.1400*. PhD thesis, University of Wisconsin, 1967.
- [55] W. James. *The principles of psychology*. Harvard University Press, 1983. Reprint of the 1890 edition.
- [56] O. Jespersen. *Essentials of English Grammar*. Allen & Unwin, London, 1933.
- [57] M.R. Jones and M. Boltz. Dynamic attending and responses to time. *Psychological Review*, 96:459–491, 1989.
- [58] H. Kamp. Events, instants and temporal reference. In R. Baeuerle, U. Egli, and A. von Stechow, editors, *Semantics from different points of view*, pages 27–54. Springer Verlag, Berlin, 1979.
- [59] H. Kamp and C. Rohrer. unpublished progress-report for research on tenses and temporal adverbs of French, 200?
- [60] H. Kamp and C. Rohrer. Progress report on tense and temporal adverbs in french. Report, Institut fuer maschinelle Sprachverarbeitung, Universitaet Stuttgart, Unpublished.
- [61] E. Keenan. Passive in the world’s languages. In T. Shopen, editor, *Language typology and syntactic description. Volume I: Clause structure*. Cambridge University Press, Cambridge, 1985.
- [62] S. C. Kleene. *Introduction to Metamathematics*. North-Holland, Amsterdam, 1951.
- [63] W. Koehler. *The mentality of apes*. Harcourt Brace and World, New York, 1925.
- [64] R. A. Kowalski and M. Sergot. A logic-based calculus of events. *New Generation Computing*, 4:65–97, 1986.
- [65] R.A. Kowalski. Using meta-logic to reconcile reactive with rational agents. In *Meta-logics and logic programming*, pages 227–242. MIT Press, 1995.

- [66] M. Krifka. *Nominalreferenz und Zeitkonstitution*. Fink, München, 1989.
- [67] F. Landman. The progressive. *Natural Language Semantics*, 1:1–32, 1992.
- [68] A. Lascarides and N. Asher. Temporal interpretation, discourse relations and commonsense entailment. *Linguistics and Philosophy*, 16(5):437–493, 1993.
- [69] P. Laserson. *Plurality, Conjunction and Events*. Kluwer, Dordrecht, 1995.
- [70] R. Le Poidevin. *Travels in four dimensions – the enigmas of space and time*. Oxford University Press, Oxford, 2003.
- [71] R. (ed.) Le Poidevin. *Questions of time and tense*. Clarendon Press, Oxford, 1998.
- [72] D. Lewis. Scorekeeping in a language game. *Journal of Philosophical Logic*, 8:339–359, 1979.
- [73] G. Link. Algebraic semantics for event structures. In J. Groenendijk, M. Stokhof, and F. Veltman, editors, *Proceedings of the Sixth Amsterdam Colloquium*, 1987.
- [74] K. Mani and P.N. Johnson-Laird. The mental representation of spatial descriptions. *Memory and cognition*, 10:181–187, 1982.
- [75] D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W.H. Freeman, San Francisco, 1982.
- [76] B. McGonigle, M. Chalmers, and A. Dickinson. Concurrent disjoint and reciprocal classification by *cebus apella* in serial ordering tasks: evidence for hierarchical organization. *Animal Cognition*, In press.
- [77] F.T. Melges. Identity and temporal perspective. In R.A. Block, editor, *Cognitive models of psychological time*, pages 37–58. Lawrence Erlbaum, 1990.
- [78] L. Michaelis. A unification-based model of aspectual type-shifting. University of Colorado at Boulder; see <http://spot.colorado.edu/michaeli/>, 2001.
- [79] J.A. Michon. Implicit and explicit representations of time. In R.A. Block, editor, *Cognitive models of psychological time*, pages 37–58. Lawrence Erlbaum, 1990.
- [80] G.A. Miller and P.N. Johnson-Laird. *Language and Perception*. Harvard University Press, 1976.
- [81] M. Moens and M. Steedman. Temporal ontology and temporal reference. *Comput. Ling.*, 14:15–28, 1988.
- [82] Y. Moschovakis. Sense and denotation as algorithm and value. In J. Oikkonen and J. Väänänen, editors, *Lecture Notes in Logic*. Association for Symbolic Logic, A.K. Peters, Ltd., Natick, Mass., 1993.
- [83] Y. Moschovakis. What is an algorithm. In B. Engquist and W. Schmid, editors, *Mathematics Unlimited – 2001 and beyond*. Springer, Berlin, 2001.
- [84] Y.N. Moschovakis. A logical calculus of meaning and synonymy. *Submitted.*, 2003.
- [85] R. Naumann and Ch. Piñón. Decomposing the progressive. In P. Dekker, M. Stokhof, and Y. Venema, editors, *Proceedings of the 11th Amsterdam Colloquium*. University of Amsterdam, Amsterdam, 1997.
- [86] F. Nauze and M. van Lambalgen. Serious computing with tense. In *Computational Semantics 3*. Kluwer, Dordrecht and Boston, 2004.
- [87] T. Parsons. *Events in the Semantics of English*. MIT University Press, Cambridge MA, 1990.

- [88] E. Poppel. A hierarchical model of temporal perception. *Trends in Cognitive Sciences*, 1:56–60, 1997.
- [89] P. Portner. *Situation Theory and the Semantics of Propositional Expressions*. PhD thesis, University of Massachusetts at Amherst, 1991.
- [90] H. Poutsma. *The Infinitive, the Gerund and the Particles of the English Verb*. Noordhoff, Groningen, 1923.
- [91] G. Pullum. English nominal gerund phrases as noun phrases with verb-phrase heads. *Linguistics*, 29:763–799, 1991.
- [92] S. Pulman. Aspectual shift as type coercion. *Transaction of the Philological Society*, 95:279–317, 1997.
- [93] J. Pustejovsky. Type coercion and lexical selection. In J. Pustejovsky, editor, *Semantics and the lexicon*. Kluwer, Dordrecht, 1993.
- [94] H. Reichenbach. *Elements of Symbolic Logic*. Macmillan, London, 1947.
- [95] B. Russell. *Our knowledge of the external world*. Allen and Unwin, London, 1914.
- [96] B. Russell. On order in time. *Proc. Camb. Phil. Soc.*, 32:216–228, 1936.
- [97] E. S. Savage-Rumbaugh. *Kanzi, the ape at the brink of the human mind*. Doubleday, London, 1994.
- [98] S. Scalise. *Generative Morphology*. Foris, Dordrecht, 1984.
- [99] P. Schachter. A nontransformational account of gerundive nominals in english. *Linguistic Inquiry*, 7:1–55, 1976.
- [100] M. Shanahan. Representing continuous change in the event calculus. In *Proceedings ECAI 90*, pages 598–603, 1990.
- [101] M. Shanahan. A circumscriptive calculus of events. *Artificial Intelligence*, 77:249–287, 1995.
- [102] M. Shanahan. A logical account of the common sense informatic situation for a mobile robot. London, 1996.
- [103] M. Shanahan. Robotics and the common sense informatic situation. In *Working Notes of Common Sense 96, The Third Symposium on Logical Formalisation of Commonsense*, pages 186–198, 1996.
- [104] M.P. Shanahan. *Solving the frame problem*. The M.I.T. Press, Cambridge MA, 1997.
- [105] A. Spencer. *Morphological Theory*. Blackwell, Oxford, 1991.
- [106] R.F. Staerk. From logic programs to inductive definitions. In W. Hodges, M. Hyland, C. Steinhorn, and J. Truss, editors, *Logic: from foundations to applications (European Logic Colloquium '93)*, pages 453–481. Oxford University Press, Oxford, 1996.
- [107] L. Stassen. *Intransitive Predication*. Clarendon Press, Oxford, 1997.
- [108] M. Steedman. *The productions of time*. Manuscript, 1997. Available at <http://www.cogsci.ed.ac.uk/~steedman>.
- [109] M. Steedman. Temporality. In J. van Benthem and A. ter Meulen, editors, *Handbook of Logic and Language*, chapter 16, pages 895–938. Elsevier Science, 1997.
- [110] M. Steedman. Plans, affordances and combinatory grammar. *Linguistics and Philosophy*, 26, 2003.
- [111] H. Sten. *Les temps du verbe fini (indicatif) en français moderne*. Historik-filologiske Meddelelser. Det Kongelige Danske Videnskabernes Selskab, 1952.

- [112] K. Stenning. Anaphora as an approach to pragmatics. In M. Halle, J. Brennan, and G.A. Miller, editors, *Linguistic theory and psychological reality*. MIT Press, 1978.
- [113] K. Stenning. On making models: a study of constructive memory. In T. Myers, K. Brown, and B. McGonigle, editors, *Reasoning and discourse processes*. Academic Press, 1986.
- [114] K. Stenning. *How did we get here? A question about human cognitive evolution*. Amsterdam University Press, 2003.
- [115] K. Stenning and M. van Lambalgen. A working memory model of relations between interpretation and reasoning. 2003. Submitted to *Cognitive Science*.
- [116] K. Stenning and M. van Lambalgen. *Human reasoning and cognitive science*. MIT University Press, Cambridge, MA., 2004.
- [117] P.J. Stuckey. Negation and constraint logic programming. *Information and Computation*, 118:12–33, 1995.
- [118] T. Suddendorf and M. Corballis. Mental time travel and the evolution of mind. *Genetic, social and general psychology monographs*, 123:133–167, 1997.
- [119] S. K. Thomason. On constructing instants from events. *Journal of Philosophical Logic*, 13:85–96, 1984.
- [120] S. K. Thomason. Free construction of time from events. *Journal of Philosophical Logic*, 18:43–67, 1989.
- [121] T. Trabasso and N.L. Stein. Using goal-plan knowledge to merge the past with the present and future in narrating events on line. In M.H. Haith, J.B. Benson, R.J. Roberts, and B.F. Pennington, editors, *The development of future-oriented processes*, pages 323–352. University of Chicago Press, 1994.
- [122] A.S. Troelstra and D. van Dalen. *Constructivism in mathematics. An Introduction*. Elsevier, Amsterdam, 1988. Studies in Logic Vols. 121 and 123.
- [123] J. van der Does and M. van Lambalgen. A logic of vision. *Linguistics and Philosophy*, 23(1):1–92, 2000.
- [124] M. van Lambalgen and F. Hamm. Intensionality and coercion. In R. Kahle, editor, *Intensionality*. ASL Lecture Notes in Logic, A.K. Peters, 2003.
- [125] M. van Lambalgen and F. Hamm. Moschovakis' notion of meaning as applied to linguistics. In M. Baaz and J. Krajicek, editors, *Logic Colloquium '01*. ASL Lecture Notes in Logic, A.K. Peters, 2003.
- [126] Z. Vendler. *Linguistics in Philosophy*. Cornell University Press, Ithaca, New York, 1967.
- [127] Z. Vendler. *Adjectives and Nominalizations*. Mouton, The Hague, 1968.
- [128] H. Verkuyl. *A Theory of Aspectuality. The interaction between Temporal and Atemporal Structure*. Cambridge University Press, Cambridge, 1993.
- [129] A. G. Walker. Durées et instants. *Revue Scientifique*, 85:131–134, 1947.
- [130] D. Westerståhl. Quantifiers in formal and natural languages. In D. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic, Vol IV*. Reidel, Dordrecht, 1989.
- [131] G.J. Whitrow. *The natural philosophy of time*. Clarendon Press, Oxford, second edition, 1980.
- [132] G.J. Whitrow. *What is time?* Clarendon Press, Oxford, 2003. Abridged edition of [131].

- [133] J. Wrobel. *Language and schizophrenia*. John Benjamins, Amsterdam, 1989.
- [134] J.M. Zacks and B. Tversky. Event structure in perception and cognition. *Psychological Bulletin*, 127(1):3–21, 2001.
- [135] S. Zucchi. Incomplete events, intensionality and imperfective aspect. *Natural Language Semantics*, 7:179–215, 1999.