Tears in the Rain

Tomas By
Department of Computer Science
Sheffield University
T.By@dcs.shef.ac.uk

February 2002

Submitted for the degree of Doctor of Philosophy
All those moments will be lost in time, like tears in rain.

Blade Runner (1982)
Contents

Introduction 1

1 Time in discourse 2
   1.1 The example text 4
   1.2 Output example 5

2 Time, tense, and aspect in English 7
   2.1 Morphology and syntax of verb groups 7
      2.1.1 Verb morphology 7
      2.1.2 Auxiliaries for aspect and voice 8
      2.1.3 Modal auxiliaries 9
      2.1.4 Operators and inversion 9
      2.1.5 Classification of verb groups 9
      2.1.6 Structure of verb groups 10
      2.1.7 List of verb groups 11
   2.2 Temporal meaning of verb groups 13
      2.2.1 Finite verb groups without modal verbs 13
      2.2.2 Finite verb groups with modal verbs 15
      2.2.3 Tense-less phrases 16
      2.2.4 Shifts in tense and mood 19
   2.3 Time adjuncts 19
      2.3.1 Units of time 20
      2.3.2 Location and duration 22
      2.3.3 Containment 23
      2.3.4 Combination with perfect aspect 24
      2.3.5 Repetition and frequency 25
   2.4 Aspectual expressions 25

3 Theoretical foundations of tense and aspect 27
   3.1 Time points 27
   3.2 Situation types 29
      3.2.1 Mass nouns and count nouns 30
      3.2.2 Stative versus dynamic 31
      3.2.3 Exception: stance 31
      3.2.4 Situations with a conclusion 32
      3.2.5 Task performance and duration 33
3.2.6 Dynamic situations ........................................ 34
3.2.7 Activities / accomplishments / achievements ........ 35
3.3 Type-shifting ....................................................... 36
  3.3.1 Event nucleus and transitions ......................... 38
  3.3.2 Generative lexicon ........................................ 40
3.4 Formal representation of time ................................ 42

4 Beyond the sentence .............................................. 46
  4.1 The example text ............................................ 47
  4.2 Time points ..................................................... 47
  4.3 Narrative progression ........................................ 48
  4.4 Discourse representations .................................. 49
    4.4.1 Verb phrases and time points ....................... 49
    4.4.2 Discourse interpretation algorithm ................. 51
    4.4.3 The example ............................................. 52
    4.4.4 Details ................................................ 53
    4.4.5 Summary ............................................... 56
  4.5 Discourse grammar ........................................... 57
    4.5.1 The example ............................................ 59
    4.5.2 Details ................................................ 60
    4.5.3 Summary ............................................... 62
  4.6 Partitioned representations ................................ 62
    4.6.1 Space builders ......................................... 64
    4.6.2 Descriptions ........................................... 65
    4.6.3 Time spaces ............................................ 68
    4.6.4 Verb groups ............................................ 69
    4.6.5 The example ............................................ 70
  4.7 Segments and stack ........................................... 73
    4.7.1 Tense trees ............................................ 73
    4.7.2 Time points ............................................ 74
    4.7.3 Situation types ......................................... 75
    4.7.4 The example ............................................ 75
    4.7.5 Details ................................................ 77
    4.7.6 Summary ............................................... 80
  4.8 Summary ....................................................... 82

5 Formal syntax of verb groups ................................ 86
6  **Formal semantics of time, tense and aspect**  
6.1 States and events ........................................ 93  
6.2 Tense and aspect ........................................ 95  
6.3 Internal time ........................................... 97  
6.4 Lexical semantics ....................................... 97  
   6.4.1 Main verbs ........................................ 98  
   6.4.2 Verb particles .................................... 98  
   6.4.3 Aspectual verbs ................................... 99  
   6.4.4 Progressive aspect ................................ 100  
   6.4.5 Nominalizations ................................... 100  
   6.4.6 Eventive nouns ................................... 101  
6.5 Perfect aspect .......................................... 102  
6.6 Formal representation of time ....................... 104  

7  **Interpretation of tense and aspect in discourse**  
7.1 External time .......................................... 105  
7.2 Simple sentences ..................................... 109  
7.3 Complex sentences .................................... 109  
7.4 Details of the example ................................ 112  

Conclusions ................................................. 120  

Further work .............................................. 122  

Summary .................................................... 128  

Acknowledgements ........................................... 129  

Glossary ..................................................... 130  

Appendix ..................................................... 131  
   A.1 Grammar .............................................. 133  
   A.2 Lexicon ............................................... 139  
   A.3 Meaning representation ............................. 146  
   A.4 Semantic interpretation ............................ 155  
   A.5 Segmentation ........................................ 167  
   A.6 Discourse interpretation ........................... 176  

References ................................................ 186  

Index ....................................................... 191
Introduction

Of the countless problems preventing computational understanding of natural language, the most central, perhaps, is the interpretation of verb phrases. Together with noun phrases, they are the major syntactic constituent, at least in Indo-European languages, and they are usually seen as the main part of sentences. While opinions differ about the details, and there are exceptions such as nominalizations, there is general agreement about the typical meaning of noun phrases, namely reference to objects in the world, and to abstract concepts. For verb phrases, the situation is the reverse. Those, like the copula, whose meaning is simple, from the representational point of view, are exceptions, and there is no widely agreed-on formal interpretation of the majority of verbs, which refer to situations occurring in time.

Much of the blame for this surely belongs to the influence of mathematical logic, where time has no importance and verbs are seen only as relations between individuals, the things referred to by noun phrases. Even in those theories of meaning representation that translate ordinary verbs into complex expressions built up from primitives, the temporal part of their meaning has often been ignored.

While it may well be that other components of language meaning have rightfully been given higher priority than the temporal, they will all have to be considered in a full computational theory of language. Substituting one tense for another in a sentence can make a great difference in meaning, and in a discourse the problem is greater since the meaning of a given tense is relative to those used in earlier sentences.

Describing the constructions in English related to time, the computational theories of them that have been proposed, and suggesting some improvements on these, are the topics of this thesis.

The first chapter formulates the technical problems addressed, introduces an example text that will be used throughout the thesis, and shows the suggested computational analysis of the first sentence of this text. Chapter two is a comprehensive description of the ways to express temporal information in English. A selection of the most relevant literature, in philosophy, linguistics, and computer science, on temporal phenomena, are reviewed in chapters three and four. The final three chapters contain my own treatment of: verb group syntax, temporal semantics, and the interpretation of time in discourse.
1 Time in discourse

Temporal information in natural language is mostly expressed by tense, aspect, and time adjuncts. Among the languages of the world the lexical and syntactic forms, as well as the specific meanings, of tenses and aspects vary greatly, and the linguistic literature on the topic is immense. Unfortunately, from the point of view of computational linguistics, most of these studies are not immediately useful, since they do not descend into the abyss of technical minutiae to the levels required for computer programming.

The possible combinations of tenses and aspects are many, and the range of meaning they can express is wide. In discourse, time interpretation is further complicated by the context dependence of both tenses, and many time adjuncts. It is possible, through the use of calendar dates and clock times, to indicate a time with little dependence on the immediate discourse context, but time statements in language are normally relative to the previous text.

Broadly speaking, four major issues can be distinguished.

- Verbs differ in the temporal profile of their meanings. Extreme examples are ‘sit,’ which refers to a state that holds without change for an extended period, versus ‘nod,’ which involves continuous movement throughout its short duration. The term used here for these classes is situation type (Quirk, Greenbaum, Leech and Svartvik, 1985, p. 177) and they are discussed in section 3.2 and, from a more technical point of view, in section 6.1.

  In different linguistic contexts, the same verb sense can have different situation type. This phenomenon is called type-shifting and is influenced by, among other things, grammatical subject and objects, verb particles, and time adjuncts (see section 3.3 and page 150).

- Some of the items under consideration here, good examples in English being aspectual verbs and progressive aspect, serve to select a part of the temporal extent of the verb they modify. ‘Begin to sit,’ for example, refers to a point in time and not the whole extended period of sitting. Following Connolly (1976, p. 5), this will be known as internal temporal meaning, as opposed to external, discussed below. The technical details are presented in sections 6.2 and 6.3.

- External time is the typical meaning of tense and perfect aspect: evoking temporal locations in the past or the future where the situations that are mentioned take place.

  Quite often, these times are under-specified. Normal past tense, for example, only tells us that the time is earlier than ‘now,’ but the exact location is not identified. In discourse, a time introduced in this manner can co-refer with times used earlier, similar to pronoun anaphora. More discussion of this can be found in section 7.1.

- In complex sentences, the various verbs can be temporally connected, by prepositions or conjunctions for example, but they can also be temporally independent. The term used here for these unconnected sections of time is segment (see section 7.3).
Very little of the work needed to make computers understand language can reasonably be said to have been completed. There are few, if any, useful programming tools; no complete formal grammar exists for any language; there is no consensus on fundamental issues like precisely what information, and in what format, needs to be in the lexicon. A consequence of this is that for any problem that is interesting from the linguistic point of view, the researcher either has to write tedious amounts of uninteresting code to provide the higher level input, such as sentence meaning representations, that he needs, or he can proceed from some assumed format, although no code exists to compute it, which means the work is based on a fantasy.

Here, the first option has been chosen, and a complete system has been implemented,\(^1\) that reads a discourse, sentence by sentence, identifies objects and events, and keeps track of the temporal information. It supports all the major time-related constructions in English, including all possible syntactic forms of tense and aspect, and a wide, but possibly incomplete, range of meanings for them. Temporal prepositions and conjunctions are handled, as are aspectual verbs and verb particles.

Of the problems listed on the previous page, the models of tense, aspect, and temporal meaning that have been proposed in the computational literature mainly concern external time, and, to some extent, situation types. Internal time has been largely ignored, and temporal segments do not seem to be recognised by any theory other than the one proposed here, which handles all four.

\[\text{(Input)} \quad \text{Situation type ontology} \quad \text{(Output)}\]

```
Lexical semantics of main verbs & nominalizations

Lexical semantics / Verb particles / Aspectual verbs

Verb group meaning

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\rightarrow

\righta
In the system described in this thesis, the input is first parsed by a Prolog definite clause grammar into a logicist form, where logical variables are used to connect the expressions. Next, these are interpreted and symbolic times, eventualities, entities, and temporal relations are identified. The first two are derived mainly from verb phrases, entities correspond roughly to noun phrases, and the relations are typically encoded by prepositions and conjunctions. Finally, the data is added, segment by segment, to a representation of the whole discourse. At this stage it is crucial to find all existing entities, eventualities, and times that co-refer with the new ones.\(^4\) There might not be any, \textit{e.g.}, if it is the first sentence in the discourse, but normally at least some times and entities can be expected to have been mentioned previously.

Although the system is complete, in the sense that it takes real sentences as input, some shortcuts have been made. The lexicon is limited to those words that occur in the example that is introduced in the next section, and a few more chiefly from closed classes. Other data, such as the concept catalogue, only contains what is needed to support the lexicon. The temporal meaning that the system computes is restricted to time points and their order. Duration and repetition, for example, is not included in the semantic representation.

Compared with other systems for interpreting temporal information in discourse that are described in the literature (see chapter 4), the one presented here makes more extensive use of basic techniques from computational linguistics and knowledge representation, and handles a wider range of temporally related phenomena. The set of possible meanings for verb groups, including auxiliaries, in particular, is greatly simplified by most other approaches. Here, a table\(^5\) provides a simple and clear implementation of the many-to-many mapping between verb group form and meaning, divided into internal and external, as mentioned earlier. Perfect aspect (see section 6.5) is also more completely interpreted, and a number of more esoteric constructions, such as nominalizations, eventive noun phrases, and non-finite verb phrase complements, which are ignored by most other systems, are supported.

### 1.1 The example text

The following extract from W.S. Maugham's \textit{Of Human Bondage}, which is quoted by Reichenbach (1947, p. 288), is used as an example throughout the thesis.

> But Philip ceased to think of her a moment after he had settled down in his carriage. He thought only of the future. He had written to Mrs. Otter, the maîtresse to whom Hayward had given him an introduction, and had in his pocket an invitation to tea on the following day. When he arrived in Paris he had his luggage put on a cab and trundled off slowly through the gay streets, over the bridge, and along the narrow ways of the Latin Quarter. He had taken a room at the Hotel des Deux Ecoles, which was in a shabby street off the Boulevard du Montparnasse; it was convenient for Amirasano's School at which he was going to work.

\(^4\)For entities, this problem is known as 'anaphora resolution.'

\(^5\)See page 106 and the procedure \texttt{inf1/3} in appendix A.5.
There are several examples of contextual time specifications in this text: the
adjunct 'on the following day' depends on the identity of the current day, the
conjunctions 'a moment after' and 'when' relates two events temporally, and the
perfect forms 'had written' and 'had taken' indicate that those events took place
before the situation directly described by the narrative. Most of the sentences
are in simple past tense, and the events are by default understood to have
occurred in the same order as they are described.

1.2 Output example

The output produced for the first sentence in the example is shown below. Three
time symbols are used to represent the temporal information: ‘t1’ is when the
settling down takes place, ‘t0’ is the cessation of thinking, and ‘now’ is when
the utterance is made. Between the first two, one moment (x3) passes. Other
than that and the order, which is indicated by the order of the symbols after
the ‘times:’ label, there is no quantitative information about these times. The
columns of eventuality symbols (e.g. e2, e3, e4) under the time symbols indicate
that those eventualities take place at the time in the first row (e.g. t1).

[Philip ceased to think of her a moment after.
he had settled down in his carriage]

sentence has 1 segment(s)

segment 1: 1 interpretation(s)
[past] + [past.past]
times: t1 t0 now
e2 e0
e3 e1

e4
duration from t1 to t0 is x3
e0: activity [concept=contemplation,agent=x1,object=x0]
e1: transition(e0,"e0" []
e2: transition(x1"location=x4,"x1"location=x4) [...]
e3: activity [concept=movement,agent=x1,object=x1,target=x2,source=x4]
e4: transition("x1"location=x2.x1"location=x2) [...]

x0: entity [is_a=female]
x1: entity [name=Philip.is_a=male]
x2: entity [poss_bv=x1.is_a=vehicle]
x3: entity [is_a=moment]
x4: entity [is_a=location]

Five events are represented. ‘Cease to think’ refers to a change, with no duration,
from thinking to not thinking, and is encoded as an end point (e1) of a
thinking activity (e0).

e0: activity [concept=contemplation,agent=x1,object=x0]
e1: transition(e0,"e0" []

Each of these (e0, e1) is an instance of a situation type.6 The ‘concept’ at-
tribute provides a link to the concept database (appendix A.3), but it is not

6The situation type primitives are listed in the table on page 95.
actually used. The other two attributes, ‘agent’ and ‘object,’ are thematic roles. The exact assignments of these are not particularly relevant here and can easily be changed, so they will not be discussed further.

The expression ‘transition(e0, e0)’ is intended to mean that the eventuality ‘e0’ ceases to be the case. In the present implementation, durations are not used; all eventualities are either instantaneous or take one unit of time and in the latter case cannot overlap, so explicitly indicating that an eventuality ends is redundant. But it seems desirable nevertheless to represent all the information that is stated in the sentence and adding support for duration and overlapping intervals is an obvious topic for further work (see page 123).

The other sentence, ‘Philip settled down in his carriage,’ is interpreted as a complete movement from one location to another (e2, e3, e4).

\[\begin{align*}
e2 &: \text{transition}(x1\text{-}location=x4, x1\text{-}location=x4) \\
e3 &: \text{activity} \quad [\text{concept}=\text{movement}, \text{agent}=x1, \text{object}=x1, \text{target}=x2, \text{source}=x4] \\
e4 &: \text{transition}(x1\text{-}location=x2, x1\text{-}location=x2) \quad [\ldots]
\end{align*}\]

\[\begin{align*}
x1 &: \text{entity} \quad [\text{name}=\text{Philip}, \text{is\_a}=\text{male}] \\
x2 &: \text{entity} \quad [\text{poss\_by}=x1, \text{is\_a}=\text{vehicle}] \\
x4 &: \text{entity} \quad [\text{is\_a}=\text{location}]
\end{align*}\]

The ellipsed attribute lists are identical to the one for ‘e3,’ since all three eventualities originate from the same verb phrase. Here, ‘transition(‘x1’ location=x2, x1\text{-}location=x2)’ means that ‘x1’ goes from not (‘) having the location ‘x2,’ which is the carriage, to having it. The symbol ‘x4’ is automatically generated (i.e., it is only implied in the text) and refers to the place where Philip is before he settles down in the carriage. Again, the particular representation used is not crucial; it has been chosen somewhat arbitrarily and it can be changed with little trouble.

In summary, what the computational analysis above tells us is that Philip (x1) moved from an otherwise unknown location (x4) to his carriage (x2) in one unit of time, which is what the ‘activity’ (e3) takes. The ‘transitions’ (e2, e4) are instantaneous. One moment (x3) later, he thinks for one unit of time (e0) and then stops (e1). All this takes place in the past.
2 Time, tense, and aspect in English

It is hard to know if a solution is correct without a clear grasp of the factual properties of the problem. In order to find good computer representations of temporal information in language, it is necessary to know the full range of meaning that can be expressed, and designing the analysis code requires knowledge of the possible surface forms. This chapter aims to provide these things, relying mainly on Quirk et al. (1985).

Tense and aspect are often taken to be semantic terms, which, at least in English, allows some scope for confusion and terminological disagreement, since many expressions contain elements of both meanings. A more utilitarian method is to say that tense is a morphological property of verbs and aspect a syntactical property of verb phrases (Quirk et al., 1985, p. 189). It follows from this that English has two tenses: past and present, and two aspects: perfect and progressive. Those constructions that are often called 'future tenses' are all realised by modal verbs, and they are, for that reason, not categorised as tenses here.

2.1 Morphology and syntax of verb groups

According to standard terminology (Quirk et al., 1985), the heads of verb phrases are main verbs and the modifiers are auxiliary verbs. A verb that has all the morphological forms and can only be main verb, is known as a full verb.

The class of auxiliary verbs is not exclusive; it is hard to find a verb that can never be a main verb. Broadly speaking, the auxiliaries can be divided into two groups. The first are those that serve to express aspect and voice. The second group are the modal verbs.

2.1.1 Verb morphology

The morphological forms of English verbs are: the base form, the finite forms and the two participles. Finite forms have three morphologically encoded properties: tense (past or present), person and number. The other three forms do not have this. For most verbs many of these forms have the same realisation so there are usually only three, four or five different word tokens. An exception is 'be' which has eight.

<table>
<thead>
<tr>
<th>BASE FORM</th>
<th>FULL VERBS</th>
<th>MODAL</th>
<th>be</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>call</td>
<td>—</td>
<td>be</td>
<td></td>
</tr>
<tr>
<td></td>
<td>speak</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>can</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>am/are/is</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FINITE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>called</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>spoke</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>could</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>was/were</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FINITE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>calling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>speaking</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>being</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NON-FINITE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>called</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>spoken</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>been</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NON-FINITE</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The base form can be both finite, in subjunctive and imperative groups, and non-finite, in infinitives.
2.1.2 Auxiliaries for aspect and voice

Perfect, progressive and passive are all composed of an auxiliary verb followed by a participle. Perfect is the simplest, being always created with the auxiliary verb 'have.'

The eagle has landed.

The progressive forms are normally constructed with the auxiliary 'be.'

The professor types his own letters. (a permanent habit)
The professor is typing his own letters. (while his secretary is ill)

Although they are not generally (Quirk et al., 1985) considered instances of progressive form, the aspectual verbs can have a similar syntactic function.

The professor started typing his own letters. (when she fell ill)
The professor stopped typing his own letters. (when she came back)
The professor kept typing his own letters. (after she came back)

The auxiliary 'be' is also typically used to form the passive.

Joseph wrote the book. (ACTIVE)
The book was written by Joseph. (PASSIVE)

But in some cases, 'get' can be used as well (Quirk et al., 1985, p. 161).

John got caught by the police.

It would also seem that other copular verbs, such as 'become' and can serve as passive auxiliary.

Her vacillation incensed me. (ACTIVE)
I became incensed at her vacillation. (PASSIVE)

The following table sums up the auxiliary verbs available for marking aspect and voice.

<table>
<thead>
<tr>
<th>PERFECT</th>
<th>have</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROGRESSIVE</td>
<td>be, begin, finish, start, stop, keep, continue, keep on, stay, give up, resume</td>
</tr>
<tr>
<td>PASSIVE</td>
<td>be, get, become</td>
</tr>
</tbody>
</table>

Both 'be' and 'have' can also function as main verbs, and there are similarities in meaning between this case and the progressive and perfect forms.

When 'have' is a main verb it indicates possession which, like the perfect forms, expresses a state.

I have a cold. (state of me being ill)
They have arrived. (state of them being here)

The main verb 'be' links together two entities, or an entity and a description. As Reichenbach (1947, p. 292) points out, the present participle, which is used in progressive aspect, is similar to adjectives syntactically, since both combine
with 'be,' and semantically, indicating a state or at least a situation that lasts longer than what the simple tense of the same verb would indicate.

He produces. SIMPLE PRESENT
He is producing. PRESENT PROGRESSIVE
He is productive. Copula + Adjective

In this example, from Reichenbach (1947, p. 292), the same core meaning is expressed by the verbs and the adjective, but the latter two forms indicate a more permanent state of affairs than the first.

2.1.3 Modal auxiliaries

The modal auxiliaries include the modal verbs, which do not usually occur as main verbs, some other verbs like 'dare' and 'need,' called marginal modals in Quirk et al. (1985), and also certain idiomatic expressions.

| MODAL VERBS | can, may, will, shall, could, might, would, should, must |
| Marginal modals | dare, need, ought to, used to |
| Idioms | be to, be about to, be going to, have to, have got to |

Since modal verbs are always finite, they can only occur in finite verb groups, and then only at the beginning. Most, if not all, modals can also be main verbs, however, in which case they behave like any other main verb.

2.1.4 Operators and inversion

If the first word in a verb group is an auxiliary, it can, in questions and negative constructions, change places with the subject of the sentence. This is called inversion, and the auxiliary is said to be an operator. In case the verb group has no auxiliary, the semantically content-less 'do' is used as operator.

*Can you believe it?* Inversion, question.
*Do you believe him?* Inversion, question.
*She could not find it.* Operator, negation.
*Never did he lose hope.* Inversion, negation.
*Where has she gone?* Inversion, question.
*I do hope they survived.* Emphatic positive.
*I do not hope they failed.* Operator, negation.
*Did he not win?* Inversion, negation, question.

Since inversion, questions, and negation are not closely connected to time, tense, or aspect, they will not be discussed further.

2.1.5 Classification of verb groups

In an English verb group the main verb comes last. Since there can be only one of them, all the preceding verbs in the group must be auxiliaries. If the first verb (main or auxiliary) is finite then the group is finite, otherwise it is
non-finite. A finite verb can never be in any other position in the verb group than first (Quirk et al., 1985, pp. 149–150).

There are two types of verb groups that do not fit perfectly into this categorisation: imperatives and subjunctives. They both use the base form of the verb and, like the non-finite groups, do not exhibit tense, person or number distinctions or combine with modal auxiliaries. A possible exception is the past subjunctive, which only exists for the verb be.

If that be the case, we must take action. PRESENT SUBJUNCTIVE
If that were the case, I would be surprised. PAST SUBJUNCTIVE

But the difference between the two forms of the subjunctive has less to do with tense than with degree of reality, i.e., mood. The past subjunctive indicates a hypothetical or counterfactual situation.

Since the imperatives and subjunctives, in contrast to the non-finite verb groups, can act as the verb element of independent clauses, Quirk et al. (1985, pp. 149–150) classes them as different moods of finite verb group. Only indicatives, which are finite, exhibit distinctions of tense, person, and number.

<table>
<thead>
<tr>
<th>INDICATIVE (FINITE)</th>
<th>He smokes.</th>
<th>They should have called a cab.</th>
<th>tense, modals person/number</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMPERATIVE (FINITE)</td>
<td>Put that in your pipe and smoke it.</td>
<td>Call a cab.</td>
<td></td>
</tr>
<tr>
<td>SUBJUNCTIVE (FINITE)</td>
<td>I recommend that we smoke them out.</td>
<td>She suggested I call a cab.</td>
<td></td>
</tr>
<tr>
<td>INFINITIVE (NON-FINITE)</td>
<td>To smoke was a mistake.</td>
<td>What I did was call a cab.</td>
<td></td>
</tr>
<tr>
<td>PARTICIPLE (NON-FINITE)</td>
<td>Having smoked all his life, he died of cancer.</td>
<td>Calling a cab, he dressed swiftly.</td>
<td></td>
</tr>
</tbody>
</table>

Modal verbs are always finite, so the only spot they can occupy in the table above is as first word in indicative group. They do not occur in imperative, subjunctive or non-finite verb groups.

2.1.6 Structure of verb groups

The structure of verb groups is made up of four components (Quirk et al., 1985, pp. 151–154):

- MODAL: modal auxiliary followed by base form.
- PERFECT: the auxiliary have followed by past participle.
- PROGRESSIVE: the auxiliary be followed by present participle.
- PASSIVE: the auxiliary be followed by past participle.

When two or more of these occur in the same verb group, they can only come in the order above. A sequence perfect-progressive-progressive (* have been examining), for example, is not grammatical (Quirk et al., 1985, p. 152).

The components will also overlap partially, so that the final participle in one component and the initial auxiliary in the following component are realised by
the same word in the group. Quirk et al. (1985, p. 152) uses the group ‘may have been examined’ as an example, and it breaks down into the following pieces.

<table>
<thead>
<tr>
<th>MODAL</th>
<th>PERFECT</th>
<th>PASSIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>may</td>
<td>Auxiliary may</td>
<td>Auxiliary be</td>
</tr>
<tr>
<td>have</td>
<td>Infinitive have</td>
<td>Auxiliary have</td>
</tr>
<tr>
<td>been</td>
<td>PAST PARTICIPLE</td>
<td>PAST PARTICIPLE</td>
</tr>
<tr>
<td>examined</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Each column is one word in the group and the middle two words participate in two components each.

2.1.7 List of verb groups

All possible forms of one single verb group are listed in the table on page 12. The first word in the group, being finite, can have a past and present form. For the modal verbs, the table lists only ‘will,’ partly for brevity and also because the difference in meaning between it and, for example, ‘would’ is more than just tense. So ‘will’ can be replaced with any of the other modal auxiliaries (section 2.1.3), and be in progressive and passive constructions can be replaced by the other verbs listed in section 2.1.2.

If two or more verb groups occur next to each other, additional complexity can be achieved. The following sentence, for example (Quirk et al., 1985, p. 154) contains the verb sequence PASSIVE + PERFECT PASSIVE.

Jackson was believed | to have been killed.

Non-finite verb clauses can be used in various places as nominals and modifiers.

| Adverb                  | She called hoping to borrow a record. |
|                        | His job, digging ditches, makes him strong. |
| Nominal clause, subject| Playing music makes her happy.       |
| Nominal clause, object  | She likes playing music.             |
| Pre-modifier, noun phrase | I talked to the retired policeman. |
| Post-modifier, noun phrase | The car being repaired by him is red. |
| Adjective complement    | John is busy writing letters.       |
| Prep. complement        | He is guilty of forging the signature. |

And verbs can become nouns through conversion or nominalization.

<table>
<thead>
<tr>
<th>Verb</th>
<th>Conversion</th>
<th>Nominalization</th>
</tr>
</thead>
<tbody>
<tr>
<td>They started to quarrel.</td>
<td></td>
<td>He destroyed it.</td>
</tr>
<tr>
<td>Then came the quarrel.</td>
<td></td>
<td>The destruction pleased her.</td>
</tr>
</tbody>
</table>

For more on nominalizations, see page 18 and section 3.2.1.
### Finite Verb Groups

#### Indicative Mood

<table>
<thead>
<tr>
<th>Tense</th>
<th>Form</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present</td>
<td>examine/examines</td>
<td>PRESENT</td>
</tr>
<tr>
<td></td>
<td>examined</td>
<td>PAST</td>
</tr>
<tr>
<td>Modal</td>
<td>will examine</td>
<td></td>
</tr>
<tr>
<td>Perfect</td>
<td>have/has examined</td>
<td>PRESENT PERFECT</td>
</tr>
<tr>
<td></td>
<td>had examined</td>
<td>PAST PERFECT</td>
</tr>
<tr>
<td>Progressive</td>
<td>am/are/is examining</td>
<td>PRESENT PROGRESSIVE</td>
</tr>
<tr>
<td></td>
<td>was/were examining</td>
<td>PAST PROGRESSIVE</td>
</tr>
<tr>
<td>Passive</td>
<td>am/are/is examined</td>
<td></td>
</tr>
<tr>
<td></td>
<td>was/were examined</td>
<td></td>
</tr>
<tr>
<td>Perfect/Progressive</td>
<td>have/has been examining</td>
<td>PRESENT PERFECT PROGRESSIVE</td>
</tr>
<tr>
<td></td>
<td>had been examined</td>
<td>PAST PERFECT PROGRESSIVE</td>
</tr>
<tr>
<td>Perfect/Passive</td>
<td>have/has been examined</td>
<td></td>
</tr>
<tr>
<td></td>
<td>had been examined</td>
<td></td>
</tr>
<tr>
<td>Progressive/Passive</td>
<td>am/are/is being examined</td>
<td></td>
</tr>
<tr>
<td></td>
<td>was/were being examined</td>
<td></td>
</tr>
<tr>
<td>Modal/Perfect</td>
<td>will have been examining</td>
<td></td>
</tr>
<tr>
<td>Modal/Progressive</td>
<td>will have been examined</td>
<td></td>
</tr>
<tr>
<td>Modal/Passive</td>
<td>will be examined</td>
<td></td>
</tr>
<tr>
<td>Perfect/Passive</td>
<td>have/has been being examined</td>
<td></td>
</tr>
<tr>
<td></td>
<td>had been being examined</td>
<td></td>
</tr>
</tbody>
</table>

#### Imperative Mood

<table>
<thead>
<tr>
<th>Form</th>
<th>Tense</th>
</tr>
</thead>
<tbody>
<tr>
<td>examine</td>
<td>Present</td>
</tr>
<tr>
<td>have examined</td>
<td>PAST</td>
</tr>
<tr>
<td>be examining</td>
<td>PRESENT</td>
</tr>
<tr>
<td>be examined</td>
<td>PAST</td>
</tr>
</tbody>
</table>

#### Subjunctive Mood

<table>
<thead>
<tr>
<th>Tense</th>
<th>Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present</td>
<td>examine</td>
</tr>
<tr>
<td></td>
<td>were [examining]</td>
</tr>
<tr>
<td></td>
<td>be examining</td>
</tr>
<tr>
<td></td>
<td>be examined</td>
</tr>
</tbody>
</table>

### Non-Finite Verb Groups

#### Infinitives

<table>
<thead>
<tr>
<th>Tense</th>
<th>Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perfect</td>
<td>to examine</td>
</tr>
<tr>
<td></td>
<td>to have been examined</td>
</tr>
<tr>
<td></td>
<td>to be examining</td>
</tr>
<tr>
<td></td>
<td>to be examined</td>
</tr>
<tr>
<td>Progressive</td>
<td>to have been examining</td>
</tr>
<tr>
<td></td>
<td>to be examining</td>
</tr>
<tr>
<td></td>
<td>to be examined</td>
</tr>
<tr>
<td>Passive</td>
<td>to have been examined</td>
</tr>
<tr>
<td></td>
<td>to be examined</td>
</tr>
<tr>
<td>Perfect/Progressive</td>
<td>to have been being examined</td>
</tr>
<tr>
<td></td>
<td>to be being examined</td>
</tr>
<tr>
<td>Perfect/Passive</td>
<td>to have been being examined</td>
</tr>
<tr>
<td></td>
<td>to be being examined</td>
</tr>
<tr>
<td>Perfect/Progressive/Passive</td>
<td>to have been being being examined</td>
</tr>
<tr>
<td>PerERICA/PRO/Passive</td>
<td>to have been being being examined</td>
</tr>
</tbody>
</table>
2.2 Temporal meaning of verb groups

As seen in the table on page 12, there is a large number of morphologic and syntactic forms of verb groups. The range of meaning these express is also large, but there is no simple mapping. In general, the most unmarked forms, such as the simple present, have a wide range of possible meanings, while more complicated forms have fewer. Perfect progressive, for example, seems to have one meaning only.

2.2.1 Finite verb groups without modal verbs

In a finite group without modals, the main verb can be either present or past, and, with the help of auxiliaries, the group can have perfect or progressive aspect, or both.

**Simple present**

Simple present is the most common form in timeless and general statements such as scientific facts. When used in narrative or discourse it can refer not only to the indexical present time, but also past and future.

<table>
<thead>
<tr>
<th>Type</th>
<th>Sample Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timeless/eternal</td>
<td>God is merciful.</td>
</tr>
<tr>
<td>Habitual</td>
<td>She sleeps late.</td>
</tr>
<tr>
<td>Performative</td>
<td>I call it a duck.</td>
</tr>
<tr>
<td>Narrative</td>
<td>Germany invades Poland.</td>
</tr>
<tr>
<td>Reportative (near past)</td>
<td>He scores.</td>
</tr>
<tr>
<td>(Near) future</td>
<td>The train leaves at nine.</td>
</tr>
</tbody>
</table>

The perhaps most natural way to refer to future time in English is to use simple present and a time adjunct, as in the last example above (see also 2.2.2).

**Simple past**

Past tense can, of course, be used for states and events in the past, relative to the current time of the discourse. The other three cases below are special in that the times of the events involved are present or future.

<table>
<thead>
<tr>
<th>Type</th>
<th>Sample Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Past event/state/habit</td>
<td>He moved to London.</td>
</tr>
<tr>
<td>Politeness</td>
<td>I wondered if you could lend me some money.</td>
</tr>
<tr>
<td>Hypothetical</td>
<td>I wish I was young again.</td>
</tr>
<tr>
<td>Counterfactual</td>
<td>If I only had time, I'd think of the perfect crime.</td>
</tr>
</tbody>
</table>

The use of past tense for present in the last two sentences is sometimes called back-shift (section 2.2.4).

**Present perfect**

The English present perfect refers to a state, event, or habit in the past, and which has present relevance.
I have bought a watch (and I still have it).

If the watch was bought and subsequently lost, the present perfect would not be appropriate.

There is an interesting difference between simple past and present perfect in combination with time adjuncts that refer to a specific time in the past. With most of these, only simple past can be used. The ones that accept present perfect typically refer to a very recent past (Comrie, 1985, pp. 32, 78, 84; Comrie, 1976, p. 54).

I saw her yesterday.  * I have seen her yesterday.
I saw her last week.  * I have seen her last week.
I saw her two days ago. * I have seen her two days ago.
I just saw her.    I have just seen her.
I saw her recently. I have seen her recently.

Other peculiarities involving present perfect and time adjuncts are discussed in section 2.3.4.

**Past perfect**

Past perfect is ambiguous between the past of either simple past or present perfect.

<table>
<thead>
<tr>
<th>The train <strong>arrived</strong></th>
<th><strong>SIMPLE PAST</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>The train <strong>has arrived</strong></td>
<td><strong>PRESENT PERFECT</strong></td>
</tr>
<tr>
<td>The train <strong>had arrived</strong></td>
<td><strong>PAST PERFECT</strong></td>
</tr>
</tbody>
</table>

A time adverbial combined with past perfect can refer to either the event, if it is a past of simple past, or the time when the present perfect is stated. In the first case we have a sentence like the following.

The Germans **had invaded** Poland in 1939.

While in the second, one like this.

The Germans were snug in 1940, because they **had invaded** Poland.

An effect of the underlying present perfect in the last example above is that the event must have some relevance at the time indicated by the adjunct.

**Progressive**

The progressive form typically indicates a situation that has duration and is not completed. This is the imperfective meaning. Used with a state, it suggests that it is temporary. It can also refer to a sequence of identical, punctual events.

- Event (imperfective) The train **is arriving**.
- (Temporary) habit He **is running** every morning.
- Repetition The phone **is ringing**.
- Politeness I **was wondering** if you can lend me some money.
- Future I **am leaving** tomorrow.
Like the simple past, progressive (present or past), can be used to express politeness. In this case there is no tense involved, meaning-wise, since the event is future in all cases, and past progressive is just ‘extra polite.’

**Perfect progressive**

Perfect progressive combines the meanings of perfect and progressive; it typically refers to a temporary and incomplete state of affairs that lasts up to the present, or, in the case of past perfect progressive, up to some other time given by context.

I have been writing a book.

If the book is finished, or have been accidentally destroyed, the perfect progressive is not the best choice here.

### 2.2.2 Finite verb groups with modal verbs

One could perhaps say that all modal verbs have a component of future in their meaning (Binnick, 1991, p. 8). A sign of this is that they can never combine directly with a verb in past tense.

- He may have done it. * He may have done it.
- He might have done it. * He might have done it.

The modal verbs ‘will’ and ‘shall’ can be used to express future and little else, and those constructions are often called ‘future tenses.’ But those verbs can also have other meanings, and there are alternatives for indicating future, such as present tense and the progressive aspect. When the two modals have the future meaning, they are close synonyms, but shall is less common and normally restricted to first person subject.

- **Future (prediction)** We shall know tomorrow. (First person)
- **Future (prediction)** You will feel better tomorrow. (Second person)
- **Future (volition)** I shall arrive next week.
- **Future (volition)** I will deliver it tomorrow.

There are also the phrases ‘be going to,’ ‘be about to,’ and ‘be to,’ with similar semantics.

- **Future (prediction)** It is going to rain.
- **Future (volition)** I am going to win the game.

Other meanings of will include present and habitual/timeless.

- **Present (prediction)** That will be the police, knocking on the door.
- **Present (prediction)** They will have finished by now.
- **Habitual** On Sundays, they will go to church.
- **Timeless** Diamond will cut glass.

The past forms of these auxiliaries can be used to indicate future in the past, with the restriction that ‘should’ can only be used in back-shift constructions (section 2.2.4).
Future in the past    Later, he *would regret* it.
Future in the past    She asked if she *should go.*  (*Shall I go?*)
Future in the past    They *were going to come.*

In addition, *would* can express politeness and hypothetical meaning.

Politeness    *Would* you spare me some change?
Hypothetical    *It would* be nice if that happened.

Besides the meanings discussed here, the other modal verbs typically express possibility, necessity, permission, and obligation in various combinations, and, focus kept on tense and aspect, nothing more shall be said about that.

The main verb in a group with a modal auxiliary is always a base form or a participle, and tense distinction only applies to the modal. The perfect and progressive aspects, however, can be formed with additional auxiliaries.

They *will have been finishing* by now.

In such a construction, the meaning is, not surprisingly, a combination of the modal meaning and the meaning of the aspect. It seems that some of the more special senses are not available for this type of construction. A timeless meaning, for example, is not easy to express with both a modal verb and progressive aspect.

They *will go* to church on Sundays.    Habitual, present or in the future
They *are going to church* on Sundays.    Present habitual
They *will be going to church* on Sundays.    Habitual in the future
They *were going to church* on Sundays.    Habitual in the past

Sometimes, the modal and aspect forms have similar meaning and the combination is redundant.

*I leave* tomorrow.    *(PRESENT)*
*I will leave* tomorrow.    *(Modal)*
*I am leaving* tomorrow.    *(PRESENT PROGRESSIVE)*
*I will be leaving* tomorrow.    *(Modal + PROGRESSIVE)*

All these sentences have similar content, a (volitional) future event.

### 2.2.3 Tense-less phrases

Tense is expressed by finite verb forms, and phrases without those include non-finite groups and nominalizations. Since all modals are finite, they cannot occur in tense-less phrases.

#### Non-finite verb groups

In a non-finite verb group tense is not marked directly, being instead relative to the governing verb.
Tears in the Rain

I expected him to finish. (yesterday)
I expected him to have finished. (before yesterday)

I expect him to finish. (today)
I expect him to have finished. (before today)

I will expect him to finish. (tomorrow)
I will expect him to have finished. (before tomorrow)

But the other forms — perfect, progressive, and passive — are possible.

I expect him to finish.
I expect him to be finishing. PROGRESSIVE
I expect him to have finished. PERFECT
I expect him to have been finishing. PERFECT PROGRESSIVE

I expect it to be finished. PASSIVE
I expect it to be being finished. PASSIVE + PROGRESSIVE
I expect it to have finished. PASSIVE + PERFECT
I expect it to have been finished. PASSIVE + PERFECT + PROGRESSIVE

Generally speaking, the simple forms of non-finite verb groups, in the absence of adjuncts, describe situations that are either timeless, or simultaneous with the event they modify, and the perfect forms situations that are prior to them. The present relevance that the perfect form expresses in finite phrases is perhaps less pronounced here, as can be seen by the paraphrases below, especially for the last example.

<table>
<thead>
<tr>
<th>Participle</th>
<th>Finite phrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whistling merrily, he left.</td>
<td>He whistled merrily when he left.</td>
</tr>
<tr>
<td>Having whistled merrily, he left.</td>
<td>He left after he whistled merrily. He left after he had whistled merrily.</td>
</tr>
<tr>
<td>Having left, he whistled merrily.</td>
<td>He whistled merrily after he left. He whistled merrily when he had left. He whistled merrily after he had left.</td>
</tr>
</tbody>
</table>

There seems to be little difference in meaning between the last three finite phrases above, which presumably means that the perfect form here expresses anteriority and not much more, and that the perfect in the last sentence is redundant. Another indication is that the prepositions ‘before’ and ‘after’ do not combine well with the perfect form.

Before leaving, he whistled merrily.
After leaving, he whistled merrily.
? Before having left, he whistled merrily.
? After having left, he whistled merrily.

Since the perfect indicates that the whistling is posterior to the leaving, adding ‘before’ is contradictory and adding ‘after’ redundant.

Non-finite groups will happily combine with time adjuncts. There is little difference between modifying a sentence adverbial and modifying the main verb group.
Whistling merrily, he left at noon. (Modifying finite group)
Leaving at noon, he whistled merrily. (Modifying sentence adverbial)

If the verb meanings allow it, there can be time adjuncts modifying both.

After leaving at noon, he whistled merrily for an hour.
Having left in the morning, he whistled merrily all day.

Yesterday he was hoping to do it next week.

Progressive aspect in non-finite phrases has more or less the same range of meaning as in finite phrases, and in infinitives the syntax is also similar. In participle phrases, however, the syntactic realisation of progressive form is sometimes awkward, and the present participle alone assumes progressive meaning.

I saw him paint the phone booth.  Bare infinitive
I saw him painting the phone booth.  Present participle
† I saw him being painting the phone booth.  Present participle + PROGRESSIVE

In other cases, such as the sentence adverb below, the infinitive form cannot be used, the present participle has the simple present meaning, and progressive meaning has no syntactic realisation.

Being hungry, I bought a hamburger.  Participle
† Be hungry, I bought a hamburger.  Bare infinitive
† To be hungry, I bought a hamburger.  Inactive

I bought a hamburger when I was hungry.  SIMPLE PAST
† I bought a hamburger when I was being hungry.  PAST PROGRESSIVE

With a passive construction, progressive is formed with the present participle of be, similarly to a finite phrase.

I saw the phone booth painted by him.  Past participle
I saw the phone booth being painted by him.  Past participle + PROGRESSIVE

A special distinction that has no direct counterpart in finite phrases, and is not possible for all verbs in the dominating clause, is between infinitive and participle.

He tried to paint the phone booth.  Infinitive
He tried painting the phone booth.  Participle

The difference in meaning depends on the dominating verb, and for ‘try’ it is between trying, and by implication failing, to paint the booth, and painting the booth in order to achieve something else (Quirk et al., 1985, pp. 1191–1193).

Nominalisations

Derived nominalisations do not have tense or aspect.

**Finite verb phrase**  **Nominalisation**
Germany invaded Poland in 1939.  The invasion was in 1939.
Germany is invading Poland now.  The invasion happens now.
Germany had invaded Poland in 1939.  The invasion had happened in 1939.
But they can be modified by temporal adjuncts.

The 1939 invasion of Poland by Germany was successful.
The ongoing invasion of Poland is fictional.

Nominalizations are also discussed in section 2.1.7.

2.2.4 Shifts in tense and mood

In indirect speech with a complex sentence, the tense of the subordinate clause is shifted back one step, so to speak, from the tense of the governing verb.

'I am tired.' He said he was tired.
* He said he is tired.

'I have been waiting.' She said she had been waiting.
* She said she has been waiting.

Something similar occurs in certain hypothetical and counterfactual sentences.

I wish I was young again.
* I wish I am young again.

Mood can also shift, in some circumstances. When reporting an imperative, for instance, infinitive or subjunctive can be used (Binnick, 1991, p. 78).

'Jump!' I told him to jump. INFINITIVE
I requested that he jump. PRESENT SUBJUNCTIVE

There is no shift in tense here, as the verb forms involved do not have it (see the table on page 12).

2.3 Time adjuncts

The other major time-related part of the language, besides verb tense and aspect, are the time adjuncts. Syntactically, there is a wide choice of categories for these to belong to.

Noun phrase An hour passed.
Prepositional phrase It was on Monday.
Adverb We saw it yesterday.
Adjective Mondays are a weekly occurrence.
Conjunction She left when the game was over.
Non-finite clause Leaving the party, I met my brother.
Verb-less clause Finally too curious, he read the letter.

Meaning-wise, a general classification is into the three groups: location, duration, and frequency. Location adjuncts identify the point in time when something occurred, either using a calendar system or in relation to context or some other known event.
The duration adjuncts reveal how long it took.

Measurement: five minutes
Relation: quicker than last time

And the frequency adjuncts how many times.

Measurement: ten times
Relation: every time the church bells ring

The distinction between location and duration can be compared to the distinction between ordinal and cardinal numbers. An ordinal number identifies a position on a scale such as the time axis, and a cardinal number gives the size of an interval.

Location: The light flashed at 20:33 p.m. (ORDINAL)
Duration: The light was on for two minutes. (CARDINAL)

But these are only the simplest cases. It is quite possible to use a cardinal measurement to identify a time location and vice versa.

Location: The light flashed after two minutes. (CARDINAL)
Duration: The light was on until 20:33 p.m. (ORDINAL)

A perhaps curious fact about natural language is that it does not seem to harbour any partiality towards certain particular lengths of time. It is no more difficult and time consuming to talk about glacial movements than microprocessor-operations. This is mainly because of units whereby words or phrases of similar length and complexity can stand for amounts of time that differ greatly not just in magnitude but also in precision.

2.3.1 Units of time

The original basis for human time measurement is the planetary motions. Days, months, and years can be observed directly in the changes of the sun, moon, and seasons. The division of a day into hours, minutes, and seconds is pre-historic and might be related to the Sumerian counting system which was sexagesimal.

In addition to the ancient units, we have the modern scientific ones that uses Greek letter prefixes.

Traditional: year, month, day, hour, minute, second
century, millennium, aeon

Scientific: millisecond, picosecond ...

The use of units has two benefits: it makes it easier to handle different sizes of magnitude, and it allows the precision of the measurements to be varied. It

1The same distinction is made in Bennett (1975, p. 114), using the terms ‘caendrical’ and ‘non-caendrical’ time units.

2Based on 60.
is perfectly possible to express any amount of time in any unit. Five hundred years, for example, can be written as:

\[ 15778800000 \text{ seconds} \]

Assuming 365.25 days per year.
Likewise, ten seconds can be written as:

\[ 0.00000031681 \ldots \text{years} \]

But this is cumbersome, both because it requires more digits, and also because it makes incorrect statements about the precision. If we say that the Ottoman Empire lasted for five hundred years, it can perhaps be taken to mean anything between four hundred and six hundred, depending on the context. (Counting from the fall of Constantinople to the end of the Great War gives 465 years.)

If we desire to make a very exact statement about some event a long time ago, units let us do that too, with few digits:

At precisely 13:01 p.m. on December 25th 539 BC.

This flexibility of precision is available for cardinal time specifications as well.

<table>
<thead>
<tr>
<th>Location</th>
<th>Duration</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>12:45:01</td>
<td>five seconds</td>
<td>second</td>
</tr>
<tr>
<td>three o’clock</td>
<td>one hour</td>
<td>hour</td>
</tr>
<tr>
<td>1 January / today</td>
<td>three days</td>
<td>day</td>
</tr>
<tr>
<td>1999</td>
<td>seven years</td>
<td>year</td>
</tr>
</tbody>
</table>

Using a smaller unit gives higher precision, for the same interval.

<table>
<thead>
<tr>
<th>Length</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>six months</td>
<td>183 days plus/minus 14 days</td>
</tr>
<tr>
<td>183 days</td>
<td>183 days plus/minus 12 hours</td>
</tr>
</tbody>
</table>

It is not necessary to use numbers to specify times. Determiners can combine directly with units.

this week, next week, last week, that week, some week
this year, next year, last year, that year, some year

There is a natural connection between units of time and the typical length of the event under discussion. If someone had stated, for example, that the Ottoman Empire lasted a certain large number of seconds, then he would presumably be asked how he could know it so exactly.

This is also evident from the fact that indexical words like then and now seem to imply some specific unit that is given by the context.

When will you start the compilation? Now (in a few seconds)
When are you going to move? Now (today or tomorrow)
When will the new airport be finished? Now (this year or so)

If it so happens that the airport is being opened at the same time the question is asked it would be more appropriate to answer something like:
When will the new airport be finished? This very minute / as we speak.

In addition to selecting the appropriate unit of measurement, time specifications in natural language can be qualified by, for example, adverbs like 'exactly' and 'roughly.' There are more ways to express this but they all boil down to the same thing: a certain amount of time, with a certain precision.

2.3.2 Location and duration

If a duration is given a position on the time axis the result is a time span. A time adjunct specifying a span will most naturally combine with a state or activity (section 3.2), that has extension in time.

I worked from 8 a.m. until 5 p.m.

But it can also be used to give an outer bound on the location of an instantaneous occurrence.

The painting disappeared between 9 p.m. and midnight.

Some time adjuncts specify only the start point or end point in a span. This means the duration will be undefined, if it is not given by the context.

It disappeared after 9 p.m.
It disappeared before midnight.

In the table below the following symbols will be used to indicate this.

→ o Up to the end point.
 o → From the start point onwards.

In Quirk et al. (1985, p. 481) these are called 'forward span' and 'backward span.'

Additionally, the precision of both the location and the length of a span can be adjusted.

<table>
<thead>
<tr>
<th>Start</th>
<th>End</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vague</td>
<td>Vague</td>
<td>about six months starting sometime in June</td>
</tr>
<tr>
<td>Vague</td>
<td>Vague</td>
<td>183 days from sometime in June</td>
</tr>
<tr>
<td>Vague</td>
<td>Precise</td>
<td>from June sometime until November 30th</td>
</tr>
<tr>
<td>Vague</td>
<td>Precise</td>
<td>—</td>
</tr>
<tr>
<td>Precise</td>
<td>Vague</td>
<td>about six months from June 1st onwards</td>
</tr>
<tr>
<td>Precise</td>
<td>Vague</td>
<td>—</td>
</tr>
<tr>
<td>Precise</td>
<td>Precise</td>
<td>from June 1st until November 30th</td>
</tr>
</tbody>
</table>

If any two of the start time, end time, and length are precise, then the third will be well-defined also.
The table below lists the most common temporal prepositions/conjunctions.

### LOCATION

<table>
<thead>
<tr>
<th>Preposition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>at ordinal</td>
<td>The train arrived at 8:33.</td>
</tr>
<tr>
<td>in cardinal</td>
<td>It will open in 5 minutes.</td>
</tr>
<tr>
<td>cardinal ago</td>
<td>They left 10 minutes ago.</td>
</tr>
<tr>
<td>within cardinal</td>
<td>She won within 5 minutes ago.</td>
</tr>
<tr>
<td>after cardinal</td>
<td>It closed after 10 seconds.</td>
</tr>
<tr>
<td>cardinal before ordinal/event</td>
<td>I left one minute before she did.</td>
</tr>
<tr>
<td>cardinal after ordinal/event</td>
<td>I left two minutes after that.</td>
</tr>
<tr>
<td>when event</td>
<td>He stopped when she told him to.</td>
</tr>
<tr>
<td>cardinal earlier</td>
<td>They opened five minutes earlier.</td>
</tr>
<tr>
<td>cardinal later</td>
<td>It broke down a minute later.</td>
</tr>
</tbody>
</table>

### LOCATION in a span

<table>
<thead>
<tr>
<th>Preposition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>on ordinal</td>
<td>He arrived on Monday.</td>
</tr>
<tr>
<td>in ordinal</td>
<td>The race was in June.</td>
</tr>
<tr>
<td>in cardinal</td>
<td>He fell once in a hundred years.</td>
</tr>
<tr>
<td>between ordinal and ordinal</td>
<td>The shop opens between 8 and 9.</td>
</tr>
<tr>
<td>before ordinal/event</td>
<td>I arrived before 9am.</td>
</tr>
<tr>
<td>after ordinal/event</td>
<td>I left after the job was done.</td>
</tr>
<tr>
<td>following event</td>
<td>Following the fire, it disappeared.</td>
</tr>
<tr>
<td>during eventive n.p./ordinal</td>
<td>They arrived during the night.</td>
</tr>
<tr>
<td>by ordinal</td>
<td>It was over by noon.</td>
</tr>
</tbody>
</table>

### DURATION (STATE/ACTIVITY)

<table>
<thead>
<tr>
<th>Preposition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>for ordinal</td>
<td>She played for five minutes.</td>
</tr>
<tr>
<td>from ordinal</td>
<td>It was open from 8:00.</td>
</tr>
<tr>
<td>to/till/until ordinal/event</td>
<td>He waited until noon.</td>
</tr>
<tr>
<td>throughout ordinal/event</td>
<td>She worked throughout the day.</td>
</tr>
<tr>
<td>while event</td>
<td>He slept while they worked.</td>
</tr>
<tr>
<td>since ordinal/event</td>
<td>It has been idle since it broke.</td>
</tr>
</tbody>
</table>

Location and duration can also be specified by an adjective (my year-long absence), adverb (It happened yesterday), or noun phrase (I spent the night in jail).

### 2.3.3 Containment

Containment adverbials, or 'containers' (Binnick, 1991, pp. 190, 307) are similar to durative adverbials in that they specify a length of time during which something happens, but with the difference that the period is an outer bound.

3Bennett and Partee (1978) call these 'frame adverbials.'

4When can also have a timeless meaning: 'When it rains, it pours,' and a conditional meaning: 'When the engine stopped, the plane crashed' where the events are not necessarily simultaneous.

5There is another, counterfactual, sense of before, as in 'She died before she finished the book.'

6Since can also mean 'because,' as in 'We stopped using it since it broke.'
and that there must be a definite end point to the happening. Typical examples are ‘in,’ ‘within,’ and ‘inside of.’

He drove there in a day.
They arrived within an hour.
The bridge was finished inside of a week.

In the previous section these are listed as indicating a location, since they can be used in that sense as well. One could perhaps claim that there is no real difference, and that the event that is specified in the examples in this section is the end of the driving, arriving, and bridge-building.

2.3.4 Combination with perfect aspect

In contrast to the others in the table on page 23, the prepositions ‘for,’ ‘since,’ and ‘by,’ in the temporal sense that they are used there, seem to share a preference for perfect aspect in the verb group they modify. The first has two somewhat different meanings, and when it refers to an unspecified period that is not connected to the time of utterance, simple tense is used.

John lived in London for three years in the 1970s. simple past
† John has lived in London for three years (in the 1970s). present perfect

And in the other sense, when the period extends up to, and includes, the utterance time, only perfect is acceptable.

† John lived in London for three years (from three years ago).
John has lived in London for three years.

Similarly, ‘since,’ which has only one sense, requires perfect aspect.

* John lived in London since 1980.
John had lived in London since 1980.

The third preposition, ‘by,’ displays the same reluctance to modify simple tenses as in the preceding examples.

* John moved to London by 1980.
John had moved to London by 1980.

But, unlike ‘since,’ it will happily combine with constructions involving ‘be.’

John was living in London by 1980. progressive
John was promoted by 1980. passive
John was dead by 1980. copula

A modal construction is also acceptable.

John would move to London by 1980.

With these exceptions, most time adjuncts can combine freely with all tense and aspect categories.
2.3.5 Repetition and frequency

In addition to location and duration, time adjuncts can express the number of times something happened, and the frequency, which is the time between repetitions. There is a mathematical relationship between these, and if, for instance, the number of repetitions during a certain time span is known, then the frequency can be calculated. Some adjuncts specify only the number of occurrences: 'once,' 'twice,' 'many times.' Then there are those that give the frequency either by comparison to some context-dependent standard or by relation to other events.

Comparison: often, rarely
Relation: whenever the bell tolls, every time the bell tolls

It is also possible to express the frequency in combination with either position or duration.

Position: daily, weekly, once a month, every Monday, on Mondays
Duration: five times in a week

And of course all three can be specified to any degree of precision.

He visited her every day, Monday through Friday, last week, at noon.

The syntactic structure of time adjuncts can be complex, but their meaning is really quite straightforward: position, duration, and frequency, where the spans can be open-ended, and with arbitrary precision.

2.4 Aspectual expressions

In contrast to, for example, the Slavic languages, English has few morphological or grammatical phenomena that by general agreement are considered aspectual. The following (Dowty, 1979, p. 52) is a good summary of the problem.

Aspect is distinguished from tense from the point of view of semantics in that tenses serve to relate the time of a situation described to the time of speaking, whereas aspect markers serve to distinguish such things as whether the beginning, middle or end of an event is being referred to, whether the event is a single one or a repeated one, and whether the event is completed or possibly left incomplete. By this use of the term aspect, the only instances of pure aspect markers in English are the progressive 'tense' and the habitual quasi-auxiliary used to.

Between all human languages, the main aspectual categories that can be distinguished syntactically or morphologically, are the following (Comrie, 1976).

perfective An event or situation viewed as a whole, complete with beginning, middle, and end. Can also mean that it is finished or completed.

imperfective The opposite of perfective; a situation seen from within. Also incomplete, unfinished.

progressive An ongoing non-stative situation.
**habitual** Something that holds continuously or is repeated during and extended period.

**perfect** Present relevance of a past situation.

The English perfect and progressive are not clear instances of these. One could perhaps say that, in general, English perfect combines perfect and perfective meaning, while English progressive combines progressive and imperfective, but there are several exceptions to this rule (section 2.2.1).

Another class of aspectual meaning concerns the internal structure of happenings: beginning, middle, end; interruption, continuation and so on. In English this is mainly expressed by aspectual verbs. Binnick (1991, pp. 180, 207) illustrates the potential complexity of this with the following examples.

She ran.
She started to cease running.
She continued to run.
She was about to continue to run.
She was continuing to run.
She started to cease to continue to run.
She ceased to continue to run.
She resumed starting to cease to continue to run.
She had been about to continue to run.
She had been continuing to run.
She had continued to run.

In addition to the beginning of a situation (the running), the beginning itself can be interrupted and resumed and so on.

Besides perfect and progressive, the main syntactic aspectual categories in English are the aspectual verbs and time adjuncts. The major (aspectual) semantic classes they can be divided into are listed below, together with the few alternative grammatical realizations they can have (Binnick, 1991, pp. 59, 145, 147, 154; Dinsmore, 1991, p. 208).

<table>
<thead>
<tr>
<th>Prospective</th>
<th>Verb groups</th>
<th>to be about to, to be on the point of</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ingressive</td>
<td>Aspectual verb</td>
<td><strong>start, begin</strong></td>
</tr>
<tr>
<td></td>
<td>Verb suffix</td>
<td><strong>strengthen</strong></td>
</tr>
<tr>
<td>Termination</td>
<td>Aspectual verb</td>
<td><strong>stop, abort, interrupt</strong></td>
</tr>
<tr>
<td>Completion</td>
<td>Aspectual verb</td>
<td><strong>finish</strong></td>
</tr>
<tr>
<td></td>
<td>Verb particle</td>
<td><strong>ate the cake / ate up the cake</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>broke/pulled/tore/ripped off</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>pulled away</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>pulled out</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>read through</strong></td>
</tr>
<tr>
<td>Stability/change</td>
<td>Auxiliary verb</td>
<td><strong>have a cake / get a cake</strong></td>
</tr>
<tr>
<td>Habitual</td>
<td>Auxiliary</td>
<td><strong>use to</strong></td>
</tr>
<tr>
<td></td>
<td>Time adjunct</td>
<td><strong>always</strong></td>
</tr>
<tr>
<td>Repetition</td>
<td>Time adjunct</td>
<td><strong>once, twice, many times</strong></td>
</tr>
</tbody>
</table>

This table is by no means complete, and these meanings can also be the main content of sentences, as in ‘It is my habit to ...’ for expressing habituality.
3. Theoretical foundations

The two major origins for computational theories of tense and aspect are time points (Reichenbach, 1947) and the classification of verb, or 'situation,' types which dates back to Aristotle.

3.1 Time points

The most obvious part of the meaning of verb tense is that it expresses the time when something happens, relative to the time of the statement.

- The eagle landed yesterday. PAST
- The eagle lands now. PRESENT
- The eagle will land tomorrow. 'FUTURE'

In the sentences above there are two times involved: the time when the eagle lands, and the time when the statement is made. For simplicity, disregarding the fact that neither the making of statements nor the landing of eagles are instantaneous, they can, following Reichenbach (1947, p. 288), be referred to as point (or time) event and point of speech.

Reichenbach’s innovation was to generalise this analysis to also cover past perfect, through the introduction of a third time point, the point of reference, lying between the other two, and signifying the perspective from which the event is considered.

- The eagle landed. PAST
- The eagle had landed. PAST PERFECT

In a discourse, the point of reference in one sentence can coincide with the event point in another.

The pigeon was safe. The eagle had landed.

Here, the reference point in the second sentence is the time when the pigeon was safe, which is later than when the eagle landed and earlier than the point of speech.

- The eagle had landed.

<table>
<thead>
<tr>
<th>Event point</th>
<th>Reference point</th>
<th>Speech point</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Eagle lands,)</td>
<td>(Pigeon safe,)</td>
<td>'now'</td>
</tr>
</tbody>
</table>

| E | R | S |

Returning to the past tense, it does not require a great leap of the imagination to say that the reference point is in fact involved here also, being identical to the event point. Similarly, in the present tense all three points can be said to coincide. (The table below is identical to one found in Reichenbach (1947, p. 290) apart from the examples and formatting.)
This table illustrates clearly the symmetry of Reichenbach's analysis. The distinction between past, present, and future mirrors the relation between reference point and speech point, as shown below. For the other dimension, based on the relation between event point and reference point, Reichenbach uses the terms ANTERIOR (roughly corresponding to perfect), SIMPLE, and POSTERIOR.

<table>
<thead>
<tr>
<th></th>
<th>PAST</th>
<th>PRESENT</th>
<th>'FUTURE'</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAST PERFECT</td>
<td>The eagle had landed.</td>
<td>The eagle has landed.</td>
<td>The eagle will have landed.</td>
</tr>
<tr>
<td></td>
<td>E - R - S</td>
<td>E - S, R</td>
<td>S - E - R</td>
</tr>
</tbody>
</table>

In addition to the nine categories given by these two distinctions, the relation between speech point and event point, when the reference point alone is either first or last, produces four more. So the total, in Reichenbach's system, is thirteen possible tense categories, listed below together with Reichenbach's own examples (Reichenbach, 1947).

<table>
<thead>
<tr>
<th>E - R - S</th>
<th>Reichenbach</th>
<th>Traditional</th>
</tr>
</thead>
<tbody>
<tr>
<td>E - R - S</td>
<td>ANTERIOR PAST</td>
<td>I had seen John.</td>
</tr>
<tr>
<td>R - E - S</td>
<td>SIMPLE PAST</td>
<td>I saw John.</td>
</tr>
<tr>
<td>R - S, E</td>
<td>POSTERIOR PAST</td>
<td>He would win the race.</td>
</tr>
<tr>
<td>R - S - E</td>
<td>ANTERIOR PRESENT</td>
<td>He was going to win the race.</td>
</tr>
<tr>
<td>E - S, R</td>
<td>ANTERIOR PRESENT</td>
<td>I have seen John.</td>
</tr>
<tr>
<td>S, E - R</td>
<td>SIMPLE PRESENT</td>
<td>I see John.</td>
</tr>
<tr>
<td>S - E - R</td>
<td>POSTERIOR PRESENT</td>
<td>Now I shall go.</td>
</tr>
<tr>
<td>S - E - R</td>
<td>ANTERIOR FUTURE</td>
<td>I shall have seen John.</td>
</tr>
<tr>
<td>S - E - R</td>
<td>SIMPLE FUTURE</td>
<td>I shall go tomorrow.</td>
</tr>
<tr>
<td>S - R - E</td>
<td>POSTERIOR FUTURE</td>
<td>I shall be going to see him.</td>
</tr>
</tbody>
</table>

Not all of these map easily to verb tenses in English, however. Most of Reichenbach's examples use modal verbs, and the translation is not totally systematic. The form 'shall + present', for example, is rendered as either posterior present (S, R - E) or simple future (S - R, E), depending on context (Reichenbach, 1947, pp. 295–296). Likewise, 'would' + present is converted to both R - S, E (Reichenbach, 1947, p. 293) and R - E - S (Reichenbach, 1947, p. 297).

Kamp and Reyle (1993, pp. 523–526, 593–601; section 4.4), but it seems mis-guided.

These authors do not allow the reference time to be later than the utterance time (Kamp and Reyle, 1993, p. 598), which in Reichenbachian terms means that the 'future' category is not recognised. Other than that, and the terminology, their approach (Kamp and Reyle, 1993, pp. 512–513, 598) is identical to Reichenbach’s.

<table>
<thead>
<tr>
<th>(Reichenbach)</th>
<th>(Kamp &amp; Reyle)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAST</td>
<td>Reference point before utterance time.</td>
</tr>
<tr>
<td>PRESENT</td>
<td>Reference point same as utterance time.</td>
</tr>
<tr>
<td>ANTERIOR</td>
<td>Location time before reference point.</td>
</tr>
<tr>
<td>SIMPLE</td>
<td>Reference point same as location time.</td>
</tr>
<tr>
<td>POSTERIOR</td>
<td>Location time after reference point.</td>
</tr>
</tbody>
</table>

But their classification of verb groups (Kamp and Reyle, 1993, pp. 595–601) is different from his.

- **ANTERIOR PAST**: had written
- **SIMPLE PAST**: was writing, had written
- **POSTERIOR PAST**: would write, would have written
- **ANTERIOR PRESENT**: wrote, had written
- **SIMPLE PRESENT**: writes, has written
- **POSTERIOR PRESENT**: will write, will have written

Perfect aspect, which Reichenbach analyses as anterior, with the event before the reference point, is here spread out evenly over all categories.

### 3.2 Situation types

As mentioned on page 2, situation types identify the temporal profile of the verb meaning. Classification of these has a long history, going back, at least, to Aristotle. Other more or less synonymous terms are *verb aspect* (Dowty, 1979, p. 52; Binnick, 1991, p. 170), *aspectual class* (Dowty, 1979, p. 52; Hwang and Schubert, 1992), *aspectual type* (Moens and Steedman, 1988, p. 16; Galton, 1995, p. 226) *Aristotelian aspect* (Binnick, 1991, pp. 142, 170–178), *propositional aspect* (Steedman, 1997, p. 899, 905), and *semantic aspect* (Grover, Hitzeman and Moens, 1994, p. 6; Hitzeman, Moens and Grover, 1995, p. 257).

An obvious distinction is between static situations (states) and dynamic ones. The former are constant throughout their duration, while the latter have, as part of their core meaning, a behaviour in time.

- **I am happy.** STATIVE
- **I am running.** DYNAMIC

A stative proposition combines naturally with perfect.

- **I am tired.**
  - **I have been tired (all day).**
But in combination with progressive the meaning of the whole verb group is modified.

I am being tired.

The group ‘is tired’ + progressive could perhaps be expected to indicate that the tiredness is manifesting itself at the time of speech, but this does not seem to be the case. Indeed this is the typical meaning of the simple present in this case. The last sentence above would rather seem to express that the speaker only pretends to be tired. As an excuse to retire for the evening this sentence would sound odd.

A dynamic sense, in contrast, will straightforwardly combine with progressive.

Germany invades Poland (now).
Germany is invading Poland (now).

There is little difference between these two sentences, but the second might put slightly greater emphasis on the fact that the invasion is simultaneous with the time of speech.

### 3.2.1 Mass nouns and count nouns

A stative verb will not normally combine with frequency adverbials, since it makes no sense for it to be pluralised. Quirk et al. (1985, pp. 177–178), and Galton (1984), observes that this is somewhat parallel to the distinction between mass and count nouns, in that nominalizations of events typically correspond to count nouns, that can be pluralised.

<table>
<thead>
<tr>
<th>A SINGLE EVENT</th>
<th>A PLURALITY OF EVENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Germany invaded Poland.</td>
<td>Poland was invaded many times.</td>
</tr>
<tr>
<td>There was an invasion of Poland.</td>
<td>There were many invasions of Poland.</td>
</tr>
</tbody>
</table>

While nominalizations of stative propositions become mass nouns.

<table>
<thead>
<tr>
<th>STATIVE</th>
<th>DYNAMIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>I know that.</td>
<td>I selected the fish.</td>
</tr>
<tr>
<td>MASS NOUN</td>
<td>COUNT NOUN</td>
</tr>
<tr>
<td>My knowledge impressed her.</td>
<td>My selection was hasty.</td>
</tr>
</tbody>
</table>

This seems to hold for most cases, but there are exceptions.

<table>
<thead>
<tr>
<th>STATIVE</th>
<th>DYNAMIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>I believe in God.</td>
<td>I advised her.</td>
</tr>
<tr>
<td>COUNT NOUN</td>
<td>MASS NOUN</td>
</tr>
<tr>
<td>My belief was mistaken.</td>
<td>I gave her advice.</td>
</tr>
</tbody>
</table>

There are also nouns that are both count and mass, such as ‘experience.’

<table>
<thead>
<tr>
<th>STATIVE</th>
<th>DYNAMIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td>I experienced the storm.</td>
</tr>
<tr>
<td>COUNT NOUN</td>
<td>MASS NOUN</td>
</tr>
<tr>
<td>My experience of the storm [...]</td>
<td>I have experience of storms.</td>
</tr>
</tbody>
</table>
Besides that no absolute generalisation can apparently be made, the observation seems to be of limited practical utility.

### 3.2.2 Static versus dynamic

The observation above that only dynamic propositions can occur in the progressive without some modification of the meaning, seems to be the most often quoted principle for distinguishing those two situation types (Vendler, 1967, p. 99; Dowty, 1979, p. 55; Quirk et al., 1985, pp. 178, 200-202; Binnick, 1991, p. 173).

Some other characteristics that can be used for this classification are given below (Dowty, 1979, pp. 55–56; Binnick, 1991, pp. 173–175; Quirk et al., 1985, p. 178).

A dynamic expression not in the progressive (and without modifiers) will normally have a habitual meaning, while a static will not.

I *run*. (habitual)
I *am fit*. (present)

Verbs such as *force* and *persuade* normally combine with dynamic expressions only.

She made me *do it.*
* She made me *like* ice-cream.

Likewise, adverbs like *deliberately* and *carefully.*

He *closed* the door slowly.
* He *liked* the film slowly.

Frequency adjuncts normally modify dynamic expressions.

They often *have dinner at Dorsia.* (dynamic)
* They often *have three children.* (static)

And there is a preference for dynamic expressions in pseudo-deft constructions.

What he did was *close* the door.
* What he did was *like* the film.

It is not hard to find counter-examples to these maxims, but they nevertheless indicate a real distinction.

### 3.2.3 Exception: stance

According to Quirk et al. (1985, pp. 205–206), some senses of certain verbs (live, stand, sit, lie) fall between the categories static and dynamic, and are given the separate label *stance*.
<table>
<thead>
<tr>
<th>(present)</th>
<th>STATIVE</th>
<th>STANCE</th>
<th>DYNAMIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>PERFECT (unbroken state)</td>
<td>He has always liked cheese.</td>
<td>He has lived in London.</td>
<td>He drives home.</td>
</tr>
<tr>
<td>PERFECT (separate events)</td>
<td>—</td>
<td>He has lived in London many times.</td>
<td>He has driven all his life.</td>
</tr>
<tr>
<td>frequency adverbials</td>
<td>—</td>
<td>He lives there frequently.</td>
<td>He drives frequently.</td>
</tr>
<tr>
<td>PROGRESSIVE</td>
<td>—</td>
<td>He is living in London.</td>
<td>He is drawing home.</td>
</tr>
</tbody>
</table>

An alternative analysis might perhaps suggest that ‘live’ has several senses with different situation type.

3.2.4 Situations with a conclusion

In the Metaphysics (Aristotle, 1933, 1048b), Aristotle draws a distinction between ‘motion,’ and ‘actualisation,’ the difference being that ‘motions’ have, as a necessary component of their meaning, a conclusion.

Since no action which has a limit is an end, but only a means to the end, as, e.g., the process of thinning; and since the parts of the body themselves, when one is thinning them, are in motion in the sense that they are not already that which it is the object of the motion to make them, this process is not an action, or at least not a complete one, since it is not an end; it is the process which includes the end that is an action. e.g., at the same time we see and have seen, understand and have understood, think and have thought; but we cannot at the same time learn and have learnt, or become healthy and be healthy. We are living well and have lived well, we are happy and have been happy, at the same time; otherwise the process would have had to cease at some time, like the thinning-process; but it has not ceased at the present moment; we both are living and have lived.

Now of these processes we should call the one type motions, and the other actualisations. Every motion is incomplete — the processes of thinning, learning, walking, building — these are motions, and incomplete at that. For it is not the same thing which at the same time is walking and has walked, or is building and has built, or is becoming and has become, or is being moved and has been moved, but two different things; and that which is causing motion is different from that which has caused motion. But the same thing at the same time is seeing and has seen, is thinking and has thought. The latter kind of process, then, is what I mean by actualisation, and the former what I mean by motion.

It might be easier to follow this if the examples are listed separately.

<table>
<thead>
<tr>
<th>‘Actualisation’</th>
<th>‘Motion’</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>see (view)</td>
<td>thin (lose fat)</td>
<td>Having less fat</td>
</tr>
<tr>
<td>understand (know)</td>
<td>learn</td>
<td>New knowledge</td>
</tr>
<tr>
<td>think</td>
<td>become healthy</td>
<td>Better health</td>
</tr>
<tr>
<td>live well (be healthy)</td>
<td>walk (move)</td>
<td>New location</td>
</tr>
<tr>
<td>be happy</td>
<td>build</td>
<td>New artifact</td>
</tr>
<tr>
<td>live</td>
<td>move</td>
<td>New location</td>
</tr>
</tbody>
</table>
The authorities (Dowty, 1979, pp. 52–53; Binneck, 1991, pp. 143, 172) agree that Aristotle’s ‘actualisations’ include states, which is obvious and perhaps slightly confusing since there can be no state with a conclusion.

In the linguistic literature (Comrie, 1976, pp. 44–48; Binneck, 1991, p. 189) the terms telic and atelic are sometimes used for the distinction between situations with an intrinsic termination (telic) and those without (atelic). But, in contrast to Aristotle’s terms, these are restricted to situations with duration (Comrie, 1976, p. 47).

Worth noting is that conclusiveness is a property of verb groups in context, as opposed to a lexical property of verbs. Comrie (1976, pp. 45–46) offers the following examples.

- John is singing. (inconclusive)
- John is singing a song. (conclusive)
- John is singing songs. (inconclusive)
- John is singing five songs. (conclusive)

In a normal context, ‘singing’ has no unavoidable end but ‘singing a song’ does.

Now, says Comrie, consider a singing class where everybody has to sing a certain passage.

- John is singing. (conclusive)

Here this situation has a termination, since extra-sentential information tells us that John will sing for a limited and well-defined length of time.

For further discussion of this, see section 3.3.

### 3.2.5 Task performance and duration

Ryle (1949) takes Aristotle’s distinction one step further, dividing conclusive events into task performances, which are activities carried to a successful conclusion, and the instantaneous ‘purely lucky achievements.’

When a person is described as having fought and won, or as having journeyed and arrived, he is not being said to have done two things, but to have done one thing with a certain shoot. Similarly a person who has aimed and missed has not followed up one occupation by another; he has done one thing, which was a failure.

[...]

This is why we can significantly say that someone has aimed in vain or successfully, but not that he has hit the target in vain or successfully; that he has treated his patient assiduously or unassiduously, but not that he has cured him assiduously or unassiduously; that he scanned the hedgerow slowly or rapidly, systematically or haphazardly, but not that he saw the nest slowly or rapidly, systematically or haphazardly. (Ryle, 1949, pp. 144–145)

Two separate points are made here. The first is that conclusive events can succeed or fail, so that, potentially, the same physical activity can be described by three different verbs.
**TEARS IN THE RAIN**

kick  Activity with no inherent purpose, 'motion.'
score  Successfully completed action, 'actualisation.'
miss   Failure.

Secondly, there is a difference between durative and momentary conclusive events. Vendler (1967, pp. 102–103), who discussed temporal extent rather than presence or absence of intention, introduced the terms accomplishment and achievement.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Duration</th>
<th>Situation type</th>
</tr>
</thead>
<tbody>
<tr>
<td>run a mile</td>
<td>a few minutes</td>
<td>accomplishment</td>
</tr>
<tr>
<td>reach the top</td>
<td>none</td>
<td>achievement</td>
</tr>
</tbody>
</table>

While it does, usually, take a while to climb a mountain, the actual reaching of the top occurs in a single instant.¹ A consequence of this is that an accomplishment can be aborted.

Jack was building a house when he died. (accomplishment)
*I* was spotting the plane when a cloud obscured it. (achievement)

Note, however, that the following is intelligible.

He was about to spot the plane when a cloud obscured it.

But it describes the spotting as counterfactual, not aborted.

### 3.2.6 Dynamic situations

In Vendler (1967, pp. 102–103) achievements and states were distinguished from activities and accomplishments by the fact that the former do not combine with the progressive. But, other than that, states have little in common with the momentaneous and conclusive achievements. And so, although he recognised the two binary distinctions punctuality and conclusiveness, Vendler only made use of three of the four logical possibilities. The fourth category, situations that are punctual and non-conclusive, is called point and was apparently introduced by Miller and Johnson-Laird (Steedman, 1997, p. 902).

<table>
<thead>
<tr>
<th>Non-conclusive</th>
<th>Durative</th>
<th>Punctual</th>
</tr>
</thead>
<tbody>
<tr>
<td>activity</td>
<td>activity</td>
<td>point</td>
</tr>
<tr>
<td>Ernest <em>is</em> writing.</td>
<td></td>
<td>Uncas nodded sagely.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Conclusive</th>
<th>accomplishment</th>
<th>achievement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jack <em>built</em> a house.</td>
<td></td>
<td>The king <em>died.</em></td>
</tr>
</tbody>
</table>

Even though a nod is a physical act having a measurable duration — one second perhaps — it is conceptualised as indivisible. A sign of this is that the progressive (I am nodding) normally means that the action is repeated, not that one single nod is currently in progress.

¹But consider 'The climbers are reaching the summit' (Comrie, 1976, p. 43), which refers to a sequence of events over a period of time. Similar examples are discussed in section 3.3.
3.2.7 Activities / accomplishments / achievements

This section contains linguistic observations that can be used to distinguish between activities, accomplishments, and achievements (Vendler, 1967, pp. 100-103; Dowty, 1979, pp. 56-60; Binnick, 1991, pp. 175-177, 190; Verkuyl, 1993, pp. 35-49).

Durational time adjuncts come in two varieties: those that specify only the length of time, and those that also express an element of containment (see section 2.3.3).

<table>
<thead>
<tr>
<th>Duration only</th>
<th>Containment</th>
</tr>
</thead>
<tbody>
<tr>
<td>John walked for an hour.</td>
<td>* John walked in an hour.</td>
</tr>
<tr>
<td>John spent an hour walking.</td>
<td>* It took John an hour to walk.</td>
</tr>
</tbody>
</table>

**ACCOMPLISHMENT**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>* John painted a picture for an hour.</td>
<td>John painted a picture in an hour.</td>
</tr>
<tr>
<td>John spent an hour painting a picture.</td>
<td>It took John an hour to paint a picture.</td>
</tr>
</tbody>
</table>

**ACHIEVEMENT**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>* John noticed the painting for a few minutes.</td>
<td>John noticed the painting in a few minutes.</td>
</tr>
<tr>
<td>* John spent a few minutes noticing the painting.</td>
<td>It took John a few minutes to notice the painting.</td>
</tr>
</tbody>
</table>

Dowty (1979, p. 88) points out some exceptions.

**ACHIEVEMENTS**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>The soup cooled for ten minutes.</td>
<td>The soup cooled in ten minutes.</td>
</tr>
<tr>
<td>? The ship spent an hour sinking.</td>
<td>It took the ship an hour to sink.</td>
</tr>
</tbody>
</table>

The situation types under discussion differ in their implications.

**ACTIVITY**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>John is (now) walking.</td>
<td>John has walked.</td>
</tr>
<tr>
<td>John walked for an hour.</td>
<td>At any time during the hour,</td>
</tr>
</tbody>
</table>

**ACCOMPLISHMENT**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>John is (now) building a house.</td>
<td>John has not (yet) built a house.</td>
</tr>
<tr>
<td>John built a house for an hour.</td>
<td>At any time during the hour,</td>
</tr>
<tr>
<td>John built a house in an hour.</td>
<td>At any time during the hour,</td>
</tr>
</tbody>
</table>

**ACHIEVEMENT**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>John is (now) dying.</td>
<td>John has not (yet) died.</td>
</tr>
<tr>
<td>John died in an hour.</td>
<td>At any time during the hour,</td>
</tr>
</tbody>
</table>
And the ways in which they combine with ‘stop’ and ‘finish’ are different.

<table>
<thead>
<tr>
<th>ACTIVITY</th>
<th>Implies</th>
</tr>
</thead>
<tbody>
<tr>
<td>John stopped walking.</td>
<td>John did walk.</td>
</tr>
<tr>
<td>* John finished walking.</td>
<td>—</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACCOMPLISHMENT</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>John stopped painting the house.</td>
<td>John did not paint the house.</td>
</tr>
<tr>
<td>John finished painting the house.</td>
<td>John paint the house.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACHIEVEMENT</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>* John stopped noticing the painting.</td>
<td>—</td>
</tr>
<tr>
<td>* John finished noticing the painting.</td>
<td>—</td>
</tr>
</tbody>
</table>

Finally, certain adverbials such as ‘almost’ and ‘carefully’ combine differently with the various situation types.

<table>
<thead>
<tr>
<th>ACTIVITY</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>John almost walked.</td>
<td>But he changed his mind.</td>
</tr>
<tr>
<td>John walked carefully.</td>
<td>Soft steps.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACCOMPLISHMENT</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>John almost painted a picture.</td>
<td>But (1) he changed his mind.</td>
</tr>
<tr>
<td></td>
<td>or (2) he didn’t finish it.</td>
</tr>
<tr>
<td>John carefully painted a picture.</td>
<td>Next strokes.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACHIEVEMENT</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>John almost noticed the painting.</td>
<td>But he was distracted.</td>
</tr>
<tr>
<td>* John noticed the painting carefully.</td>
<td>—</td>
</tr>
</tbody>
</table>

The problem, of course, is that these principles are descriptions; they are not mechanical tests that can be straight-forwardly implemented in a computer program.

### 3.3 Type-shifting

There is general agreement (Comrie, 1976, p. 45; Dowty, 1979, pp. 60–62; Quirk et al., 1985, p. 200; Moens, 1987, p. 44; Steedman, 1997, p. 899) that the situation types apply to whole propositions, rather than words or verb groups.

To begin with, a verb that normally refers to a single event, like ‘drown’ below, can also be used in a general, timeless statement which is a state rather than an event.

<table>
<thead>
<tr>
<th>John is drowning.</th>
<th>ACCOMPLISHMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary drowned yesterday.</td>
<td>ACCOMPLISHMENT</td>
</tr>
<tr>
<td>Dolphins that get caught in nets drown.</td>
<td>STATE</td>
</tr>
<tr>
<td>If a dolphin gets caught in a net, it drowns.</td>
<td>STATE</td>
</tr>
</tbody>
</table>

A perhaps more common case of type-shifting is when a verb can be used with or without object.
Paavo is running.  
Paavo is running a mile.  

Spatial modifiers can have the same effect.

Paavo is running to the hills.

Another syntactic difference with the same meaning is whether or not the object has a specific quantity.

Wilhelm ate the apples.  
Wilhelm ate apples.

Gretchen found the mushroom.  
\text{? Gretchen found mushrooms.}  
Gretchen was finding mushrooms.

And the same holds for the subject.

The student scored well on the test.  
Students scored well on the test.

The driver broke the speed limit.  
Drivers broke the speed limit.

Vendler (1967, pp. 100-101; section 3.2.7) pointed out that durative time adjuncts like ‘for’ combine naturally with an activity, or a state, but not with accomplishments and achievements; for a containment adjunct (section 2.3.3), like ‘in,’ it is the other way around. So, if the verb meaning allows it, these adjuncts can shift the situation type.

Calvin transmogrified for an hour.  
Calvin transmogrified in an hour.

With appropriate modifiers, and verb meaning, the same verb can assume any of the dynamic situation types (section 3.2.6), at least, as in these examples (Galton, 1995, pp. 227-228).

\begin{itemize}
\item John left the room several times during the meal.  
\item It took John ten seconds to leave the room.  
\item While John was leaving the room, no one said a word.  
\item As soon as John left the room, we all began to speak.
\end{itemize}

The type-shifts can also apply recursively, with different adjuncts on each ‘level.’ Steedman (1997, p. 902) offers the following example.

\begin{itemize}
\item It took me two years to play the ‘Minute Waltz’ in less than sixty seconds for one hour without stopping.
\end{itemize}

Here, the activity of playing is turned into an achievement by the verbal complement and the containment adverbia, and then the durative adverbial indicates that the whole thing is an activity after all, which presumably means that the waltz is played many times.

\textsuperscript{2}The event is described as punctual, and has a conclusion.
3.3.1 Event nucleus and transitions

Moens (1987) suggested a mechanism to handle shifts in situation type, which was used in Moens and Steedman (1988) and Steedman (1997), and also adopted by Pulman (1997) and, to some extent, Kamp and Reyle (1993, pp. 557–570). An event, postulates Moens, implies a process leading up to, and ending with, the event itself plus a state resulting from it. This three part entity is called an event nucleus (Moens, 1987, p. 47; Moens and Steedman, 1988, p. 18; Steedman, 1997, p. 903, 928; Steedman, 2000, p. 12, 54).

An accomplishment, according to Moens, contains all three parts. In the proposition 'John ran a mile,' for example, the preparatory activity is the running, the culmination point is when the finishing line is reached and the consequent state is that described by the perfect 'John has run a mile.'

To explain how the situation types of verbs or verb groups can shift, Moens suggests a set of transformations.

<table>
<thead>
<tr>
<th>Old type</th>
<th>Transformation</th>
<th>New type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVITY</td>
<td>+ (culminating)</td>
<td>ACHIEVEMENT</td>
</tr>
<tr>
<td>ACCOMPLISHMENT</td>
<td>− (culminating)</td>
<td>ACHIEVEMENT</td>
</tr>
<tr>
<td>POINT</td>
<td>+ (consequent)</td>
<td>STATE</td>
</tr>
<tr>
<td>ACHIEVEMENT</td>
<td>− (consequent)</td>
<td>STATE</td>
</tr>
<tr>
<td>ACHIEVEMENT</td>
<td>+ (preparatory)</td>
<td>ACTIVITY</td>
</tr>
<tr>
<td>ACCOMPLISHMENT</td>
<td>− (preparatory)</td>
<td>ACHIEVEMENT</td>
</tr>
<tr>
<td>ACTIVITY</td>
<td>compression</td>
<td>POINT</td>
</tr>
<tr>
<td>ACCOMPLISHMENT</td>
<td>compression</td>
<td>POINT</td>
</tr>
<tr>
<td>POINT</td>
<td>iteration</td>
<td>ACTIVITY</td>
</tr>
<tr>
<td>ACHIEVEMENT</td>
<td>[progressive]</td>
<td>(progressive) STATE</td>
</tr>
<tr>
<td>POINT</td>
<td>iteration</td>
<td>(habitual) STATE</td>
</tr>
<tr>
<td>STATE</td>
<td>temporal binding</td>
<td>ACTIVITY</td>
</tr>
</tbody>
</table>

The idea is that lexical information about the verb, or verb group, assigns it a default situation type. Certain complements, such as time adverbials require specific situation types for the verb they combine with, and the transformations above can then be used to satisfy those requirements.

<table>
<thead>
<tr>
<th>Adverbial</th>
<th>Combines with</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>in / it took him</td>
<td>ACTIVITY</td>
<td>ACCOMPLISHMENT</td>
</tr>
<tr>
<td>for ACTIVITY</td>
<td>ACCOMPLISHMENT</td>
<td>ACCOMPLISHMENT</td>
</tr>
</tbody>
</table>

Most of the transformation can be made freely, but a couple requires that the verb is in the perfect or progressive form.

Moens and Steedman omits, without motivation, this transformation from their pictures (Moens, 1987, p. 45; Moens and Steedman, 1988, p. 18; Steedman, 1997, p. 903; Steedman, 2000, p. 10). It has been included here for completeness.
For Combine with Result

- perfect: ACHIEVEMENT (consequent) STATE
- progressive: ACTIVITY (progressive) STATE

When interpreting a sentence in Moens system, the default situation type of the verb must be transformed to the type that the modifiers expect.

In the following example (Moen, 1987, pp. 46, 50–51), the verb group is an accomplishment and the durative adverbial expects an activity.

John played the sonata for X time.

play the sonata ACCOMPLISHMENT
for requires ACTIVITY

There are, according to Moens, two different interpretations of this sentence pattern. The first is where the playing is interrupted, so that the conclusion point is never reached.

John played the sonata for a few minutes (without finishing it).

play the sonata minus culmination POINT → ACTIVITY

The other interpretation is where the sonata is played to completion many times. In Moens's situation type calculus, two transformation steps are necessary to arrive at the desired type.

John played the sonata (repeatedly) for eight hours.

play the sonata compression POINT → iteration → ACTIVITY

As Moens points out, the length of time expressed by the complement tends to indicate which of the two senses is more likely in any given proposition. A sonata has a more or less fixed length and playing it less than once must be quicker than playing it several times.

When the modifier contains the preposition ‘in,’ the situation type must be converted to an accomplishment. The following examples (Moen, 1987, pp. 51, 53) has the verb ‘arrive’ which is, lexically, an achievement, and those two situation types differ in that the former, in Moens's event nucleus, includes the preparatory activity.

arrive ACHIEVEMENT

So the interpretation of ‘arrive’ + ‘in’ adverbial involves adding the movement of the train (preparatory activity) to the proposition so that it becomes an accomplishment which is what the adverbial expects.

The train arrived in 5 minutes.

ACHIEVEMENT plus preparatory activity → ACCOMPLISHMENT

If the adverbial is of the durative type, the verb must be turned into an activity. Removing the consequent state, says Moens, allows it to be iterated thereto, via POINT.

* The train arrived for several minutes.
For months, the train arrived late.

ACHIEVEMENT minus consequent STATE → POINT → iteration → ACTIVITY
As in the earlier example, the magnitude of the time specification constrains the interpretation and, in this case, rules out the first sentence above.

One could say that Moens only goes halfway towards explaining how the situation types are determined by the syntactic complements. Time adjuncts are factors in his computations, but objects are not, and no explanation is offered for the different type assignments for ‘play’ in the examples below, which are all considered lexical by Moens.

\[
\begin{align*}
\text{play} & \quad \text{ACTIVITY} \\
\text{play the sonata} & \quad \text{ACCOMPLISHMENT} \\
\text{play sonatas} & \quad \text{ACTIVITY}
\end{align*}
\]


### 3.3.2 Generative lexicon

Pustejovsky’s (1995a) theory of a ‘generative lexicon’ includes situation type-shifting. Here, the number of situation types is limited to three, but a proposition can contain several instances of them. The following account is slightly abbreviated, and some liberties have been taken with the forms of the representations, in the interest of clarity.

In a generative lexicon, there are three situation types: state, activity, and transition (Pustejovsky, 1995a, p. 68). These can be combined using binary operators, so that the situation expressed by a proposition can involve any number of instances of the three basic types. The combination operators that will be used here are the following two.\(^4\)

\[
\begin{align*}
\text{seq}(e_1, e_2) & \quad \text{Sequence: first } e_1, \text{ then } e_2. \\
\text{sim}(e_1, e_2) & \quad \text{Simultaneous: } e_1 \text{ and } e_2 \text{ take place at the same time.}
\end{align*}
\]

As an example, Pustejovsky (1995a, p. 71) offers the verb ‘build’ which, in this analysis, involves an activity — the building — and a resulting state, of something existing after having been built.

\[
\begin{align*}
\text{build} & \\
\text{Situations} & \quad \text{activity}(e_1), \text{seq}(e_1, e_2), \text{state}(e_2) \\
\end{align*}
\]

When the situation described by a verb is separated into components like this, adjuncts can be analysed as referring to one of them only.

- The house was built in a week. \(\text{activity}(e_1)\)
- The house was built at night. \(\text{activity}(e_1)\)
- The house was built by the prisoners. \(\text{activity}(e_1)\)
- The house was built for the priest. \(\text{state}(e_2)\)
- The house was built on an island. \(\text{Both}\)
- The house was built of stone. \(\text{Both}\)

\(^4\)In Pustejovsky (1995a) there is a third, used for situations involving causativity, but we do not need it.
The last two adjuncts refers to both the activity and result, since the building location and material influences the nature of the work as well as of the house itself.

In any constellation of situation types, at least one of them must, according to Pustejovsky (1995a, p. 72), be 'headed.' This property indicates semantic prominence, and provides an avenue back to the distinctions made in section 3.2.4 and section 3.2.6. The difference between achievement and accomplishment, in this view, is which component serves as head. The various combinations of operator and headedness distribution are listed below (Pustejovsky, 1995a, p. 73).

\[
\begin{array}{ll}
\text{seq} (e_1, e_2) & \text{ACCOMPLISHMENT} & \text{build} \\
\text{seq} (e_1, e_2) & \text{ACHIEVEMENT} & \text{arrive} \\
\text{sim} (e_1, e_2) & \text{give, take} & \\
\text{sim} (e_1, e_2) & \text{buy} & \\
\text{sim} (e_1, e_2) & \text{sell} & \\
\text{sim} (e_1, e_2) & \text{marry} & \\
\end{array}
\]

This analysis might not be totally convincing. It is not, for example, made clear why an achievement necessarily involves an activity. While an arrival must perhaps be preceded by some amount of travelling, other achievements like 'discover' and 'realise' do not seem to have this structure.

But these are minor objections. Pustejovsky's theory does not require verbs to have exactly two components, and it is, for instance, possible to give the various verbs that are traditionally labelled achievements different representations.

As an example of how generative lexicon theory explains type-shifting, Pustejovsky (1995a, pp. 122–125) uses the verb 'bake' which can be either an activity or an accomplishment.

Mary baked all day. \textit{ACTIVITY}
Mary baked a cake. \textit{ACCOMPLISHMENT}

The following representations are used for the two words 'bake' and 'cake.'

\begin{tabular}{ll}
\text{bake} & \text{cake} \\
Situations & activity(e) \quad & \text{Situations} & state(e) \\
Arguments & animate(x_1), \quad & \text{Arguments} & food(x_1) \\
\quad & mass(x_2) & \text{\quad default(mass(x_2))} \\
Agentive & bake(e, x_1, x_2) & \text{Agentive} & bake(e_1, x_2) \\
\quad & \text{Constitutive} & \quad & \text{exists(e_3, x_1)} \\
\quad & \text{Telic} & \quad & \text{act(e_2, x_1)} \\
\end{tabular}

By default, 'bake' is, according to Pustejovsky (1995a, p. 123) an activity. But when combined with 'cake,' as in the second sentence above, the two values for 'agentive' are unified, thereby linking the activity and the state together via the 'formal' value in the representation of 'cake.'
bake a cake

Situations  \( \text{activity}(e_1), \text{seq}(e_1, e_2), \text{state}(e_2) \)
Arguments  \( \text{animate}(x_1), \text{artifact}(x_2), \text{material}(x_3) \)
Agentive  \( \text{bake}(e_1, x_1, x_3) \)
Formal  \( \text{exists}(e_1, x_2) \)

The expression ‘\( \text{seq}(e_1, e_2) \)’ is the one that Pustejovsky (1995a, p. 73) calls accomplishment.

According to Pustejovsky (1995a, p. 16, 82; 1995b, pp. 80–81), a result of this approach is a simple solution to the ‘imperfective paradox.’ This paradox consists of the fact that the imperfective sentence ‘Mary was baking a cake’ does not entail that Mary did bake a cake, since she might have given up half-way through. Dowty (1979, pp. 133–138) first drew attention to this problem, and it has been thoroughly discussed in the literature (Cooper, 1985, pp. 19–24; Moens, 1987, pp. 63–68; Pustejovsky, 1991, p. 51; Lascarides, 1991; Verkuyl, 1993, pp. 44–45, 206–209; White, 1994, pp. 28–39; Pustejovsky, 1995a, p. 16; Galton, 1995, p. 219; Steedman, 1997, pp. 899, 914–915, 925; van Eijck and Kamp, 1997, p. 231). The answer to the paradox, says Pustejovsky (1995a, p. 82), is that an imperfective, which is one possible meaning of the progressive aspect, asserts only the activity part of the proposition.\(^5\)

baking a cake

Situations  \( \text{activity}(e_1) \)
Arguments  \( \text{animate}(x_1), \text{material}(x_3) \)
Agentive  \( \text{bake}(e_1, x_1, x_3) \)

A possible problem with this analysis is that it seems to ignore the intended goal, i.e. producing a cake, since it only says that some objects (flour, butter) were acted on in a certain way, without distinguishing the situation from the case where these things are done without any intention to finish the cake.

Considered as a theory of tense and aspect, Pustejovsky’s generative lexic is quite complementary to Moens’s transition system (section 3.3.1). The latter does not explain how complements can force the situation types to shift, while it accounts for the meaning of progressive and perfect, something that Pustejovsky does not.

3.4 Formal representation of time

Long before computer programming, there was formal logic (Bocheński, 1951, 1970). In antiquity, logical propositions always had a temporal component, as Aristotle makes clear in On Interpretation (Aristotle, 1983).

A verb is a sound which not only conveys a particular meaning but has a time-reference also. (16b6)

Of all propositions a verb or a tense of a verb must form part. The definition, for instance, of ‘man,’ unless ‘is,’ ‘was’ or ‘will be’ is added or something or other of that kind, does not constitute a proposition. (17a11)

\(^5\)See page 100 for a similar analysis.
His contemporaries the Stoics agree (Mates, 1961, p. 36).

The task of understanding the theory of Diodorus Cronus is made difficult by the fact that he apparently thought of propositions as though they contained time-variables. His examples always include expressions like 'it is day,' and he says that these are true at certain times and false at others, or that they become true and become false.

The tone of this quote reveals the position in twentieth century mathematical logic, that propositions are timeless. But the ancient view persisted throughout the middle ages, as in William of Ockham (Bocheński, 1970, pp. 230–231).

Now we must see how syllogisms are to be made from propositions about the past and the future. Here it is to be known that when the middle term is a common term, if the subject of the major supposes for things which are, the minor should be about the present and not the future or the past [...] After the Renaissance, logicians lost interest in time and tense (Prior, 1957, p. 104), and when logic turned into a branch of mathematics in the nineteenth century, temporal considerations were no longer relevant. In the twentieth century, the logicians returned to time (Prior, 1957, 1967, 1968), and the first representations used were two operators P and F standing for 'has been the case,' and 'will be the case,' respectively (Prior, 1957, p. 8).

\[
\begin{align*}
  p & \quad \text{God is dead.} \\
  P \, p & \quad \text{God was dead.} \\
  F \, p & \quad \text{God will be dead.}
\end{align*}
\]

Later, this was extended to allow the specification of time intervals using constants, variables, and quantifiers (but no predicates). This system is known as 'metric tense logic' (Prior, 1968, p. 88).

\[
\begin{align*}
  P \, n \, p & \quad \text{God was dead \( n \) time ago.} \\
  F \, n \, p & \quad \text{God will be dead in \( n \) time.}
\end{align*}
\]

Using these operators, approximations can then be defined of other temporal meanings in natural language, such as the conjunctions 'since' and 'until' (van Benthem, 1995, pp. 279–280).

\[
\begin{align*}
  S \, p \, q & \quad \text{`q has been true since \( p \) was true.'} \\
  U \, p \, q & \quad \text{`q will be true until \( p \) is true.'}
\end{align*}
\]

But definitions like these are mainly useful if the meaning representation formalism is limited to predicate logic. Encoding the meaning of 'since' and 'until' in a computer program can be done in many different ways, and the particular solution is not of great interest. It is only one single technical problem out of an infinite multitude.

Much of the work in temporal logic has been devoted to introspective pursuits like proving soundness and completeness, which are also not of great relevance for computer implementations, unless predicate logic is the only formalism used to represent meaning.

\[6\text{Unless a term refers to time explicitly as in `\( \neg \exists x \text{remembers}(x, \text{18 April 1775}) \).'}\]
A fundamental question that is still, to some extent, open (van Benthem, 1995, pp. 280–299; Galton, 1995, pp. 200–210), about the formal representation of time, is whether the atomic units are points or intervals. In a famous paper, Allen (1984) claims that only the latter are necessary, but Galton (1990) argues persuasively that both are needed, and this seems to have been accepted by Allen (1995, pp. 406–407). The original argument (Allen, 1984, pp. 127–128) for eliminating instantaneous points from the formal representation is that there is no natural interpretation of the basic case where a property holds during an interval $I_1$, changes truth-value at a point $P$, and then does not hold during the interval $I_2$, where $I_1$ and $I_2$ meet at $P$.

<table>
<thead>
<tr>
<th>$P$ is part of both $I_1$ and $I_2$</th>
<th>Inconsistent$^8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$ is part of $I_1$ only</td>
<td></td>
</tr>
<tr>
<td>$P$ is part of $I_2$ only</td>
<td>The choice is arbitrary</td>
</tr>
<tr>
<td>$P$ is not part of either $I_1$ or $I_2$</td>
<td>Counter-intuitive</td>
</tr>
</tbody>
</table>

Galton (1990, pp. 166–167) claims that the notion of a point being part of an interval is meaningless and that the relevant relationships are: a point limiting an interval, and being within it. For the example above we get the following.

$P$ limits $I_1$ (At the end)
$P$ limits $I_2$ (At the beginning)
$P$ is not within either $I_1$ or $I_2$ (but it is within the join of $I_1$ and $I_2$)

Instants, says Galton (1990, pp. 161–165, 167–169), are also necessary for representing continuous change, whose essence is that the changing object passes through a continuous range of states, each of which obtains for an isolated instant (Galton, 1990, p. 167). In Allen’s system there are no instantaneous states (Allen, 1984, p. 128) and consequently it cannot distinguish between, for example, an object being at rest and its having zero velocity while in motion.$^9$

Galton’s solution is to have two types of states (Galton, 1990, pp. 169–170).

<table>
<thead>
<tr>
<th>Points</th>
<th>State of motion$^{10}$</th>
<th>State of position</th>
</tr>
</thead>
<tbody>
<tr>
<td>If it holds</td>
<td>Cannot hold at isolated instants.</td>
<td>Can hold at isolated instants.</td>
</tr>
<tr>
<td>Then</td>
<td>At an instant</td>
<td>Throughout an interval</td>
</tr>
<tr>
<td>examples</td>
<td>It must hold throughout some interval within which the instant fails.</td>
<td>It must hold at the limits of it.</td>
</tr>
<tr>
<td>Generally,</td>
<td>A body’s being at rest or in motion.</td>
<td>A body’s being in a particular position, or moving at a particular speed, or in a particular direction.</td>
</tr>
<tr>
<td>Any state of affairs which consists of some continuously variable quantity’s either remaining fixed or undergoing a change of value.</td>
<td>Any state which consists of some continuously variable quantity’s assuming a particular value.</td>
<td></td>
</tr>
</tbody>
</table>

Being at rest is here considered a state of motion, while having zero velocity is a state of position (Galton, 1990, p. 169). This has the intuitive consequences that an object can have zero velocity for an instant while being at rest can only hold during an interval. Additionally, of course, being at rest happens to imply that the velocity is zero.

$^7$Note that the ‘time points’ of Rédei (1947, section 3.1), like the ‘punctual’ situation types (section 3.2.6), are not necessarily, or even typically, instantaneous.

$^8$Since the property that holds throughout $I_1$ but not $I_2$ will both hold and not hold at $P$.

$^9$Like a ball has at the topmost point of its trajectory when thrown vertically upwards.

While Allen (1984) has no points, he allows arbitrarily small intervals, and any interval can be divided into sub-intervals (Allen, 1984, p. 128). The relation between the interval during which a situation occurs (I below), and its sub-intervals, is then used by Allen (1984, pp. 131–132) to classify situation types.

<table>
<thead>
<tr>
<th>Situation type</th>
<th>Allen (1984)</th>
<th>Sub-intervals</th>
<th>Culmination</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATE</td>
<td>'property'</td>
<td>Also holds throughout all sub-intervals of I.</td>
<td>—</td>
</tr>
<tr>
<td>ACTIVITY</td>
<td>'process'</td>
<td>Also occurs over some sub-interval of I.</td>
<td>No</td>
</tr>
<tr>
<td>ACCOMPLISHMENT</td>
<td>'event'</td>
<td>There is no sub-interval of I over which it occurs.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Note that the punctual situation types, points and achievements (section 3.2.6), are not recognised by Allen (1984). The characterisation of activities above, that they take place during some, but not necessarily all, sub-intervals, is, as Allen (1984, p. 135) admits and Galton (1990, pp. 180–181) argues at some length, odd. Any activity will involve some form of change, like movement. Otherwise it would be a state. But on a given level of abstraction, it is homogeneous. The activity of ‘running,’ for example, can involve stationary stretching and occasional pausing, where the physical movements are not the same as in the actual running. But this is a question of what particular sense of the word is used, and not a problem of temporal logic.

While Allen (1984, pp. 128, 132–133) uses two formal expressions to connect situations and times, Galton (1990) suggests nine. The use of instants leads Galton (1990, p. 171) to three different relations: at an instant, during an interval, and throughout an interval. Punctual events, which Allen also does not have, and a ‘progressive operator’ (Galton, 1990, pp. 178, 181–182) gives the following three times three cases:

<table>
<thead>
<tr>
<th>State holds at instant.</th>
<th>State holds at some time during interval.</th>
<th>State holds throughout interval.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Punctual event occurs</td>
<td>(Punctual or durative) event occurs within interval.</td>
<td>Durative event occurs throughout interval.</td>
</tr>
<tr>
<td>at instant.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Durative event is in</td>
<td>Durative event is in progress at some point during interval.</td>
<td>Durative event is in progress throughout interval.</td>
</tr>
<tr>
<td>progress at instant.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In the system described in the Appendix, punctual and durative situations are distinguished (see page 95), but the times are symbolic and have no specified duration, so no position has been adopted on the issue of instants versus intervals.

11This is a round-about way of stating that an accomplishment cannot be said to have occurred until it is finished. Compare the ‘imperfective paradox,’ page 42.
12In his later work, Allen (1995, p. 406) mentions achievements but they are not used in the discourse interpretation algorithm (section 4.7).
13Stated ‘hold’ and events occur.’
14Compare LOCATION, LOCATION in a SPAN, and DURATION on page 23 and in section 6.6.
15See section 3.2.6.
16I.e. imperfective meaning; applies to durative events only (Galton, 1990, p. 182).
4 Beyond the sentence

In discourse, the times, which are typically under-specified, and events mentioned in each sentence must be integrated into one representation of the whole text. In the simplest case, two sentences describe events that are simultaneous, or the Reichenbachian (1947) ‘reference point’ in one sentence co-refers temporally with events in a second sentence (section 4.2). Another common phenomenon is that the sequence of sentences reflects the temporal sequence of the events they describe. This is termed ‘default narrative progression’ and is discussed in section 4.3, together with conceptual relations that can also influence the temporal meaning of sentences in discourse. The events described in a text can also belong to more than one period of time, as when a narrative shifts back and forth between situations occurring on two separate days a month apart.\footnote{1} In the example (section 1.1), three events\footnote{2} are understood to have taken place some time before the journey, and this is marked by perfect aspect (see also section 4.1 below).

Besides recognising all the linguistic forms discussed in chapter 2, and computing the correct meanings, a computational system for interpreting temporal information in discourse ought to specify the operations involved with enough formality to allow implementation in a programming language. Among the theories in the literature, this abstract ideal is probably most closely approximated by Allen (1995), which describes algorithms for all the traditional parts of computational linguistic analysis: parsing, semantic interpretation, knowledge representation, discourse structure and speech act recognition. A systematic and credible attempt to express the temporal meanings available in English computationally is Cutrer (1995), and to a lesser extent Dinsmore (1991), but these are less formal than Allen. The system described in Hitzeman et al. (1995) is eminently formal, being implemented in Prolog, but is limited in coverage and the amount of useful information produced. Kamp and Reyle (1993), finally, describe a discourse interpretation algorithm that produces logical formulae representing the meaning of the text.

<table>
<thead>
<tr>
<th>Discourse representation theory</th>
<th>Semantic theory expressed in predicate logic; discourse interpretation algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discourse grammar</td>
<td>Implemented system for interpreting temporal information in discourse</td>
</tr>
<tr>
<td>Partitioned representations</td>
<td>Informal description of approach to meaning representation</td>
</tr>
<tr>
<td>Segments and stack</td>
<td>Complete discourse analysis algorithm</td>
</tr>
</tbody>
</table>

These four are the major systems for interpreting tense and aspect in the recent literature. As complete algorithms they are quite diverse, but there is some overlap in the technical details. The idea of using partitioned representations, for example, is also used by Allen (1995), and discussed by Kamp and Reyle (1993) and Hitzeman et al. (1995), but under different names.

\footnote{1}{Or the two, or more, situations can have different modality, as when general, timeless, statements occur in a narrative text.}

\footnote{2}{Hayward’s giving an introduction; Philip’s writing a letter and taking a room.}
4.1 The example text

For comparison, the same example — taken from W. S. Maugham’s *Of Human Bondage*, quoted in Reichenbach (1947, p. 288), discussed in section 1.1, and reproduced here — is analysed in each of the four systems.

But Philip ceased to think of her a moment after he had settled down in his carriage. He thought only of the future. He had written to Mrs. Otter, the maîtresse to whom Hayward had given him an introduction, and had in his pocket an invitation to tea on the following day. When he arrived in Paris he had his luggage put on a cab and trundled off slowly through the gay streets, over the bridge, and along the narrow ways of the Latin Quarter. He had taken a room at the Hotel des Deux Ecoles, which was in a shabby street off the Boulevard du Montparnasse; it was convenient for Amitrano’s School at which he was going to work.

Most of the relevant phenomena are represented here. There is perfect aspect, temporal conjunctions (after, when) and adjuncts (on the following day, slowly), aspectual verbs (cease), modal constructions (be going to), and a tense-less clause (put). There is even a verb particle with aspectual meaning, ‘trundle off.’ A paraphrase of the final part of that sentence would be ‘Philip trundled slowly along the narrow ways,’ with no particle. Notable absences are present tense and progressive aspect. The events and states involved are listed below, in the order they are understood to occur.

<table>
<thead>
<tr>
<th>Event</th>
<th>Timeframe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hotel is in a shabby street.</td>
<td>Holds from remote past to remote future.</td>
</tr>
<tr>
<td>Hotel is convenient.</td>
<td></td>
</tr>
<tr>
<td>Hayward gives introduction.</td>
<td>Events in near past.</td>
</tr>
<tr>
<td>Philip writes letter.</td>
<td></td>
</tr>
<tr>
<td>P. takes a room at hotel.</td>
<td></td>
</tr>
<tr>
<td>Invitation is in P.’s pocket.</td>
<td>State that holds throughout narrative.</td>
</tr>
<tr>
<td>P. settles down in carriage.</td>
<td>Events in narrative.</td>
</tr>
<tr>
<td>P. ceases to think of her.</td>
<td>(a moment later)</td>
</tr>
<tr>
<td>P. arrives in Paris.</td>
<td></td>
</tr>
<tr>
<td>Luggage is put on a cab.</td>
<td></td>
</tr>
<tr>
<td>P. trundles off slowly.</td>
<td></td>
</tr>
<tr>
<td>P. meets Mrs. Otter for tea.</td>
<td>Event that will take place one day later.</td>
</tr>
<tr>
<td>P. works at Amitrano’s School.</td>
<td>Repetitive event starting in near future.</td>
</tr>
</tbody>
</table>

Reconstructing this sequence is a necessary part of comprehending the text. Not all of it can be derived from tense and aspect, however. The relative timelessness of the hotel’s properties, for instance, is not stated expressly. From the linguistic evidence alone, a possible interpretation would be that Philip will work only one day and that, as soon as he has checked out in the morning, the hotel somehow leaves the shabby street and moves to a better neighbourhood.

4.2 Time points

Although Reichenbach (1947; section 3.1) does not discuss texts involving more than one sentence, his examples include complex sentences with connectives, and
his account can be readily generalised to multi-sentence discourse, as Comrie (1981) and Dinsmore (1982), show. In Reichenbach’s system, the time points of multiple sentences, and also of multiple propositions in the same sentence, interact in accordance to the following two principles.

- Time adjuncts refer to the reference point in the sentence they modify.
- By default, all reference points are the same.

If there are no time adjuncts, all the reference points must be identical. Since all the statements are made at the same time, the speech points must also be the same. These examples are from Reichenbach (1947, p. 293).

| I had mailed the letter | E - R | - S |
| when John came | R,E | - S |
| and told me the news. | R,E | - S |

These principles explain why the following sentence is ungrammatical.

* I had mailed the letter.  
  when John has come.

The use of present perfect (anterior present) puts the reference point in the second phrase at the same time as the speech points, so the two reference points are separate, which is not acceptable.

Riechenbach’s first maxim, that time adjuncts refer to the reference point, does not have absolute validity, however. Comrie (1981) gives this example.

| You say that you will finish at six. | S | - | R,E |
| You’re slow. | S | - | E | - | R |
| I’ll have finished at five. | S | - | E | - | R |

The adjunct in the last sentence refers to the event, not the reference point.

### 4.3 Narrative progression

Even without aspects, modal verbs, and time adjuncts, temporal meaning can be expressed. Allen (1995, p. 517) presents the following example, where the events take place in the same order they are described.

| Jack went to the store. | EVENT |
| He bought some roses. | EVENT |

From this, a tentative generalisation could be made that the situations referred to by verbs in simple tense always follow in sequence. But this holds for events only; states typically overlap (Allen, 1995, p. 517; Hitzen et al., 1995, p. 253; Kamp and Reyle, 1993, pp. 522–523, 543–545; van Eijck and Kamp, 1997, p. 228).

| Jack went to the store. | EVENT |
| He felt well again. | STATE |
And if the sentences are conceptually related, other possibilities exists (Allen, 1995, pp. 517–520).

Jack went to the store.
His mother sent him. (Causation, occurred before)

Jack went to the store.
He walked along the river. (Elaboration, same event)

Jack went to the store.
He finally remembered where it was. (Enablement, occurred before)

Nevertheless, in the absence of any other information, the default seems to be that the sentential order mirrors the actual order of events.

4.4 Discourse representations

One of the most well-known attempts to describe the technicalities of interpreting tense, aspect, and time adjuncts in multi-sentence text is ‘discourse representation theory’ (Kamp and Reyle, 1993; van Eijck and Kamp, 1997). This theory is closely based on predicate logic but uses partitions (section 4.6) to represent negation, conditionals, and plurals. In general terms, a set of logical formulae is built up incrementally from the sentences in a discourse, with noun phrases mapped to variables\(^3\) and verbs to predicates. Tensed verbs also introduce variables that stand for states, events and time points. Kamp and Reyle (1993, pp. 559-560) discusses Vendler (1967) and his classification of situation types (section 3.2), but in practice they only distinguish between states and events. Something very similar to Moens’s event nucleus (Moens, 1987) is also discussed, but, again, without being made used of.

4.4.1 Verb phrases and time points

There are some restrictions on the range of verb groups recognised in discourse representation theory: present tense events are not allowed (Kamp and Reyle, 1993, p. 544), and the meaning of the progressive aspect is essentially ignored (Kamp and Reyle, 1993, p. 576). Perfect aspect is analysed as always being a state (Kamp and Reyle, 1993, pp. 559, 572-573), and the perfect of an event is taken to mean the state that results from it. As mentioned in section 3.1, Kamp and Reyle (1993) use time points (Reichenbach, 1947) to analyse tense and aspect, but their classification (Kamp and Reyle, 1993, p. 601) is unorthodox.

\(^3\)A noun phrase always introduces a new variable, and might also introduce formulae involving that variable (Kamp and Reyle, 1993, pp. 61–62)
Reichenbach's (1947) terms are used here instead of the original ones which are given on page 29. Their meaning is repeated below.

<table>
<thead>
<tr>
<th></th>
<th>ANTERIOR</th>
<th>Location time before reference point.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SIMPLE</td>
<td>Reference point same as location time.</td>
</tr>
<tr>
<td>POSTERIOR</td>
<td>Location time after reference point.</td>
<td></td>
</tr>
<tr>
<td>PAST</td>
<td>Reference point before utterance time.</td>
<td></td>
</tr>
<tr>
<td>PRESENT</td>
<td>Reference point same as utterance time.</td>
<td></td>
</tr>
</tbody>
</table>

Attentive readers will note that ambiguity, to some extent, is manifest in the mapping from verb groups to reference point constraints. Kamp and Reyle (1993, p. 596) observe that, in a context where the whole narrative lies in the past, the second sentence below is not good.

Bill had come home at seven. Now he was writing a letter.

* Now he wrote a letter.

Their explanation for this is that the time adjunct 'now' must combine with a progressive, which, according to them, is always a state. The classification of a sentence like 'He wrote a letter' is then arrived at by a *reductio ad absurdum*: the time adjunct 'now' always refers to the reference point (Kamp and Reyle, 1993, pp. 599, 612; Reichenbach, 1947, p. 294), so a verb group it modifies must be *simple* (event time same as reference point); since (it is assumed) it cannot combine with *simple past*, as in the example above, this verb group must belong to another category. The only remaining possibility, say Kamp and Reyle (1993, p. 597), is *ANTERIOR PRESENT*.

Now he was writing a letter. **SIMPLE PAST**

* Now he wrote a letter.

He wrote a letter. **ANTERIOR PRESENT**

To be on the safe side, Kamp and Reyle (1993, pp. 596–597) deny inclusion in the *simple class* not just to verb groups modified by 'now,' but to events in general, so they, if in the past tense, must always be *ANTERIOR*.

The greatest difference, time point-wise, between discourse representation theory and Reichenbach is the analysis of perfect aspect. Kamp and Reyle (1993, pp. 598–599) arrive at three possible categorisations of past perfect. Two of

---

4Kamp and Reyle (1993, pp. 500–501) call this past while it is normally called past progressive.
them are the same as for simple past, but also marked as ‘perfect.’ This binary feature (Kamp and Reyle, 1993, p. 559) implies that the phrase is treated as a state, but besides that it has no practical relevance. The third category is anterior past but without the ‘perfect’ tag. Any mechanical method of assigning the right analysis to a given verb group is not presented in Kamp and Reyle (1993).

In addition to Reichenbachian time points (section 3.1), Kamp and Reyle use a narrative point (Kamp and Reyle, 1993, pp. 594-595) to keep track of the default narrative progression (section 4.3). These two points, and the set of logical formulas, are computed by a discourse interpretation algorithm which uses the classification of the verb groups in each new sentence. Non-temporal content, such as typical noun phrases, is also discussed in Kamp and Reyle (1993), but for clarity we will ignore the details here.

4.4.2 Discourse interpretation algorithm

Discourse interpretation algorithm (temporal information)
(Kamp and Reyle, 1993, pp. 543, 545, 602, 610)

1. Add new time referent $t$ (location time).
   If there is an adverb, add condition ‘Adverb($t$).’

2. Choose narrative point $(NP)$.
   $(NP) = \text{time of the most recent state or event of same 'tense.'}$
   (Kamp and Reyle, 1993, p. 545)

3. **Situation type**
   
   - **STATE** Add new state referent $s$.
     
     - If $\text{SIMPLE}$ then add condition ‘$s$ includes $t$.’
     - else add condition ‘$s$ overlaps $t$.’
   
   - **EVENT** Add new event referent $e$.
     
     - Add condition ‘$t$ includes $e$.’

4. If sentence is not discourse-initial:
   
   - **STATE** Add condition ‘$s$ includes NP.’
   - **EVENT** Add condition ‘NP precedes $e$.’

   (Kamp and Reyle, 1993, p. 545)

5. Choose reference point $(RP)$.
   
   - If $RP$ is before utterance time then it ‘follows from the context.’

   (Kamp and Reyle, 1993, p. 610)

6. **Category**
   
   - **ANTERIOR** Add condition ‘$t$ precedes $RP$.’
   - **SIMPLE** Add condition ‘$t = RP$.’
   - **POSTERIOR** Add condition ‘$RP$ precedes $t$.’

A weak point in the algorithm is step 5, choice of reference point. The rules only specify that it should be either same as the utterance time or before it. In the latter case there might obviously be many time referents to choose from and

---

$^5$They have a different name for it.

$^6$‘Tense, here, means past or ‘future.’ Perfect aspect is ignored.
Kamp and Reyle (1993, pp. 601, 606, 608) point out, repeatedly, that they have no way to pick the right one mechanically. Interestingly, the way the algorithm is formulated, the available information about the relation between reference point and location time is not actually used to help constrain the choice of reference point.

4.4.3 The example

Of the sentences in the example (section 4.1), two contain temporally relevant syntactic constructions that are not discussed by Kamp and Reyle (1993): the aspectual verb ‘cease’ and the non-finite verb ‘put,’ and those will be ignored. For the others, the tense/aspect category and situation type (state or event) is listed below. Compound sentences have been paraphrased.

Philip ceased to think of her. —
P. had settled down in his carriage. Past perfect event
P. thought only of the future. Past state
P. had written to Mrs. Otter. Past perfect event
Hayward had given P. an introduction. Past perfect event
P. had an invitation to tea. Past state
P. arrived in Paris. Past event
P. had his luggage put on a cab. —
P. trundled off slowly through the streets. Past event
P. trundled slowly over the bridge. Past event
P. trundled slowly along the narrow ways. Past event
P. had taken a room at the hotel. Past perfect event
The hotel was in a shabby street. Past state
The hotel was convenient for the school. Past state
P. was going to work at the school. ‘Past future’ state

As mentioned above, both past and past perfect are ambiguous in their mapping to discourse representation theory’s reference point constraints. Reichenbach (1947) analyses the former as simple past (see the table on page 28) and Kamp and Reyle (1993) allows that possibility for sentences that are modified by ‘now.’ For other cases, and for all eventive verb phrases, they prefer anterior present.

Now she was happy. SIMPLE PAST
He wrote a letter (yesterday). ANTERIOR PRESENT

Past perfect can have the same classification as past, but with the added label ‘perfect,’ and there is a third possibility, anterior past.

Now she had sent off her proposal. SIMPLE PAST (perfect)
She had sent it off earlier. ANTERIOR PRESENT (perfect)
Fred arrived at ten. He had set off at six. ANTERIOR PAST

From the examples in Kamp and Reyle (1993, pp. 596–599), it seems that time adjuncts is the main factor in deciding which interpretation to use. Since the adjunct ‘now,’ according to Kamp and Reyle (1993, pp. 599, 612), always refers to the reference point, the phrase it modifies must be simple past. The choice, in other cases, between anterior present and anterior past is less obvious. Reichenbach (1947) classified past perfect as the latter (see the table
on page 28, and Comrie (1981) agrees with that (section 4.2) but claims, in contrast to Kamp and Reyle, that adjuncts can modify the event time as well as the reference time.

Simple past event phrases, in discourse representation theory, can only be anterior present. For states, simple past, which is what Reichenbach (1947) suggested, is used.

Philip ceased to think of her.  —

P. had settled down in his carriage.  SIMPLE PAST (perfect) STATE
P. thought only of the future.  SIMPLE PAST STATE
P. had written to Mrs. Otter.  ANTERIOR PRESENT (perfect) EVENT
Hayward had given P. an introduction.  ANTERIOR PRESENT (perfect) EVENT
P. had an invitation to tea.  SIMPLE PAST STATE
P. arrived in Paris.  ANTERIOR PRESENT EVENT
P. had his luggage put on a cab.  —
P. trundled off slowly through the streets.  ANTERIOR PRESENT EVENT
P. trundled slowly over the bridge.  ANTERIOR PRESENT EVENT
P. trundled slowly along the narrow ways.  ANTERIOR PRESENT EVENT
P. had taken a room at the hotel.  ANTERIOR PRESENT (perfect) EVENT
The hotel was in a shabby street.  SIMPLE PAST STATE
The hotel was convenient for the school.  SIMPLE PAST STATE
P. was going to work at the school.  POSTERIOR PAST EVENT

Since the first phrase in the text, 'ceased to think,' is not included, the first sentence given to the algorithm is in past perfect. For a human reader, that might sound odd, but, as will be seen below, this does not seem to be a problem for Kamp and Reyle’s interpretation algorithm. A full chronicle of it's application on these sentences is provided in the next section.

4.4.4 Details

The original text of the example is on page 47, and the paraphrase used here is shown on page 52. Interpretation, in discourse representation theory, consists of the following steps, for each sentence.

- The situation type and tense/aspect category of the verb group is determined. All categories used by Kamp and Reyle (1993) are listed in the table on page 50.
- Application of the interpretation algorithm (see page 51) yields new variables and formulas.

Both of these steps provide considerable latitude for individual interpretations, as discussed in section 4.4.

Narrative and reference point are defined by rules in the algorithm on page 51, which also provides the rule numbers.

<table>
<thead>
<tr>
<th>Sentence</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Philip had settled down in his carriage.</td>
<td>(Past perfect event)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SIMPLE PAST (perfect) STATE</th>
<th>STATE s₁: P. has settled down.</th>
</tr>
</thead>
<tbody>
<tr>
<td>New variables: t₁, s₁</td>
<td>s₁ includes t₁ (rule 3)</td>
</tr>
<tr>
<td>Ref. point = t₁ (rule 5)</td>
<td>t₁ = t₁ (rule 6)</td>
</tr>
</tbody>
</table>
Since the first sentence is discourse-initial, there is no narrative point. Reference point must be \( t_1 \) as there is no alternative.

**Sentence Two**

<table>
<thead>
<tr>
<th>Philip thought only of the future. (Past state)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Simple past state</strong></td>
</tr>
<tr>
<td>New variables: ( t_2, s_2 )</td>
</tr>
<tr>
<td>Narr. point = ( t_1 ) (rule 2)</td>
</tr>
<tr>
<td>Ref. point = ( t_1 ) (rule 5)</td>
</tr>
</tbody>
</table>

Here, the only information available about the reference point is that it should be before the utterance time (because the phrase is past). But this is true for the whole discourse. We choose \( t_1 \), although \( t_2 \) is equally possible, given the description in Kamp and Reyle (1993).

**Sentence Three**

<table>
<thead>
<tr>
<th>Philip had written to Mrs. Outer. (Past perfect event)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Anterior present (perfect) state</strong></td>
</tr>
<tr>
<td>New variables: ( t_2, s_3 )</td>
</tr>
<tr>
<td>Narr. point = ( t_2 ) (rule 2)</td>
</tr>
<tr>
<td>Ref. point = utt. time (rule 5)</td>
</tr>
</tbody>
</table>

Writing is an event, so the phrase must be anterior present rather than simple past. This, or present in particular, means that the reference point is the utterance time.

**Sentence Four**

<table>
<thead>
<tr>
<th>Hayward had given Philip an introduction. (Past perfect event)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Anterior present (perfect) state</strong></td>
</tr>
<tr>
<td>New variables: ( t_4, s_4 )</td>
</tr>
<tr>
<td>Narr. point = ( t_3 ) (rule 2)</td>
</tr>
<tr>
<td>Ref. point = utt. time (rule 5)</td>
</tr>
</tbody>
</table>

The fourth sentence is similar to the third, with reference point equal to utterance time.

**Sentence Five**

<table>
<thead>
<tr>
<th>Philip had an invitation to tea on the following day. (Past state)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Simple past state</strong></td>
</tr>
<tr>
<td>New variables: ( t_5, s_5 )</td>
</tr>
<tr>
<td>Narr. point = ( t_4 ) (rule 2)</td>
</tr>
<tr>
<td>Ref. point = ( t_2 ) (rule 5)</td>
</tr>
</tbody>
</table>

For the fifth sentence we pick \( t_2 \) as reference point, since that seems to make most sense. There is nothing in Kamp and Reyle (1993) to help determine this, and, from a formal point of view, all previous time variables are equally probable.
Sentence Six

Philipp arrived in Paris. (Past event)

**ANTERIOR PRESENT EVENT**

EVENT $e_6$: P. arrives.

New variables: $t_6, e_6$

Narr. point $= t_5$ (rule 2)

Ref. point $= \text{utt. time (rule 5)}$

**The sixth sentence describes an event, so again we have anterior present, with reference point equal to utterance time.**

Sentence Seven

Philipp trundled off slowly through the streets. (Past event)

**ANTERIOR PRESENT EVENT**

EVENT $e_7$: P. trundles off.

New variables: $t_7, e_7$

Narr. point $= t_6$ (rule 2)

Ref. point $= \text{utt. time (rule 5)}$

Sentence seven, eight, and nine, are similar to six, temporality-wise.

Sentence Eight

Philipp trundled slowly over the bridge. (Past event)

**ANTERIOR PRESENT EVENT**

EVENT $e_8$: P. trundles.

New variables: $t_8, e_8$

Narr. point $= t_7$ (rule 2)

Ref. point $= \text{utt. time (rule 5)}$

Sentence Nine

Philipp trundled slowly along the narrow ways. (Past event)

**ANTERIOR PRESENT EVENT**

EVENT $e_9$: P. trundles.

New variables: $t_9, e_9$

Narr. point $= t_8$ (rule 2)

Ref. point $= \text{utt. time (rule 5)}$

Sentence Ten

Philipp had taken a room at the hotel. (Past perfect event)

**ANTERIOR PRESENT (perfect) STATE**

STATE $s_{10}$: P. has taken a room.

New variables: $t_{10}, s_{10}$

Narr. point $= t_9$ (rule 2)

Ref. point $= \text{utt. time (rule 5)}$

The tenth sentence is also anterior present, so the reference point is utterance time.

Sentence Eleven

The hotel was in a shabby street. (Past state)

**SIMPLE PAST STATE**

STATE $s_{11}$: The hotel is [ . . . ]

New variables: $t_{11}, s_{11}$

Narr. point $= t_{10}$ (rule 2)

Ref. point $= t_{10}$ (rule 5)

Sentence eleven is past, so the reference point is determined ‘from the context’ (Kamp and Reyle, 1993, p. 610). In this case, there are ten times to choose
from, and $t_{10}$ seems best since the tenth sentence is also about the hotel.

### Sentence Twelve

The hotel was convenient for the school. (Past state)

**Simple Past State**

- **State $s_{12}$**: The hotel is [...]
- **New variables**: $t_{12}, s_{12}$
- **Narr. point** = $t_{11}$ (rule 2)
- **Ref. point** = $t_{10}$ (rule 5)

- $s_{12}$ includes $t_{12}$ (rule 3)
- $s_{12}$ includes $t_{11}$ (rule 4)
- $t_{12} = t_{10}$ (rule 6)

With respect to reference point, the same situation holds here as in the previous sentence, so we keep $t_{10}$.

### Sentence Thirteen

Philip was going to work at the school. (Past future' state)

**Posterior Past Event**

- **Event $e_{13}$**: P. is going to work.
- **New variables**: $t_{13}, e_{13}$
- **Narr. point** = $t_{12}$ (rule 2)
- **Ref. point** = $t_{10}$ (rule 5)

- $t_{13}$ includes $e_{13}$ (rule 3)
- $t_{12}$ precedes $e_{13}$ (rule 4)
- $t_{10}$ precedes $t_{13}$ (rule 6)

In the last sentence, the reference point appears to be the same as in sentence twelve, that is $t_{10}$, since they are both about the school.

### 4.4.5 Summary

The following table shows the results of processing the example using the interpretation algorithm. For the two phrases that were omitted, because their syntactic form is not discussed by Kämp and Reyle, no information was produced.

<table>
<thead>
<tr>
<th>Sentences</th>
<th>Variables</th>
<th>NP</th>
<th>RP</th>
<th>Temporal Formule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Philip ceased [...]</td>
<td>$s_1, t_1$</td>
<td>$t_1$</td>
<td>$t_1 \in s_1$</td>
<td></td>
</tr>
<tr>
<td>P. had settled down</td>
<td>$s_2, t_2$</td>
<td>$t_1$</td>
<td>$t_2 \in s_2, t_1 \in s_2, t_2 = t_1$</td>
<td></td>
</tr>
<tr>
<td>P. thought only [...]</td>
<td>$s_3, t_3$</td>
<td>UT</td>
<td>$t_3 \in s_3, t_2 \in s_3, t_3 &lt; \text{utt. time}$</td>
<td></td>
</tr>
<tr>
<td>P. had written.</td>
<td>$s_4, t_4$</td>
<td>UT</td>
<td>$t_4 \in s_4, t_3 \in s_4, t_4 &lt; \text{utt. time}$</td>
<td></td>
</tr>
<tr>
<td>Hayward had given.</td>
<td>$s_5, t_5$</td>
<td>$t_2$</td>
<td>$t_2 \in s_5, t_3 \in s_5, t_2 = t_3$</td>
<td></td>
</tr>
<tr>
<td>P. had an invitation.</td>
<td>$s_6, t_6$</td>
<td>UT</td>
<td>$t_0 \geq e_0, t_5 &lt; t_0 &lt; \text{utt. time}$</td>
<td></td>
</tr>
<tr>
<td>P. arrived in Paris.</td>
<td>$s_7, t_7$</td>
<td>UT</td>
<td>$t_7 \geq e_7, t_0 &lt; e_7, t_7 &lt; \text{utt. time}$</td>
<td></td>
</tr>
<tr>
<td>P. had [... put [...]</td>
<td>$s_8, t_8$</td>
<td>UT</td>
<td>$t_8 \geq e_8, t_7 &lt; e_8, t_8 &lt; \text{utt. time}$</td>
<td></td>
</tr>
<tr>
<td>P. trundled off [...]</td>
<td>$s_9, t_9$</td>
<td>UT</td>
<td>$t_9 \geq e_9, t_8 &lt; e_9, t_9 &lt; \text{utt. time}$</td>
<td></td>
</tr>
<tr>
<td>P. trundled [... bridge.</td>
<td>$s_{10}, t_{10}$</td>
<td>UT</td>
<td>$t_{10} \in s_{10}, t_0 \in s_{10}, t_{10} &lt; \text{utt. time}$</td>
<td></td>
</tr>
<tr>
<td>P. trundled [... ways.</td>
<td>$s_{11}, t_{11}$</td>
<td>$t_{10}$</td>
<td>$t_{11} \in s_{11}, t_{10} \in s_{11}, t_{11} = t_{10}$</td>
<td></td>
</tr>
<tr>
<td>P. had taken a room.</td>
<td>$s_{12}, t_{12}$</td>
<td>$t_{10}$</td>
<td>$t_{12} \in s_{12}, t_{11} \in s_{12}, t_{12} = t_{11}$</td>
<td></td>
</tr>
<tr>
<td>The hotel was [...]</td>
<td>$s_{13}, t_{13}$</td>
<td>$t_{10}$</td>
<td>$t_{13} \geq e_{13}, t_{12} &lt; e_{13}, t_{10} &lt; t_{13}$</td>
<td></td>
</tr>
</tbody>
</table>

With a permissive interpretation of the mathematics, the temporal relationships can be seen as in the picture below.
While for the most part, this is fairly consistent with the actual meaning of the text, there are some minor problems. As Dinsmore (1991, p. 210) points out, the reference point, in discourse representation theory, is simply a piece of the machinery, with no independent intuitive meaning, as it has in Reichenbach (1947) and the partitioned representation based approaches (section 4.6).

Another, and perhaps in practice more significant, problem is that for perfect aspect, only the time when the relevance of the result manifests itself — Reichenbach’s (1947) reference time — is identified; when the actual event took place is not. In our example, the writing of the letter and the giving of the introduction presumably occur on separate occasions before the ride in the carriage. But, in discourse representation theory, these times are not located, and all the three sentences have the same temporal position.

The temporal extent of states is also not considered by Kamp and Reyle (1993). In their defence can be said that this is usually not expressed linguistically, and world knowledge is needed for determining it. The invitation’s being in Philip’s pocket, for example, is likely to be temporary, while the hotel’s properties, for practical purposes, can be considered eternal.

### 4.5 Discourse grammar

A conceptually minimalistic approach to the modelling of temporal information in discourse is suggested by Hitzenm et al. (1995). In their model, the main component is a set of tables listing all possible pairwise combinations of situation types. Normally this means adjacent sentences, but they also divide the discourse into ‘threads’ (Kameyama, Passonneau and Poesio, 1993; Hitzenm et al., 1995, p. 256; Grover et al., 1994, pp. 4, 6, 12–17), so that a sentence can combine not only with the immediately previous one, but with the last one in another thread.

Any number of threads, in this model, can be available for extension with a new sentence, as long as the situation types of the new sentence and the last in the thread are listed in the appropriate table, or if the new sentence has a suitable time adjunct or rhetorical ‘cue phrase.’ It is not clear, however, when the creation of a new thread is forced. As long as the new sentence can be appended to the current thread, it will be, and the circumstances where that is not possible are not explained. Also not clear is to what extent adjuncts and
‘cue phrases’ have been implemented. Of the latter, only ‘because’ (Hitzeman
et al., 1995, pp. 254, 257, 259; Grover et al., 1994, pp. 3, 6, 8) and ‘as a result’
(Hitzeman et al., 1995, p. 258; Grover et al., 1994, pp. 6–7) are discussed,
and they are taken to encode a time relation directly, so that ‘because’ in the
following sentence (Hitzeman et al., 1995, p. 259; Grover et al., 1994, p. 8)
implies that the pushing preceded the falling.

John fell because Mary pushed him.

There are, in principle, three ways to integrate a new sentence $S$ into the rep-
resentation (Grover et al., 1994, p. 17), listed below together with the implied
temporal relationships. Each thread has a temporal focus (TF), which is the
latest non-state in the thread.

<table>
<thead>
<tr>
<th>Add $S$ to current thread.</th>
<th>Add $S$ to an old thread.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current TF precedes $S$.</td>
<td>Old TF precedes $S$.</td>
</tr>
<tr>
<td>S precedes current TF.</td>
<td>$S$ precedes current TF.</td>
</tr>
</tbody>
</table>

The situation types (see section 3.2) used by Hitzeman et al. are event, activity,
and state. Perfect and progressive aspect are discussed, as well as present, past,
and ‘future’ tense (Hitzeman et al., 1995, p. 257; Grover et al., 1994, p. 6). In
the tables, however (Hitzeman et al., 1995, p. 255; Grover et al., 1994, pp. 5,
41–47), only past and past perfect occurs. So, with three situation types, there
are six possible sentence types. The number of ways in which a new sentence,
$S_{new}$, can be added to the discourse representation is eight.

<table>
<thead>
<tr>
<th>PAST</th>
<th>EVENT</th>
<th>$S$ precedes $S_{new}$.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAST PERFECT</td>
<td>ACTIVITY</td>
<td>$S_{new}$ precedes $S$.</td>
</tr>
<tr>
<td>STATE</td>
<td>$TF$ precedes $S_{new}$.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S_{new}$ precedes $TF$.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S_{new}$ and $S$ overlap.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S_{new}$ and $TF$ overlap.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S_{new}$ and $S$ are part of same event.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$S_{new}$ and $TF$ are part of same event.</td>
<td></td>
</tr>
</tbody>
</table>

Since the temporal focus, $TF$, is the same as $S$ unless that is a state, the eight
temporal relations will, for four out of six sentence types, collapse to four. That
means the total number of entries in the tables (Hitzeman et al., 1995, p. 255;
Grover et al., 1994, pp. 5, 41–47), for the categories listed above, is 192.7 The
tables only state whether or not a given combination is legal, and the total
number of legal entries is between 96 and 100.8

If more than one insertion point, for a new sentence, is found acceptable by the
tables, then Hitzeman et al. suggest the computation of a ‘semantic distance’
(Hitzeman et al., 1995, p. 258; Grover et al., 1994, pp. 3, 4–5, 7, 14–16) to
determine which of them is most worthy. This is computed from the noun
phrases in each sentence via another table. In the code given in Grover et al.
(1994, pp. 14–15), this table has four entries.

7 $6 \times 2 + 8 + 6 \times 4 + 4 = 192$.
8 One of the tables is given in three different versions (Hitzeman et al., 1995, p. 255; Grover
et al., 1994, pp. 5, 42).
4.5.1 The example

To see how well our example (section 4.1) fares, in the hands of Hitzeman et al., we first need to establish the situation types. This is not discussed to any greater length in Hitzeman et al. (1995) or Grover et al. (1994), but, assuming no major deviations from the normal practice, the classification of the sentences will be as follows. Aspecual verbs, non-finite phrases, and modals are not discussed by Hitzeman et al., so sentences with those components are ignored.

Philip ceased to think of her. —
P. had settled down in his carriage. PAST PERFECT EVENT
P. thought only of the future. PAST ACTIVITY
P. had written to Mrs. Otter. PAST PERFECT EVENT
Hayward had given P. an introduction. PAST PERFECT EVENT
P. had an invitation to tea. PAST STATE
P. arrived in Paris. PAST EVENT
P. had his luggage put on a cab. —
P. trundled off slowly through the streets. PAST ACTIVITY
P. trundled slowly over the bridge. PAST ACTIVITY
P. trundled slowly along the narrow ways. PAST ACTIVITY
P. had taken a room at the hotel. PAST PERFECT EVENT
The hotel was in a shabby street. PAST STATE
The hotel was convenient for the school. PAST STATE
P. was going to work at the school. —

Only a fraction, 8 in fact, of the 36 possible sentence type pairs occur here, and they are listed below.

<table>
<thead>
<tr>
<th>PAST EVENT</th>
<th>PAST ACTIVITY</th>
<th>S2 just after S1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>S2 same event S1</td>
</tr>
<tr>
<td>PAST ACTIVITY</td>
<td>PAST ACTIVITY</td>
<td>S2 same event S1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S2 overlap S1</td>
</tr>
<tr>
<td>PAST ACTIVITY</td>
<td>PAST PERFECT EVENT</td>
<td>S2 proceed S1</td>
</tr>
<tr>
<td>PAST STATE</td>
<td>PAST EVENT</td>
<td>S2 just after TF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S2 proceed TF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S2 overlap S1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S2 same event TF</td>
</tr>
<tr>
<td>PAST STATE</td>
<td>PAST STATE</td>
<td>S2 just after TF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S2 same event S1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S2 same event TF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S2 overlap S1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S2 overlap TF</td>
</tr>
<tr>
<td>PAST PERFECT EVENT</td>
<td>PAST ACTIVITY</td>
<td>S2 just after S1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S2 same event S1</td>
</tr>
<tr>
<td>PAST PERFECT EVENT</td>
<td>PAST STATE</td>
<td>S2 just after S1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S2 same event S1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S2 overlap S1</td>
</tr>
<tr>
<td>PAST PERFECT EVENT</td>
<td>PAST PERFECT EVENT</td>
<td>S2 just after S1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S2 proceed S1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S2 same event S1</td>
</tr>
</tbody>
</table>

With the situation types of the example sentences and the possible sentence pairs, we can now proceed to interpret the example.
4.5.2 Details

Like in the previous section, the discourse starts with past perfect, and this does not seem to produce any adverse effects. As it turns out, only one thread is used for our example, which means that the only question, when adding a new sentence to the interpretation, is what temporal relationship it has to the previous one.

\[
\begin{align*}
P \text{ had settled down in his carriage.} & \quad \text{P. had written to Mrs. Otter.} \\
P \text{ thought only of the future.} & \quad \text{P. thought only of the future.} \\
P \text{ past perfect event } e_1 & \quad \text{past activity } a_2 \\
\text{Temporal focus: } e_1 & \quad \text{Temporal focus: } a_2
\end{align*}
\]

According to the table (Grover et al., 1994, p. 43), the possible relations between these two sentences are ‘just after’ and ‘same event.’ It is reasonably obvious that the former is what we seek, but less clear how Hitzeman et al.’s algorithm could find it.

\[
\begin{align*}
P \text{ thought only of the future.} & \quad \text{P. had written to Mrs. Otter.} \\
P \text{ past activity } a_2 & \quad \text{past perfect event } e_3 \\
\text{Temporal focus: } a_2 & \quad \text{Temporal focus: } e_3
\end{align*}
\]

For the second pair, only one relation is possible: ‘precedes’ (Grover et al., 1994, p. 45).

\[
\begin{align*}
P \text{ had written to Mrs. Otter.} & \quad \text{H. had given P. an introduction.} \\
P \text{ past perfect event } e_3 & \quad \text{past perfect event } e_4 \\
\text{Temporal focus: } e_3 & \quad \text{Temporal focus: } e_4
\end{align*}
\]

Now, the relevant table offers three possibilities (Grover et al., 1994, p. 45).

\[
\begin{align*}
\text{S2 just after S1} \\
\text{S2 precede S1} \\
\text{S2 same event S1}
\end{align*}
\]

The events referred to are the writing and the giving. We know that the latter enabled the former, so the relation is ‘precedes,’ but there is no way Hitzeman et al.’s tables could tell us that.

\[
\begin{align*}
\text{H. had given P. an introduction.} & \quad \text{P. had an invitation to tea.} \\
P \text{ past perfect event } e_4 & \quad \text{past state } s_5 \\
\text{Temporal focus: } e_4 & \quad \text{Temporal focus: } s_5
\end{align*}
\]

Here, also, there are three possible relations (Grover et al., 1994, p. 44), none of which fits very well.

\[
\begin{align*}
\text{S2 just after S1} \\
\text{S2 same event S1} \\
\text{S2 overlap S1}
\end{align*}
\]

The least painful alternative seems to be ‘just after,’ which is, however, clearly not what the text means.

\[
\begin{align*}
P \text{ had an invitation to tea.} & \quad \text{P. arrived in Paris.} \\
P \text{ past state } s_5 & \quad \text{past event } e_6 \\
\text{Temporal focus: } s_5 & \quad \text{Temporal focus: } e_6
\end{align*}
\]
Four this sentence pair, there are four possibilities (Grover et al., 1994, pp. 5, 42).

S2 just after TF  P. arrives just after H. gives him the introduction.
S2 precede TF   P. arrives before H. gives him the introduction.
S2 overlap S1  The arrival and the having of an invitation are simultaneous.
S2 same event TF The arrival and the giving of introduction are the same event.

It is clear that the 'overlap' is the right one, but not how that conclusion is arrived at. Semantic distance is no help, since arrival is not semantically closer to invitation than to introduction.

\[
\begin{array}{l|l}
P. arrived in Paris. & P. trundled off slowly through the streets. \\
PAST EVENT \epsilon_6 & PAST ACTIVITY \epsilon_7 \\
Temporal focus: \epsilon_6 & Temporal focus: \epsilon_7 \\
\end{array}
\]

Two possible relations: 'just after,' and 'same event' (Grover et al., 1994, p. 43). The first is the right one, but this cannot be determined by the tables.

\[
\begin{array}{l|l}
P. trundled off slowly [...] & P. trundled slowly over the bridge. \\
PAST ACTIVITY \epsilon_7 & PAST ACTIVITY \epsilon_8 \\
Temporal focus: \epsilon_7 & \\
\end{array}
\]

Here, the tables (Grover et al., 1994, p. 43) list two alternatives: 'same event,' and 'overlap.' Neither is exactly right, but the first seems less wrong.

\[
\begin{array}{l|l}
P. trundled slowly over the bridge. & P. trundled slowly along the narrow ways. \\
PAST ACTIVITY \epsilon_8 & PAST ACTIVITY \epsilon_9 \\
Temporal focus: \epsilon_8 & Temporal focus: \epsilon_9 \\
\end{array}
\]

Same situation as in previous pair. We use 'same event,' even though it is not right.

\[
\begin{array}{l|l}
P. trundled slowly along [...] & P. had taken a room at the hotel. \\
PAST ACTIVITY \epsilon_9 & PAST PERFECT EVENT \epsilon_{10} \\
Temporal focus: \epsilon_9 & Temporal focus: \epsilon_{10} \\
\end{array}
\]

The only possibility here, 'precede,' is correct (Grover et al., 1994, p. 45).

\[
\begin{array}{l|l}
P. had taken a room at the hotel. & The hotel was in a shabby street. \\
PAST PERFECT EVENT \epsilon_{10} & PAST STATE \epsilon_{11} \\
Temporal focus: \epsilon_{10} & \\
\end{array}
\]

None of the three alternatives here (Grover et al., 1994, p. 44) is very good, but 'overlap' seems preferable.

S2 just after S1
S2 same event S1
S2 overlap S1

As in most of the pairs, there is no apparent way for the algorithm to choose between the possibilities.

\[
\begin{array}{l|l}
The hotel was in a shabby street. & The hotel was convenient for the school. \\
PAST STATE \epsilon_{11} & PAST STATE \epsilon_{12} \\
Temporal focus: \epsilon_{10} & \\
\end{array}
\]
For the last pair, a total of five possible relations is offered by the tables (Grover et al., 1994, p. 44).

S2 just after TF Hotel is convenient just after P. takes the room.
S2 same event S1 Hotel’s being in street and being convenient are same event.
S2 same event TF Hotel’s being convenient and P.’s taking the room are same event.
S2 overlap S1 Hotel’s being in street and being convenient are simultaneous.
S2 overlap TF Hotel’s being convenient and P.’s taking the room are simultaneous.

It is not hard, for a human, to see that ‘S2 overlap S1’ is the right one, but, again, the tables alone cannot determine that.

4.5.3 Summary

The results, such as they are, are shown in the table below. Most of the relations are correct; the exceptions are indicated by arrows. It is noteworthy that none of the problems are with sentences next to the ‘gaps.’

| Philip ceased to think of her. | — | TF |
| P. had settled down in his carriage. | EVENT $e_1$ | $a_2$ just after $e_1$ |
| P. thought only of the future. | ACTIVITY $a_2$ | $e_3$ precedes $a_2$ |
| P. had written to Mrs. Otter. | EVENT $e_3$ | $e_4$ precedes $e_3$ |
| Hayward had given P. an introduction. | EVENT $e_4$ | $s_5$ just after $e_4$ ← $e_6$ overlap $s_5$ |
| P. had an invitation to tea. | STATE $s_5$ | $a_7$ just after $e_6$ |
| P. arrived in Paris. | EVENT $e_6$ | $a_8$ same event $a_7$ ← $a_9$ same event $a_8$ ← $e_{10}$ precedes $a_9$ |
| P. had his luggage put on a cab. | — | $s_{11}$ overlap $e_{10}$ |
| P. trundled off slowly through the streets. | ACTIVITY $a_7$ | $e_{10}$ overlap $s_{11}$ |
| P. trundled slowly over the bridge. | ACTIVITY $a_8$ | $s_{12}$ overlap $s_{11}$ |
| P. trundled slowly along the narrow ways. | ACTIVITY $a_9$ | |
| P. had taken a room at the hotel. | EVENT $e_{10}$ | |
| The hotel was in a shabby street. | STATE $s_{11}$ | |
| The hotel was convenient for the school. | STATE $s_{12}$ | |
| P. was going to work at the school. | — | |

All the sentences in the example end up in the same thread, which is probably not right. Kameyama et al. (1993, p. 2), Grover et al. (1994, p. 4), and Hitzerman et al. (1995, p. 256) agree that a shift to past perfect in a sequence of sentences in past tense introduces a new thread, so in the example there should be at least four different threads. But since there are one or more entries in the tables for each sentence pair in the example, any new threads are never created.

4.6 Partitioned representations

The first suggestion, in the computational literature, of partitions in meaning representation, is almost certainly Hendrix (1975). The idea was later developed independently by Wilks et al. (Bien, 1975; Wilks and Bien, 1979, 1983; Wilks, 1985; Ballim and Wilks, 1991), Shapiro et al. (Martins and Shapiro, 1988; Rapaport, Shapiro and Wiebe, 1986; Shapiro and Rapaport, 1987), Fauconnier
(1985), and Dinsmore (1991), and has been adopted by, among others, Cutrer (1995), and, in a limited form, Kamp and Reyle (1993).

Work on the interpretation of tense and aspect using partitioned representations has been done mainly by Dinsmore (1991) and Cutrer (1995). The latter is essentially an extension of the former’s work on tense, with a different emphasis. The core that they share consists of time spaces (section 4.6.3), and Reichenbach’s (1947; section 3.1) theory applied to these spaces rather than time points.

Logical formulae, in different partitions, cannot participate in the same derivation, nor can their potential or actual contradictions emerge, so the partitions can be used to store the beliefs of different agents, information about different realities, and also statements that hold at different times.

In the left example (Ballim and Wilks, 1991, p. 155), there are two belief spaces (‘system,’ and ‘John’), and two ‘topic environments’ (Ballim and Wilks, 1991, p. 46) which simply collect the information about particular topics, here John and his cat, respectively. Dinsmore’s example, to the right (Dinsmore, 1991, p. 131), contains four spaces, marked ‘sp_N,’ representing times and beliefs.

Fauconnier prefers circles to boxes. The diagram below encodes the sentence ‘In 1929, the lady with white hair was blonde.’ (Fauconnier, 1985, p. 29).

There are two spaces here (the circles), representing different times. The arrow indicates that the two entities x1 and x2 co-refer.

Among the theories discussed in this section, the far most formally comprehensive is Sowa’s ‘conceptual graphs,’ an example of whose visual incarnations are given below (Sowa, 2000, p. 269).

Like Fauconnier’s graph, this one has two spaces, here labelled ‘situations,’ representing the same individual at different times.

9 Such as fiction, future predictions and hopes, and counterfactual beliefs.
4.6.1 Space builders

The construction of partitioned representations during grammatical analysis of text has been studied mainly by Fauconnier (1985, 1997) and Dinsmore (1991). Linguistic items, in this approach, are divided into three classes: space builders, descriptions, and frames. The last category is a catch-all for those verbs that are not space builders, which is most of them.

Space builders (Fauconnier, 1985, pp. 16–18, 110–111; Fauconnier, 1997, pp. 37, 40; Dinsmore, 1991, pp. 119–121, 149–160, 200–204) are expressions that identify the space where new content should be added, which can mean locating an existing space or creating a new one. The nature of the connections between spaces also depends on the space builders, but these connection types, and the expressions themselves, are not defined exhaustively by either Fauconnier or Dinsmore. Although no complete list of space types is given, both Fauconnier and Dinsmore give several examples. Wills et al. discuss only propositional attitude verbs.

Belief

Max believes the moon is a cheese.
He thinks they will win.
In her opinion, the moon is a cheese.
She is worried they will be late.

(Hope)

Mary hopes they will win.
I wish they will win.
Hopefully, they will win.

Fictionality

In that movie, computers are useful.
In that painting, the sky is black.
In John’s mind, he is right.
I dreamed I was a space cadet.
In reality, they will lose the game.
They will actually lose the game.

Possibility

It is possible that they will win.
They might possibly win.
They will probably win.
Maybe they will win.
Perhaps they will win.
Either you win or you learn something.

Conditional

If you buy it then you will be happy.
Assuming you buy it, you will be happy.
Given that you bought it, you will be happy.
Hypothetically speaking, you will be happy.
Suppose it rains tomorrow. Then we won’t go.

Generalisation

Every man loves a woman.
Each man loves a woman.
In chess, the queen is strongest.
At sea, there is no room for mistakes.

Perspective

Viewed from above, the country is beautiful.
To his optimistic eyes, life looked like a green field.
At the factory, they are working.

Temporal

In 1929, phones were expensive.
She will do it.
Next time you are here, we should go to Dorsia.

Other than space builders, Dinsmore (1991) and Fauconnier (1985) discuss
mainly descriptions, which are typically noun phrases, and treat most verbs simply as atomic predicates.

4.6.2 Descriptions

Aristotle, in *On Sophistical Refutations* (Aristotle, 1978, 166a25), states that modal predicates, such as 'can,' apply in two different ways.

The following examples are connected with the combination of words, for instance, 'A man can walk when sitting and write when not writing.' The significance is not the same if one utters the words separately as it is if one combines them, namely, 'a man can walk-while-sitting,' and, similarly, in the other example, if one combines the words and says 'a man can write-when-not-writing,' for it means that he can write and not write at the same time; whereas if one does not combine the words it means that, when he is not writing, he has the power to write.

From this developed the Scholastic theory of the composite and divided senses of propositions, here explained by Thomas Aquinas (Bocheński, 1970, p. 185).

Further (it is objected), if everything is known by God as seen in the present, it will be necessary that what God knows, is, as it is necessary that Socrates sits given that he is seen to be sitting. But this is not necessary absolutely, or as is said by some, by necessity of the consequent; rather conditionally, or by necessity of consequence. For this conditional is necessary: if he is seen to be sitting, he sits. Whence also, if the conditional is turned into a categorical, so that it is said: what is seen to be sitting, necessarily sits, evidently if this is understood as *de dicto* and composite, it is true; but understood as *de re* and divided, it is false.

The two senses can be paraphrased as follows.

<table>
<thead>
<tr>
<th></th>
<th>Composite (<em>de dicto</em>)</th>
<th>Divided (<em>de re</em>)</th>
</tr>
</thead>
<tbody>
<tr>
<td>He who is seen to be sitting (now) must necessarily (always) sit.</td>
<td>The statement 'If he is seen to be sitting, he sits' is necessarily true.</td>
<td></td>
</tr>
</tbody>
</table>

In the first case, *de dicto*, the predicate 'necessarily' is applied to the proposition, while in the *de re* case it applies to an entity (the man).

**Substitution in opaque contexts**


*Lenin* died in 1924.

*Vladimir Ilyich Ulyanov* died in 1924.
But this cannot be done inside an opaque context, like ‘believe,’ since the believer is not necessarily aware of all possible descriptions.

John believes Lenin died in 1925.
† John believes Vladimir Ilyich Ulyanov died in 1925.

In the latter case, let us imagine, John does not have an opinion about the time of death of the person since he has never heard the name.

<table>
<thead>
<tr>
<th>Speaker’s belief space</th>
<th>Speaker’s view of John’s beliefs</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x = \text{Lenin}$</td>
<td>$y = \text{Lenin}$</td>
</tr>
<tr>
<td>$x = \text{Vladimir Ilyich Ulyanov}$</td>
<td>-</td>
</tr>
<tr>
<td>$x$ died in 1924</td>
<td>$y$ died in 1925</td>
</tr>
</tbody>
</table>

For the speaker, both names refer to the same individual. John knows only the name ‘Lenin.’ Note, however, that the second sentence above could be used when speaking to someone else who does know both names, to report John’s mistaken belief about the date.

In the normal terminology, John’s belief that ‘Lenin died in 1925’ is called de dicto, since he would use those words himself, and the belief ‘Vladimir Ilyich Ulyanov died in 1925,’ if we attribute it to John, is called de re. But this use of the terms is, to a degree, incorrect. Whether or not John’s belief is de dicto depends on the contents of his mind, not on the observers. If John, like many people, knows the basic facts about Lenin and, for some reason, believes he died in 1925, then his belief ‘Lenin died in 1925’ is really de re. It would be de dicto if he, for instance, knew nothing whatsoever about the person ‘Lenin’ but had heard or read the statement ‘Lenin died in 1925,’ and held it to be true, for whatever cause.

Attributive versus referential

Donnellan (1966) claimed that a description can be used in two different ways. In the referential sense, the content of the description is not important beyond serving to identify some specific entity. When used attributively, on the other hand, the description itself conveys information. A good example of the latter is when someone has just read a book and says:

*The author knows a lot about horses.*    \text{ATTRIBUTIVE}

Even though there is one particular person who wrote the book and the description identifies him or her, it is the authorship role rather than the individual that is important. The speaker might, for example, be mistaken about the author’s identity and still make a correct statement, only having the wrong subject in mind.

A referential description is meant only to enable the hearer to pick out the right referent. At a party, for instance, somebody could ask:

*Who is the girl in the blue dress?*  \text{REFERENTIAL}

And this question can be answered even if the dress is in fact violet.
As Donnellan (1966, p. 297) points out, this classification of descriptions depends on the speaker’s intention, and the same sentence can be used in either manner. Suppose, for example, that an event is held that involves authors and also horses, a country fair perhaps, and that somebody, assumed to be an author, is handling a horse with skill. Then the speaker can say:

The author knows a lot about horses. Referential

And be understood even if the horseman happens to be not the author after all. Similarly, if a performance, to be held later, is discussed, and the speaker knows that somebody will wear a blue dress but not who that person will be this time, he could ask:

‘Who is the girl in the blue dress?’ Attributive

If it so happens that a change of plans results in there being no blue-dressed girl in the play, then this question has no answer.

Fauconnier (1985, pp. 39–51, 56–63) claims that the attributive sense is primary, and that the referential meaning is computed in a separate step which he calls ‘value assignment.’ Dinsmore (1991, pp. 128, 165–167) discusses the distinction and gives representations for both sense, but he does not explain how they are arrived at in practice.

Referring functions

Nunberg (Nunberg, 1978, p. 52, Nunberg, 1979, p. 156) uses the term ‘referring function’ to cover any possible relation between a term used to identify something, and the referent. A special case of this is metonymy, whereby one aspect of an entity can stand for the whole or some other part.

The red skirt got off at Picadilly.
The Pentagon fixed the mind control device.

Presumably, not only the skirt, but the wearer too, got off, and the device development was discontinued by human decision makers housed in the Pentagon.

Fauconnier (1985, pp. 18–22) suggests an analysis where ‘referring functions’ hold between entities in different spaces, and are derived from the type of link between them. A picture or a photo, for example, has an associated space that is different from ordinary reality.

You look good in that photo.
In Blade Runner, Harrison Ford gets the girl.

Here, real world descriptions (you, Harrison Ford) refer to fictional or symbolical entities whose only material existence is on photographic film or paper.

In the last sentence, says Fauconnier, the sentence adverbial ‘in Blade Runner’ triggers the construction of a movie space with an associated referring function from real world actors to characters in the film. For the proper name Harrison Ford, this function will then provide a link to Deckard, who gets the girl.
Real world  Either  The movie
Harrison Ford  —  Deckard
—  the girl  Rachel

The description 'the girl' can presumably apply in either space, since it is equally true in both.

4.6.3 Time spaces

Spaces have also been used to represent temporal information, by, among others, Fauconnier (1985, 1997), Dinsmore (1991), and Cutrer (1995). With this method, each distinct time that is introduced, either explicitly by an adjunct, or indirectly by verb tense, corresponds to a time space. Since meaning expressions in different spaces do not interact, normal logical inference can be used. The difference between this approach and modifications of the inference machinery seems to have more to do with ease of use than metaphysics, and formally the two approaches might well be equivalent since predicate logic can be used to implement a partitioned representation, as in conceptual graphs (Sowa, 2000).

Instead of a reference point, Dinsmore and Cutrer have a 'focus space,' which is where the content of a sentence is generally added (Dinsmore, 1991, p. 122).

<table>
<thead>
<tr>
<th>Reichenbach</th>
<th>Dinsmore/Cutrer/Fauconnier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speech point</td>
<td>Base space (^1) ((space_b, time_b))</td>
</tr>
<tr>
<td>Event point</td>
<td>Event space ((space_e, time_e))</td>
</tr>
<tr>
<td>Reference point</td>
<td>Focus space ((space_r, time_r))</td>
</tr>
</tbody>
</table>

The mapping from point/space combinations to verb tense/aspect classes is also slightly different in Dinsmore (1991, pp. 209, 224), compared to Reichenbach's, which is shown with examples on page 28.

\(^{10}\) See the pictures on page 63.

\(^{11}\) Cutrer (1995) and Fauconnier (1997) use a fourth space, 'viewpoint,' which might be a closer counterpart to Reichenbach's speech point, but, for clarity, that is ignored and 'base,' here, subsumes it.

\(^{12}\) Dinsmore (1991) gives no examples of these categories, but the phrases here are the only plausible ones. The same verb constructions are given by Allen (1995, pp. 410-411) for these time point configurations (but he uses different terms, see page 74).
Besides perfect and ‘prospective,’ shown above, Dinsmore discusses ‘inceptive’ and ‘terminative,’ which in English are realised by aspetual verbs.

<table>
<thead>
<tr>
<th>Tense</th>
<th>Example</th>
<th>Space Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perfect</td>
<td>had built</td>
<td>time (_e) &lt; time (_r)</td>
</tr>
<tr>
<td>Prospective</td>
<td>was going to build</td>
<td>time (_e) &gt; time (_r)</td>
</tr>
<tr>
<td>Progressive</td>
<td>was building</td>
<td>time (_e) during time (_r)</td>
</tr>
<tr>
<td>Inceptive</td>
<td>began building</td>
<td>time (_r) begins time (_e)</td>
</tr>
<tr>
<td>Terminative</td>
<td>finished building</td>
<td>time (_e) finishes time (_r)</td>
</tr>
<tr>
<td>Perfective</td>
<td>built</td>
<td></td>
</tr>
</tbody>
</table>

Tense, aspect, and aspetual verbs, in this system, are space builders (section 4.6.1; Dinsmore, 1991, pp. 65–67, 123, 126, 159–160, 211–216, 221–225), so interpretation of a verb phrase includes finding focus and event time spaces, which means either locating existing ones, or creating new.

### 4.6.4 Verb groups

The approach used by Cutrer (1995, pp. 87–93) is more or less equivalent to Dinsmore’s but formalised in a different way. She uses the following primitive tense/aspect classes.\(^{13}\)

<table>
<thead>
<tr>
<th>Tense</th>
<th>Space Conditions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Past</td>
<td>Space (_r)</td>
</tr>
<tr>
<td>Present</td>
<td>Space (_r) (^{14}), time (_e) &lt; time (_r)</td>
</tr>
<tr>
<td>Future</td>
<td>Space (_r), time (_e) &gt; time (_r)</td>
</tr>
<tr>
<td>Perfect</td>
<td>Space (_e), time (_e) &lt; time (_r)</td>
</tr>
<tr>
<td>Progressive</td>
<td>Space (_e)</td>
</tr>
</tbody>
</table>

From which the meaning of the English verb group types are expressed as follows (Cuter, 1995, pp. 140–249).

<table>
<thead>
<tr>
<th>Traditional Present</th>
<th>Cutrer’s examples</th>
<th>Cutrer’s formalisation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Present progressive</td>
<td>&amp; writing</td>
<td>PRESENT + PROGRESSIVE</td>
</tr>
<tr>
<td>Past</td>
<td>wrote</td>
<td>PAST</td>
</tr>
<tr>
<td>‘Future’</td>
<td>will/is going to write</td>
<td>PRESENT + FUTURE</td>
</tr>
<tr>
<td>Present perfect</td>
<td>has written</td>
<td>PRESENT + PERFECT</td>
</tr>
<tr>
<td>Past perfect</td>
<td>had written</td>
<td>PAST + PERFECT</td>
</tr>
<tr>
<td>‘Future perfect’</td>
<td>will have written</td>
<td>PRESENT + FUTURE + PERFECT</td>
</tr>
<tr>
<td>‘Past future’</td>
<td>would write</td>
<td>PAST + FUTURE</td>
</tr>
<tr>
<td>‘Past future perfect’</td>
<td>would have written</td>
<td>PAST + FUTURE + PERFECT</td>
</tr>
</tbody>
</table>

There are several glaring omissions here, such as past progressive and the perfect progressives (see the table on page 12), but it is still the most complete list within the computational literature discussed in this chapter.

\(^{13}\)There are two more but they are used only for the French examples.

\(^{14}\)For present, says Cutrer (1995, pp. 89, 140), Space \(_r\) can be the same as Space \(_b\), instead of a child of it.
Dinsmore’s and Cutrer’s treatment of past perfect (in the ‘past of present perfect’ sense, see page 14) is compared in the table below, to show the similarities (Dinsmore, 1991, pp. 159–160, 215; Cutrer, 1995, pp. 228–234).

<table>
<thead>
<tr>
<th>Dinsmore (past perfect)</th>
<th>Cutrer (past + perfect)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( space_b )</td>
<td>( space_{h1} )</td>
</tr>
<tr>
<td>child space: ( space_r )</td>
<td>child space: ( space_r )</td>
</tr>
<tr>
<td>( time_r &lt; time_b )</td>
<td>( time_r &lt; time_{h1} )</td>
</tr>
<tr>
<td>( space_r )</td>
<td>( space_{i1} = space_{b2} )</td>
</tr>
<tr>
<td>child space: ( space_r )</td>
<td>child space: ( space_r )</td>
</tr>
<tr>
<td>( time_r &lt; time_{i1} )</td>
<td>( time_r &lt; time_{b2} )</td>
</tr>
<tr>
<td>[( space_r )] has occurred</td>
<td>( space_e )</td>
</tr>
<tr>
<td>( space_e ) Sentence content</td>
<td>( space_e ) Sentence content</td>
</tr>
</tbody>
</table>

Each primitive, in Cutrer’s theory, has its own base space, so past perfect has two, since it is composed of two primitives. But apart from the labels, the same spaces, with the same temporal relationships, result in both cases.

A major shortcoming of both Dinsmore (1991) and Cutrer (1995) is that neither consider situation types (section 3.2), which means, for instance, that the progressive aspect is cursorily handled. Both authors simply say that an event described by the progressive takes place within a time period indicated by the focus (or base) space (Dinsmore, 1991, pp. 212, 216; Cutrer, 1995, pp. 163–170).

### 4.6.5 The example

Verb group syntax is well covered by the partitioned representationalists. Cutrer (1995, pp. 140–249) has an impressive list of tense and aspect types, and (Dinsmore, 1991, pp. 207–221) discusses aspatial verbs, time adjuncts, and even non-finite verbs (Dinsmore, 1991, pp. 216–217). Given that, they can reasonably be said to handle all the sentences in our example (section 4.1).

But Philip ceased to think of her a moment after he had settled down in his carriage.

He thought only of the future.
P. had written to Mrs. Otter.
Hayward had given P. an introduction.
P. had in his pocket an invitation to tea on the following day.

When he arrived in Paris P. had his luggage put on a cab.
P. trundled off slowly through the streets.
P. trundled slowly over the bridge.
P. trundled slowly along the narrow ways.
P. had taken a room at the hotel.
The hotel was in a shabby street.
The hotel was convenient for the school.
P. was going to work at the school.

PAST + TERMINATIVE
PAST + PERFECT
PAST
PAST + PERFECT
PAST
PAST + PERFECT
PAST
PAST
PAST
PAST + FUTURE

Neither Dinsmore nor Cutrer specifies, in an algorithmic sense, how the interpretation is done, and the present analysis of the example represents a ‘best-case,’ where tense and aspect is transformed into suitable space configurations without elaboration of the mechanical details. Narrative progression (section 4.3) is
discussed briefly by Dinsmore (1991, pp. 126, 202–204), but, as we have seen, situation types are not.

\[ \text{space}_1 \quad [\text{space}_2] \text{ ceases to occur.} \quad \text{time}_1 \text{ is past & time}_1 \text{ ends time}_2 \]

\[ \text{space}_2 \quad \text{Philip thinks of her.} \quad \text{time}_1 \text{ is one moment after time}_3 \text{ (or time}_4\text{?)} \]

\[ \text{space}_3 \quad [\text{space}_4] \text{ has occurred.} \quad \text{time}_5 = \text{time}_1 \]

\[ \text{space}_4 \quad \text{P. settles down [...]} \quad \text{time}_6 = \text{time}_5 \]

\[ \text{space}_5 \quad \text{P. thinks of the future.} \quad \text{time}_5 = \text{time}_4 \]

\[ \text{space}_6 \quad [\text{space}_7] \text{ has occurred.} \quad \text{time}_8 = \text{time}_5 \]

\[ \text{space}_7 \quad \text{P. writes to Mrs. Otter.} \quad \text{time}_9 = \text{time}_8 \]

\[ \text{space}_8 \quad \text{Hayward gives P. [...]} \quad \text{time}_{10} = \text{time}_8 \]

\[ \text{— Tea on the following day.} \quad \text{time}_{11} \text{ is after time}_{1,5,6,8,10} \]

\[ \text{space}_{11} \quad \text{P. arrives in Paris.} \quad \text{time}_{12} = \text{time}_{11} \text{ (non-finite phrase)} \]

\[ \text{space}_{12} \quad \text{Luggage is put on cab.} \quad \text{time}_{13} = \text{time}_{12} \text{ & time}_{13} \text{ starts time}_{14} \]

\[ \text{space}_{13} \quad \text{[space}_{14}\text{] starts to occur.} \quad \text{time}_{14} = \text{time}_{13} \]

\[ \text{space}_{14} \quad \text{P. trundles through [...]} \quad \text{time}_{15} = \text{just after time}_{14} \]

\[ \text{space}_{15} \quad \text{P. trundles over [...]} \quad \text{time}_{16} \text{ is just after time}_{15} \]

\[ \text{space}_{16} \quad \text{P. trundles along [...]} \quad \text{time}_{17} \quad \text{?} \]

\[ \text{space}_{17} \quad [\text{space}_{18}] \text{ has occurred.} \quad \text{time}_{17} = \text{?} \]

\[ \text{space}_{18} \quad \text{P. takes a room.} \quad \text{time}_{18} = \text{?} \]

\[ \text{space}_{19} \quad \text{The hotel is in [...]} \quad \text{time}_{19} = \text{?} \]

\[ \text{space}_{20} \quad \text{The hotel is convenient.} \quad \text{time}_{20} = \text{?} \]

\[ \text{space}_{21} \quad [\text{space}_{22}] \text{ will occur.} \quad \text{time}_{21} = \text{?} \]

\[ \text{space}_{22} \quad \text{P. works at the school.} \quad \text{time}_{22} = \text{?} \]

Most of the spaces fit into a sequence, given in the first part of the table below. Realising that time\textsubscript{11} (arrival) comes after time\textsubscript{1,5,6,8,10} (carriage-travel), rather than being simultaneous with them, seems to require knowledge about travelling. Between space\textsubscript{3/4} (settling down) and time\textsubscript{1,5,6,8,10} is one moment, but the trip in the carriage, which ends at time\textsubscript{11}, presumably takes longer than that.

The arrival has been analysed as a distinct event rather than as the end of the carriage journey, since it is a simple verb and we are not worrying about situation types. That space\textsubscript{11} (arrival) and space\textsubscript{12} (luggage on cab) are the same seems to be what is explicitly said, but knowledge of travelling would suggest that they are in fact separate.

In the second half of the table below are times whose exact position cannot be determined. Those reported by perfect aspect (time\textsubscript{7}, time\textsubscript{9}, and time\textsubscript{18}) could, so far as we know here, have taken place a hundred years earlier, and time\textsubscript{22} (work), likewise, might lie far away into the future. The hotel properties, time\textsubscript{10} (in street) and time\textsubscript{20} (convenient), fits, following Dinsmore (1991) and Cutrer (1995), into the narrative at time\textsubscript{16,17}, but in reality they extend into the past and future beyond the limits of the text.
time_{3}/time_{4} ends at time_{1,5,6,8,10}
time_{2} one moment after time_{3}/time_{4}
time_{11,12,13} after time_{1,5,6,8,10}
time_{14} starts at time_{11,12,13}
time_{15} just after time_{14}
time_{16} just after time_{15}
time_{7} before time_{1,5,6,8,10}
time_{9} before time_{1,5,6,8,10}
time_{17} perhaps simultaneous with time_{16}
time_{18} before time_{17}
time_{19,20} perhaps simultaneous with time_{17}
time_{21} perhaps simultaneous with time_{19,20}
time_{22} after time_{21}

Assuming that all the spaces can be placed in time so that they correspond to the natural understanding of the text, the configuration shown below is arrived at.

\[
\begin{array}{|c|c|}
\hline
\text{Hayward gives P. an introduction.} & space_{9} \\
\text{P. writes to Mrs. Otter.} & space_{7} \\
\text{P. takes a room at the hotel.} & space_{18} \\
\text{Philip thinks of her.} & space_{2} \\
\text{P. settles down in his carriage.} & space_{4} \\
\text{[space_{4}] has occurred.} & space_{3} \\
\hline
\text{[space_{2}] ceases to occur.} & space_{1} \\
\text{P. thinks of the future.} & space_{5} \\
\text{[space_{7}] has occurred.} & space_{6} \\
\text{[space_{8}] has occurred.} & space_{8} \\
\text{P. has an invitation.} & space_{10} \\
\hline
\text{P. arrives in Paris.} & space_{11} \\
\text{Luggage is put on cab.} & space_{12} \\
\text{[space_{14}] starts to occur.} & space_{13} \\
\text{P. trundles through [...]} & space_{14} \\
\hline
\text{P. trundles over [...]} & space_{15} \\
\text{[space_{16}] has occurred.} & space_{17} \\
\text{The hotel is in [...]} & space_{19} \\
\text{The hotel is convenient.} & space_{20} \\
\text{[space_{22}] will occur.} & space_{21} \\
\text{P. works at the school.} & space_{22} \\
\hline
\end{array}
\]

This representation is close to the intuitive meaning of the discourse, except for the duration of states such as the hotel’s properties, but given that fundamental issues like situation types (section 3.2) and narrative progression (section 4.3) are not addressed, it seems unfair to compare Dinsmore’s (1991) and Cutler’s (1995) work directly with the other more computational theories.
4.7 Segments and stack

Of the computational theories of tense and aspect, the most comprehensive in
the literature is probably the algorithm described in Allen (1995, chap. 16),
which is part of a system for processing dialogue presented semi-formally in
that book. The algorithm maintains a stack of 'discourse segments,' into which
new utterances are incorporated, taking into account tense and aspect, situation
types, discourse relations, and 'cue phrases.'

The main use of discourse segments, in Allen's theory, is to serve as contexts
for logical inference and anaphora resolution, which means they cannot contain
temporal and modal statements. Below are listed the properties of a segment
(Allen, 1995, pp. 504–505), together with an example which shows the segments

- All sentences in the segment should have
  the same topic.
- Either all sentences refer to the same time,
  or there is a simple narrative progression.\(^1\)
- All sentences have the same modality.
- Anaphoric references link to entities within
  the segment, or, for the first sentence in the
  segment, to the outer segment (the one im-
  mediately below it on the stack).

As can be seen in the picture, the segments can be nested.

Pronoun interpretation, in Allen (1995, pp. 435–438) is done by the method
suggested in Grodz, Joshi and Weinstein (1983), which is also discussed together
with temporal interpretation (Allen, 1995, pp. 524–529), but, for clarity, that
has been omitted, here, from the presentation of Allen's algorithm (page 76).

4.7.1 Tense trees

discusses 'tense trees,' a method introduced by Hwang and Schubert (1992).
These trees have a root node representing present tense and branches for past,
perfect, and 'future,' and can encode complex tenses using multiple nodes. In
the original design, shown to the left below (Hwang and Schubert, 1992, p. 234),
there is also a dashed line corresponding to subordinate clauses, but Allen's
version to the right (1995, p. 522), does not include this.

The theory (Hwang and Schubert, 1992) is that interpretation of each sentence
starts at the root node, and every morphologic or syntactic time item, such
as tense suffixes and auxiliary verbs, indicate one of the three branches. After
descending the tree as indicated by these items, the node arrived at is the one

\(^1\)Since what is true at one time might not be true at another, and likewise for different
modalities.

\(^2\)Temporal order of described situations same as sentence order, and short time between
the situations (Allen, 1995, p. 505).
representing the tense of the sentence. Hwang and Schubert (1994) extends the method to also cover time adverbials.

An immediate problem with this approach (Hwang and Schubert, 1992, p. 238; Allen, 1995, p. 524) is that it cannot handle shifts in the reference point, as in the following example.

John had left for France.  
He took the ferry.

\[
\begin{array}{c|c|c}
\text{Times} & \text{Examples} \\
\hline
E & R - S & S \\
R,E & - & S \\
\end{array}
\]

Tense tree processing would here incorrectly conclude that the leaving and the taking of the ferry are temporally distinct. Another problem (Allen, 1995, p. 522–523) is when there are two separate past perfects in a sequence of simple past sentences. This occurs in the example from *Of Human Bondage* that we have been discussing (section 4.1). Both perfects will here be assigned to the same node in the tense tree, which would indicate that they are simultaneous, but in reality they need not be related at all.

### 4.7.2 Time points

In addition to tense trees, Allen (1995, pp. 410–411, 520–521) uses Reichenbach’s (1947) time point system, and gives the following version of the verb group classification (the original is on page 28).

<table>
<thead>
<tr>
<th>Reichenbach</th>
<th>Allen</th>
<th>(Times)</th>
<th>(Examples)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIMPLE PRESENT</td>
<td>SIMPLE PRESENT</td>
<td>S.R.E</td>
<td>sings</td>
</tr>
<tr>
<td>SIMPLE PAST</td>
<td>SIMPLE PAST</td>
<td>R.E - S</td>
<td>sang</td>
</tr>
<tr>
<td>SIMPLE FUTURE</td>
<td>SIMPLE FUTURE</td>
<td>S - R.E</td>
<td>will sing</td>
</tr>
<tr>
<td>ANTERIOR PRESENT</td>
<td>PRESENT PERFECT</td>
<td>E - S.R</td>
<td>has sung</td>
</tr>
<tr>
<td>ANTERIOR PAST</td>
<td>PAST PERFECT</td>
<td>E - R - S</td>
<td>had sung</td>
</tr>
<tr>
<td>ANTERIOR FUTURE</td>
<td>FUTURE PERFECT</td>
<td>S - E - R</td>
<td>will have sung</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S.E - R</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>E - S - R</td>
<td></td>
</tr>
<tr>
<td>POSTERIOR PRESENT</td>
<td>POSTERIOR PRESENT</td>
<td>S.R - E</td>
<td>is going to sing</td>
</tr>
<tr>
<td>POSTERIOR PAST</td>
<td>POSTERIOR PAST</td>
<td>R - E - S</td>
<td>was going to sing</td>
</tr>
<tr>
<td>POSTERIOR FUTURE</td>
<td>POSTERIOR FUTURE</td>
<td>S - R - E</td>
<td>will be going to sing</td>
</tr>
</tbody>
</table>

Allen’s (1995) algorithm requires that all sentences in a segment have the same tense and aspect, except that if the first one or more have perfect aspect, it can be dropped without ending the segment (Allen, 1995, pp. 524, 528). For this
reason, and since the identification of the Reichenbachian category of a verb phrase requires prior collation of all auxiliaries and tense morphemes, Allen (1995) does not really use tense trees as described by Hwang and Schubert (1992) at all, other than for illustration.

### 4.7.3 Situation types

The temporal interpretation, in Allen’s system, of sentences within a segment depends on their situation type. Allen (1995, pp. 407–409) mentions Vendler’s (1967; section 3.2) classification but his algorithm only distinguishes between states and events (Allen, 1995, pp. 517–520). The table below lists the relation assigned, for the different combinations of situation types.

<table>
<thead>
<tr>
<th>Event1, Event2</th>
<th>By default: Event1 precedes Event2.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>If Event2 is part of Event1 then Event2 is during Event1.</td>
</tr>
<tr>
<td></td>
<td>If Event2 enables Event1 then Event2 precedes Event1.</td>
</tr>
<tr>
<td>Event, State : Exp</td>
<td>By default: Event occurs during State : Exp.</td>
</tr>
<tr>
<td>or State : Exp, Event</td>
<td>If State : Exp enables Event then State : Exp precedes Event.</td>
</tr>
<tr>
<td></td>
<td>If Event causes State : Exp then Event precedes State : Exp.</td>
</tr>
<tr>
<td>State1, State2</td>
<td>State1 and State2 are simultaneous.</td>
</tr>
</tbody>
</table>

Between two sentences in different segments, i.e. the last sentence in one segment and the first in the next, the guiding principle is that the reference times are the same (Allen, 1995, p. 521).

### 4.7.4 The example

Allen (1995) does not discuss aspecual verbs or the temporal interpretation of non-finite verbs, so the classification of the sentences in our example (see section 4.1), will be as follows.

But Philip ceased to think of her  
  a moment after  
  he had settled down in his carriage.  
He thought only of the future.  
P had written to Mrs. Otter.  
Hayward had given P. an introduction.  
P had in his pocket an invitation  
to tea on the following day.  
When he arrived in Paris  
  he had his luggage  
  put on a cab.  
P trundled off slowly through the streets.  
P trundled slowly over the bridge.  
P trundled slowly along the narrow ways.  
P had taken a room at the hotel.  
The hotel was in a shabby street.  
The hotel was convenient for the school.  
P was going to work at the school.

As earlier, the text here starts with past perfect, and this does not seem to be a problem. In the following section is an account of the application of the algorithm to the example.
Discourse interpretation algorithm, temporal information (Allen, 1995, chap. 16)

The stack contains unfinished segments (Allen, 1995, pp. 506, 525), which have the components listed below. Roughly speaking, noun phrases refer to discourse entities (Allen, 1995, p. 431) and verb phrases to situations\(^{17}\)(Allen, 1995, p. 229).

| Tense/aspect category, for each sentence in the segment. |
| Last eventuality (LE) — situation described by last sentence. |
| All situations and (temporal and discourse) relations in the segment. |

For each new sentence S, choose one segment from the stack and one of the following operations (Allen, 1995, pp. 526–527, 528–529). If the segment is not at the top of the stack, pop all the ones above it (Allen, 1995, p. 508). Once a segment has been popped, it is closed and cannot be extended again.\(^{18}\) When a new segment is created, it is pushed on the stack.

- Extend segment
  - Anaphoric components in S must be resolved to entities in segment.
  - Tense of S must be same as segment, or same minus perfect aspect.
  - Situation described by S must be part of situation described by segment.
  - Update segment with structure from S.
  - Find temporal and discourse relations between S and rest of segment.
  - Set LE of segment to situation described by S.

- Create new segment: digression (from old segment)
  - Anaphoric components in S must be resolved to entities in old segment.
  - Digression must be explicitly marked, for example by a ‘cue phrase.’
  - Create new segment using information from S.

- Create new segment: temporal shift (from old segment)
  - Anaphoric components in S must be resolved to entities in old segment.
  - Either the tense/aspect of S is different\(^{19}\) from the old segment,
    or a different time is established by a time adjunct in S.
  - Create new segment by adding temporal constraints to old segment.

- Create new segment: elaboration (of old segment)
  - Anaphoric components in S must be resolved to entities in old segment.
  - S must describe part of situation marked as LE in old segment.
  - Create new segment about (old) LE.

If more than one operation is possible, the least number of pushes and pops of the stack is sought (Allen, 1995, p. 529).

\(^{17}\)Allen (1995, p. 512, 530) notes that situations and even segments are also discourse entities, since they can be referred to anaphorically, but we ignore that, for now.

\(^{18}\)But a new segment with the same topic can be created, of course.

\(^{19}\)Allen (1995, p. 523, 528) states that the new segment must have a tense below the old in the tense tree, but that seems too restrictive as it would exclude, for example, shifts from ‘future’ to present tense.
4.7.5 Details

For the first sentence, a new segment must be created.

1: Philip had settled down [...]  
Past perfect event Event$_a$
Create new segment Segment$_1$

<table>
<thead>
<tr>
<th>Segment$_1$ (Situation$_1$) Past perfect</th>
<th>Event$_a$ occurs during Situation$_1$</th>
</tr>
</thead>
</table>

It seems obvious that the next sentence should go in the same segment as the first. They are both in past tense, and the perfect aspect can be dropped without changing segment, but, as they stand here, it is not clear that the required connection in meaning (Allen, 1995, p. 528; see also page 76) exists. In the original text (section 4.1), however, the first sentence also mentions thinking.

2: He thought only [...]  
Simple past state State$_b$
Extend Segment$_1$

<table>
<thead>
<tr>
<th>Segment$_1$ (Situation$_1$) Simple past</th>
<th>Situation$_{a,b}$ occurs during Situation$_1$</th>
<th>Event$_b$ occurs during State$_b$</th>
</tr>
</thead>
</table>

According to the table on page 75, the default interpretation of an event and a state, not causally related, next to each other in the same segment, is that the event takes place during the state.

The third sentence is in the past perfect, which forces a shift to a new segment, since Segment$_1$ is now in the simple past. The new segment is pushed on top of the stack.

3: P. had written [...]  
Past perfect event Event$_c$
Temporal shift: Segment$_2$

<table>
<thead>
<tr>
<th>Segment$_2$ (Situation$_2$) Past perfect</th>
<th>Situation$_2$ precedes Situation$_1$</th>
<th>Event$_c$ occurs during Situation$_2$</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Segment$_1$ (Situation$_1$) Simple past</th>
<th>Situation$_{a,b}$ occurs during Situation$_1$</th>
<th>Event$_b$ occurs during State$_b$</th>
</tr>
</thead>
</table>

The Reichenbachian tense table on page 74 tells us, somehow, that Situation$_2$, referred to in past perfect, precedes Situation$_1$. The exact mechanics of this are not spelt out clearly in Allen (1995).

While the fourth sentence has the same tense and aspect as the third, and they are possibly even related — the giving of the invitation causing or enabling the writing of the letter — they should probably not go in the same segment, since there is no necessary temporal relation. This means a new segment (Segment$_3$) must be created to encode sentence four, but there are now two possibilities: either pop Segment$_2$ first, and make Segment$_3$ a ‘time shift’ from Segment$_1$, or keep Segment$_2$ and do the shift from it (Allen, 1995, pp. 513–514).

According to Allen (1995, p. 506, 528), pronouns must refer to entities in the same segment, or when they are in a sentence that triggers the push of a new segment, in the one immediately below it. There is a pronoun in sentence four, namely ‘whom,’ referring to Mrs. Otter.

20 Within the narrative, the same person was involved in both events so they must both have taken place within his lifespan.
21 Which may or may not be the one which was at the top of the stack before the new sentence, as any number of segments could be popped before pushing the new segment.
He had written to Mrs. Otter, the mistress to whom Hayward had given him an introduction.

If Segment2, which holds the representation of sentence three, is popped before adding sentence four, Mrs. Otter is no longer available for anaphoric reference. So after pushing the new segment, the stack has three items on it.

<table>
<thead>
<tr>
<th>4: Hayward had given [. . ]</th>
<th>Event4 ( \text{PAST PERFECT} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temporal shift: Segment3</td>
<td>( \text{Situation}_3 ) precedes ( \text{Situation}_1 )</td>
</tr>
<tr>
<td></td>
<td>( \text{Event}_4 ) occurs during ( \text{Situation}_3 )</td>
</tr>
<tr>
<td></td>
<td>Segment2 ( \text{PAST PERFECT} )</td>
</tr>
<tr>
<td></td>
<td>( \text{Situation}_2 ) precedes ( \text{Situation}_1 )</td>
</tr>
<tr>
<td></td>
<td>( \text{Event}_4 ) occurs during ( \text{Situation}_2 )</td>
</tr>
<tr>
<td></td>
<td>Segment1 ( \text{SIMPLE PAST} )</td>
</tr>
<tr>
<td></td>
<td>( \text{Situation}_{a,b} ) occurs during ( \text{Situation}_1 )</td>
</tr>
<tr>
<td></td>
<td>( \text{Event}_4 ) occurs during ( \text{State}_6 )</td>
</tr>
</tbody>
</table>

Sentence five is in the simple past, so it can be interpreted as extending Segment1. That the having in the pocket, and the sitting and thinking, are part of the same situation, as required by Allen’s algorithm (see page 76), might not be obvious to a computer, but this seems the most likely interpretation.

<table>
<thead>
<tr>
<th>5: P. had in his pocket [. . ]</th>
<th>Segment1 ( \text{SIMPLE PAST} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple past state: ( \text{State}_6 )</td>
<td>( \text{Situation}_{a,b,e} ) during ( \text{Situation}_1 )</td>
</tr>
<tr>
<td>Pop Segment3 and Segment2</td>
<td>( \text{Event}_4 ) occurs during ( \text{State}_6 )</td>
</tr>
<tr>
<td>Extend Segment1</td>
<td>( \text{State}_2 ) and ( \text{State}_6 ) are simultaneous</td>
</tr>
</tbody>
</table>

The previous occurrence in Segment1, Philip’s thinking, was a state \( \text{State}_6 \). So, following the table on page 75, they are simultaneous.

Sentences six to ten are all events in simple past, none of them causally related. This means that they are all added to Segment1, and that they follow each other in time. The first event \( \text{Event}_7 \) (sentence six) occurs during \( \text{State}_6 \), (see table on page 75) but nothing is said about how the state relates to the others \( \text{Event}_{g,h,i,j} \).

<table>
<thead>
<tr>
<th>6: P. arrived in Paris.</th>
<th>Segment1 ( \text{SIMPLE PAST} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple past event: ( \text{Event}_7 )</td>
<td>( \text{Situation}_{a,b,e,j} ) during ( \text{Situation}_1 )</td>
</tr>
<tr>
<td>Extend Segment1</td>
<td>( \text{Event}_4 ) occurs during ( \text{State}_6 )</td>
</tr>
<tr>
<td>( \text{State}_2 ) and ( \text{State}_6 ) are simultaneous</td>
<td></td>
</tr>
<tr>
<td>( \text{Event}_4 ) occurs during ( \text{State}_6 )</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>7: P. had his luggage [. . ]</th>
<th>Segment1 ( \text{SIMPLE PAST} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple past event: ( \text{Event}_8 )</td>
<td>( \text{Situation}_{a,b,e,j} ) during ( \text{Situation}_1 )</td>
</tr>
<tr>
<td>Extend Segment1</td>
<td>( \text{Event}_4 ) occurs during ( \text{State}_6 )</td>
</tr>
<tr>
<td>( \text{State}_2 ) and ( \text{State}_6 ) are simultaneous</td>
<td></td>
</tr>
<tr>
<td>( \text{Event}_4 ) occurs during ( \text{State}_6 )</td>
<td></td>
</tr>
<tr>
<td>( \text{Event}_4 ) precedes ( \text{Event}_9 )</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>8: P. trundled off [. . ]</th>
<th>Segment1 ( \text{SIMPLE PAST} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple past event: ( \text{Event}_9 )</td>
<td>( \text{Situation}_{a,b,e,j} ) during ( \text{Situation}_1 )</td>
</tr>
<tr>
<td>Extend Segment1</td>
<td>( \text{Event}_4 ) occurs during ( \text{State}_6 )</td>
</tr>
<tr>
<td>( \text{State}_2 ) and ( \text{State}_6 ) are simultaneous</td>
<td></td>
</tr>
<tr>
<td>( \text{Event}_4 ) occurs during ( \text{State}_6 )</td>
<td></td>
</tr>
<tr>
<td>Sequence: ( \text{Event}_4, \text{Event}_7, \text{Event}_9 )</td>
<td></td>
</tr>
</tbody>
</table>
9: P. trundled over [...]  
**SIMPLE PAST EVENT** *Event*$_{i}$  
**Extend** *Segment*$_{1}$  

| *Segment*$_{1}$ (Situation$_{1}$) **SIMPLE PAST**  
| Situation$_{a,b, e, f, g, h, i, j}$ during Situation$_{1}$  
| Event$_{o}$ occurs during State$_{o}$  
| State$_{o}$ and State$_{e}$ are simultaneous  
| Event$_{j}$ occurs during State$_{e}$  
| **Sequence:** Event$_{f, g, h, i, j}$  

10: P. trundled along [...]  
**SIMPLE PAST EVENT** *Event*$_{j}$  
**Extend** *Segment*$_{1}$  

| *Segment*$_{1}$ (Situation$_{1}$) **SIMPLE PAST**  
| Situation$_{a,b, e, f, g, h, i, j}$ during Situation$_{1}$  
| Event$_{o}$ occurs during State$_{o}$  
| State$_{o}$ and State$_{e}$ are simultaneous  
| Event$_{j}$ occurs during State$_{e}$  
| **Sequence:** Event$_{f, g, h, i, j}$  

After sentence ten there is a shift in temporal perspective.

11: P. had taken a room.  
**PAST PERFECT EVENT** *Event*$_{i}$  
**Temporal shift:** *Segment*$_{4}$  

| *Segment*$_{4}$ (Situation$_{4}$) **PAST PERFECT**  
| Situation$_{4}$ precedes Situation$_{1}$  
| Event$_{i}$ occurs during Situation$_{4}$  

| *Segment*$_{1}$ (Situation$_{1}$) **SIMPLE PAST**  
| Situation$_{a,b, e, f, g, h, i, j}$ during Situation$_{1}$  
| Event$_{o}$ occurs during State$_{o}$  
| State$_{o}$ and State$_{e}$ are simultaneous  
| Event$_{j}$ occurs during State$_{e}$  
| **Sequence:** Event$_{f, g, h, i, j}$  

The past perfect indicates that Situation$_{4}$, described by sentence eleven, precedes Situation$_{1}$ (see the tense table on page 74).

Sentence twelve has simple past but the subject matter is the same as in sentence eleven: the hotel. The likely interpretation, for a human reader, is that its physical location is independent of Philip’s residence, but there is no room in Allen’s (1995) system for this analysis. Therefore, we assign both sentence twelve and thirteen to *Segment*$_{4}$.

12: The hotel was in [...]  
**SIMPLE PAST STATE** State$_{e}$  
**Extend** *Segment*$_{4}$  

| *Segment*$_{4}$ (Situation$_{4}$) **SIMPLE PAST**  
| Situation$_{4}$ precedes Situation$_{1}$  
| Situation$_{1, i}$ occurs during Situation$_{4}$  
| Event$_{i}$ occurs during State$_{i}$  

| *Segment*$_{1}$ (Situation$_{1}$) **SIMPLE PAST**  
| Situation$_{a,b, e, f, g, h, i, j}$ during Situation$_{1}$  
| Event$_{o}$ occurs during State$_{o}$  
| State$_{o}$ and State$_{e}$ are simultaneous  
| Event$_{j}$ occurs during State$_{e}$  
| **Sequence:** Event$_{f, g, h, i, j}$
13: The hotel was convenient.

**Simple Past State:** State_m

Extend Segment_4

**Segment_4** (Situation_4) SIMPLE PAST

Situation_4 proceeds Situation_1

Situation_{i,j,m} during Situation_4

Event_i occurs during State_i

State_i and State_m are simultaneous

**Segment_1** (Situation_1) SIMPLE PAST

Situation_{a,b,e,f,g,h,i,j} during Situation_1

Event_e occurs during State_e

State_e and State_m are simultaneous

Event_j occurs during State_e

Sequence: Event_{f,g,h,i,j}

The situation types (see table on page 75) indicate that Event_4 (sentence eleven) occurs during State_12 (twelve) which is simultaneous with State_13 (thirteen).

Sentence fourteen contains a modal ‘future’ construction that Allen (1995) calls posterior (see page 74). This indicates a shift to a time later than Situation_1.

14: P. was going to work [...] 

**Posterior Past Event:** Event_{n}

Pop Segment_4

Temporal shift: Segment_5

**Segment_5** (Situation_5) POSTERIOR PAST

Situation_1 proceeds Situation_5

Event_n occurs during Situation_n

**Segment_1** (Situation_1) SIMPLE PAST

Situation_{a,b,e,f,g,h,i,j} during Situation_1

Event_e occurs during State_e

State_e and State_m are simultaneous

Event_j occurs during State_e

Sequence: Event_{f,g,h,i,j}

After the last sentence the stack contains two segments: Segment_5 and Segment_1.

When these two have been popped, there are five complete segments.

### 4.7.6 Summary

Five segments are created, all of them temporally related; there are no ‘digressions’ or ‘elaborations.’

<table>
<thead>
<tr>
<th>Sentences</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>had settled down</td>
<td>Event_n New segment Segment_1.</td>
</tr>
<tr>
<td>thought</td>
<td>State_{b} Extend Segment_1.</td>
</tr>
<tr>
<td>had written</td>
<td>Event_{b} Temporal shift: Segment_2.</td>
</tr>
<tr>
<td>had given</td>
<td>Event_d Temporal shift: Segment_3.</td>
</tr>
<tr>
<td>had (invitation) arrived</td>
<td>State_e Pop Segment_5 &amp; Segment_2, Extend Segment_1.</td>
</tr>
<tr>
<td>had (luggage put) trundled off</td>
<td>Event_f Extend Segment_1.</td>
</tr>
<tr>
<td>trundled</td>
<td>Event_h Extend Segment_1.</td>
</tr>
<tr>
<td>trundled</td>
<td>Event_i Extend Segment_1.</td>
</tr>
<tr>
<td>had taken</td>
<td>Event_k Temporal shift: Segment_4.</td>
</tr>
<tr>
<td>was (in street)</td>
<td>State_1 Extend Segments.</td>
</tr>
<tr>
<td>was (convenient)</td>
<td>State_{m} Extend Segment_4.</td>
</tr>
<tr>
<td>was going to work</td>
<td>Event_n Pop Segment_4, Temporal shift: Segment_5. (Pop Segment_5, Pop Segment_1.)</td>
</tr>
</tbody>
</table>

The temporal relations established by Allen’s algorithm between the different events, states, and segments are shown in the table below.
And in the following text, the segments are indicated by boxes, as in the picture on page 73, which is from Allen (1995, p. 507). The letter indexes here are the same as the ones used for states and events in the tables above.

<table>
<thead>
<tr>
<th>Segment</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment_1</td>
<td>Situation_1 occurred during Event_e, f, g, and h.</td>
</tr>
<tr>
<td>Tense/aspect: PAST PERFECT</td>
<td>Event_e, f, g, h occurs during State_e, f, g, and h.</td>
</tr>
<tr>
<td></td>
<td>In sequence: Event_g, h, f, e.</td>
</tr>
</tbody>
</table>

| Segment_2 | Situation_2 precedes Situation_1 |
| Tense/aspect: PAST PERFECT | Event_e occurs during Situation_2 |

| Segment_3 | Situation_3 precedes Situation_1 |
| Tense/aspect: PAST PERFECT | Event_e occurs during Situation_3 |

| Segment_4 | Situation_4 precedes Situation_1 |
| Tense/aspect: PAST PERFECT | Situation_4 occurs during Event_e. |
|           | Event_e occurs during State_e. |

| Segment_5 | Situation_5 follows Situation_1 |
| Tense/aspect: POSTERIOR PAST | Event_e occurs during Situation_5. |

There are obvious general similarities between Allen’s segments and the spaces discussed in section 4.6. Dinsmore (1991) and Cutrer (1995) are more formally committed, ideologically, to partitioned representations, but Allen (1995), whose justifications for segments are strictly practical, has worked out the technical details to a higher degree.

Other than the minor issues mentioned above, the only problem with the analysis above is that the duration of the states is not taken into account, or, indeed, known at all. In particular, the Hotel’s properties (State_m) are, according to the analysis here, associated with Philip’s room booking, while in reality they have much longer spans both forwards and backwards in time.
4.8 Summary

The theories reviewed in this chapter are alike in that they all have a considerable distance left before being able to account convincingly for more than a part of the temporal content of language. Tense and time adjuncts are the only phenomena they all handle, and not even progressive aspect is analysed with any particular consistency. The table below summarises the coverage.

<table>
<thead>
<tr>
<th></th>
<th>DRT\textsuperscript{22}</th>
<th>D. grammar</th>
<th>Part. repr.</th>
<th>Seg. &amp; stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Narr. progression</td>
<td>Narr. pt.</td>
<td>(Spaces\textsuperscript{23})</td>
<td>Relations</td>
<td></td>
</tr>
<tr>
<td>Present\text/h past tense</td>
<td>Reference point\textsuperscript{24}</td>
<td>Tables</td>
<td>Spaces</td>
<td>Segments</td>
</tr>
<tr>
<td>Modals\text/h future\textsuperscript{25}</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Perfect aspect\textsuperscript{25}</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Progressive aspect</td>
<td>Ignored\textsuperscript{26}</td>
<td>—</td>
<td>Yes</td>
<td>Mentioned\textsuperscript{27}</td>
</tr>
<tr>
<td>Non-finite verbs</td>
<td>—</td>
<td>—</td>
<td>Mentioned\textsuperscript{28}</td>
<td>—</td>
</tr>
<tr>
<td>Aspectual verbs</td>
<td>—</td>
<td>—</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>Typeshifting\textsuperscript{29}</td>
<td>—</td>
<td>Mentioned\textsuperscript{29}</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Situation types\textsuperscript{31}</td>
<td>ST./EV.</td>
<td>ST./ACT./EV.</td>
<td>—</td>
<td>ST./EV.</td>
</tr>
<tr>
<td>Discourse relations</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>Yes</td>
</tr>
<tr>
<td>Time adjuncts</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Nominalizations</td>
<td>—</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Partitions</td>
<td>—</td>
<td>(Threads)</td>
<td>Spaces</td>
<td>Segments</td>
</tr>
<tr>
<td>Type of theory</td>
<td>Logic</td>
<td>Code</td>
<td>Informal</td>
<td>Algorithm</td>
</tr>
</tbody>
</table>

The differences in the handling of perfect and progressive aspect seem arbitrary, and presumably any analysis of these, and also of non-finite verbs and nominalizations, can be integrated with each of the four theories.

An important problem which, among the theories reviewed in this chapter, is only discussed at any length by Cutrer (1995, chap. 4),\textsuperscript{32} is that tensed verb groups quite commonly express other temporal meaning than the default one, such as present tense used for future, and past tense for politeness (see section 2.2).

\textsuperscript{22}Discourse representation theory (Kamp and Reyle, 1993).

\textsuperscript{23}Narrative progression is only mentioned briefly by Dismore (1991, p. 126), and not at all by Cutrer (1995).

\textsuperscript{24}Perfect aspect, in discourse representation theory, is treated as a state, and the time of the event itself is not determined (see page 57).

\textsuperscript{25}See also section 6.5.

\textsuperscript{26}Treated the same as present, see page 50.

\textsuperscript{27}Allen (1995, p. 519).

\textsuperscript{28}Dismore (1991, pp. 216-217).

\textsuperscript{29}See section 3.3.

\textsuperscript{30}Grover et al. (1994, pp. 39-31).

\textsuperscript{31}In section 3.2 the following were discussed: state (ST.), stance, event (EV.), activity (ACT.), point, accomplishment, and achievement. Pustejovsky (1995a; section 3.3.2) also uses transition.

\textsuperscript{32}Dismore (1991, p. 203), Kamp and Reyle (1993, pp. 598-601), and Allen (1995, p. 409) also mention it.
The area, among those relevant for tense and aspect, where the theories discussed here are weakest is probably situation types (section 3.2). Most only make the basic distinction between states and events. Dinsmore (1991) and Cutrer (1995) ignore them completely. It is not surprising then, that aspectual expressions and type-shifting (section 3.3) are also barely mentioned.

Regarding partitioned representations there is a schism: Kamp and Reyle (1993) take a firm stand against using them to encode temporal information, while the others all seem to favour this, to varying degrees. Dinsmore (1991), Cutrer (1995), and Allen (1995) suggest essentially the same representations, and the ‘threads’ postulated by Hitzeman et al. (1995) hint vaguely in the same direction. The issue of partitions, and the question of which type of algorithm to use for computing the temporal interpretations, are the two areas where disagreement of any substance exists. Ignoring Hitzeman et al.’s non-deterministic contribution, the following table summarises the positions.

<table>
<thead>
<tr>
<th>Discourse representation theory</th>
<th>Temporal partitions</th>
<th>Interpretation algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Kamp and Reyle, 1993; sect. 4.4)</td>
<td>No</td>
<td>Time point calculus</td>
</tr>
<tr>
<td>(Hitzeman et al., 1995; sect. 4.5)</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Partitioned representations</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>(Cutrer, 1995; Dinsmore, 1991: 4.6)</td>
<td>Yes</td>
<td>—</td>
</tr>
<tr>
<td>Segments and stack</td>
<td>Yes</td>
<td>Segments and stack</td>
</tr>
<tr>
<td>(Allen, 1995; sect. 4.7)</td>
<td></td>
<td>(page 76)</td>
</tr>
</tbody>
</table>

Allen’s algorithm requires partitions (segments), at least as a temporary representation during processing, but nothing very obvious prevents the time point manipulation approach of Kamp and Reyle from producing partitioned representations, so to some extent the two issues are independent. On the other hand, the difference between using time partitions and not doing so might be less than it at first appears. Explicit symbols are used by both Dinsmore (1991)33 and Allen (1995)34 to identify the partitions, and these symbols can then be used to express, for example, temporal relationships between them. Conversely, from the time point symbols used by Kamp and Reyle (1993) those situations that take place at the same time can be collected, and then incarcerated in a partition, ready for further computation.

None of the four theories reviewed here mention the problem of syntactically linked but temporally unrelated situations, as in the sentence ‘The hotel was convenient for the school at which he was going to work.’

The hotel was convenient for the school (Extended state)
He was going to work at the school (Future point, seen from the past)

There might be an implicature that the convenience is related to the work, but no explicit and necessary temporal connection is expressed, as can be seen from: ‘The new hotel was convenient for the school at which he had once worked.’

33 E.g. ‘sp.1,’ and ‘sp.2,’ in the picture on page 63.
34 E.g. ‘SEG1,’ ‘SEG2,’ and ‘SEG3’ in the picture on page 73.
An overview of the components of tense and aspect interpretation can be found in the picture on page 3, reproduced here.

And the following table compares the two algorithms with the system presented in this thesis, using the terms from the picture above.

<table>
<thead>
<tr>
<th>Sit. type</th>
<th>Kamp and Reyle Section 4.4 Algorithm: p. 51</th>
<th>Allen Section 4.7 Algorithm: p. 76</th>
<th>Chapters 5–7 Code in the Appendix</th>
</tr>
</thead>
<tbody>
<tr>
<td>ontology</td>
<td>STATE/EVENT</td>
<td>STATE/EVENT</td>
<td>state/activity/point/transition(*X,X)</td>
</tr>
<tr>
<td>Verbs</td>
<td>STATE/EVENT</td>
<td>STATE/EVENT</td>
<td>‘Concepts’ (section 6.4)</td>
</tr>
<tr>
<td>Asp. verbs</td>
<td>--</td>
<td>--</td>
<td>Lexical operators (section 6.4)</td>
</tr>
<tr>
<td>V. particles</td>
<td>--</td>
<td>--</td>
<td></td>
</tr>
<tr>
<td>Type-shifting</td>
<td>--</td>
<td>--</td>
<td></td>
</tr>
<tr>
<td>Verb group</td>
<td>Table (page 50)</td>
<td>Table (page 74) / ‘Tense trees,’ 4.7.1</td>
<td>Table (page 106) with separate entries for ‘internal’ meaning</td>
</tr>
<tr>
<td>meaning</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internal time</td>
<td>--</td>
<td>--</td>
<td></td>
</tr>
<tr>
<td>External time</td>
<td>Tables (pages 28 &amp; 29)</td>
<td>Table (page 74)</td>
<td>Two simple rules (page 108)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Imperfective</td>
<td>--</td>
<td>--</td>
<td>Internal meaning</td>
</tr>
<tr>
<td>Perfect</td>
<td>Ref. point only</td>
<td>Event point only</td>
<td>Ref. point &amp; Event point</td>
</tr>
<tr>
<td>Segments</td>
<td>--</td>
<td>--</td>
<td>Interpreted individually</td>
</tr>
</tbody>
</table>

As can be seen in this table, both Kamp and Reyle (1993) and Allen (1995) only really handle the ‘external’ part of the meaning of tense and aspect, which is tense plus perfect aspect, more or less.

35Situation type instances (section 6.4).
36Lexical operators (section 6.4).
37See the table on page 95.
38These are implemented by the same operators, see sections 6.2 and 6.3.
39See also the table on page 107.
40Discussed in section 6.5.
41Temporarily unconnected parts of the same sentence, not Allen’s (1995) partitions.
42See page 111 and appendix A.5.
The next three chapters contain the solutions proposed here to the problems
of interpreting temporal information in discourse, starting with the syntactic
analysis of verb groups in the next chapter. Chapter six describes the semantics,
including situation types, lexical semantics, and internal time. External time is
discussed in chapter seven, together with the handling of multiple propositions.
The descriptions of the various areas discussed on the previous page have the
following distribution:

<table>
<thead>
<tr>
<th>Area</th>
<th>Section/Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lexical semantics</td>
<td>Section 6.4</td>
</tr>
<tr>
<td>Verb group meaning (Nominalizations)</td>
<td>Section 6.2; table on page 106</td>
</tr>
<tr>
<td>Situation type ontology</td>
<td>Section 6.1; table on page 95</td>
</tr>
<tr>
<td>Type shifting</td>
<td>Page 150</td>
</tr>
<tr>
<td>Internal time</td>
<td>Section 6.3</td>
</tr>
<tr>
<td>External time (time points)</td>
<td>Section 7.1; table on page 107</td>
</tr>
<tr>
<td>Segments</td>
<td>Page 111</td>
</tr>
</tbody>
</table>

The analysis of the example (see sections 1.1 and 4.1) produced by the sys-
tem is shown in section 7.4, and a technical description of the implementation,
including source code, can be found in the Appendix.
5 Formal syntax of verb groups

One of the more comprehensive implemented grammars for English verb groups in the computational linguistics literature is the one found in Norvig (1992), which is an extension of the one in Pereira and Shieber (1987), which is based in turn on the analysis in Gazdar, Pullum and Sag (1982). A fundamental assumption made by them all is that each auxiliary verb corresponds to one node in a parse tree, so that a complex verb group has a recursive structure, as in the picture below (Gazdar et al., 1982, p. 621).

The legal ways to combine nodes, in this tree, is specified by a table (Gazdar et al., 1982, p. 599), where each row contains the auxiliary verb and the type of the embedded (right-hand) node, together with the type of the combination.

<table>
<thead>
<tr>
<th>Left child</th>
<th>Right child</th>
<th>Node</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>can/may/must/will</td>
<td>Bare infinitive</td>
<td>Finite phrase</td>
<td>can be tall</td>
</tr>
<tr>
<td>do</td>
<td>Bare infinitive (not auxiliary)</td>
<td>Finite phrase</td>
<td>do leave</td>
</tr>
<tr>
<td>have</td>
<td>Past participle</td>
<td>Perfect</td>
<td>have gone away</td>
</tr>
<tr>
<td>be</td>
<td>Present participle</td>
<td>Progressive</td>
<td>be going away</td>
</tr>
<tr>
<td>to</td>
<td>Bare infinitive</td>
<td>Infinitive</td>
<td>to be tall</td>
</tr>
<tr>
<td>is</td>
<td>Infinitive</td>
<td>Finite, copula</td>
<td>is to go away</td>
</tr>
<tr>
<td>ought</td>
<td>Infinitive</td>
<td>Finite</td>
<td>ought to go away</td>
</tr>
<tr>
<td>be</td>
<td>Predictional phrase</td>
<td>Copula</td>
<td>be easy to please</td>
</tr>
</tbody>
</table>

Since the embedded phrases (the 'right child' column above) can have types, such as 'passive,' that are not lexical, the interpretation must proceed from the inside out, and the complements must be connected at the bottom of the verb phrase tree. In the approach suggested by Pereira and Shieber (1987), the properties used in the table are limited to the lexical categories for verbs (see section 2.1.1). An auxiliary verb, in this system, is labelled with the type of phrase it combines with (Requires), and the resulting type of the whole combination (Form). The code is listed on page 87 (Pereira and Shieber, 1987, pp. 115–116; Norvig, 1992, p. 735).

These are listed on page 12.
vp(\text{Form}) \rightarrow \text{iv}(\text{Form}).
vp(\text{Form}) \rightarrow \text{tv}(\text{Form}), \text{np}.

vp(\text{Form}) \rightarrow \text{aux}(\text{Require,Form}), \text{vp}(\text{Require}).

\text{iv}(\text{Form}) \rightarrow \{ \text{iv}(V,\text{Form}) \}.
\text{tv}(\text{Form}) \rightarrow \{ \text{tv}(V,\text{Form}) \}.

\text{aux}(\text{Require,Form}) \rightarrow \{ \text{aux}(\text{Aux,Require,Form}) \}.

\text{aux}( \text{have}, \quad \text{past_participle}, \quad \text{nonfinite} ).
\text{aux}( \text{have}, \quad \text{past_participle}, \quad \text{finite} ).
\text{aux}( \text{has}, \quad \text{past_participle}, \quad \text{finite} ).
\text{aux}( \text{had}, \quad \text{past_participle}, \quad \text{finite} ).
\text{aux}( \text{having}, \quad \text{past_participle}, \quad \text{present_participle} ).
\text{aux}( \text{do}, \quad \text{nonfinite}, \quad \text{finite} ).
\text{aux}( \text{does}, \quad \text{nonfinite}, \quad \text{finite} ).
\text{aux}( \text{did}, \quad \text{nonfinite}, \quad \text{finite} ).
\text{aux}( \text{to}, \quad \text{nonfinite}, \quad \text{nonfinite} ).
\text{aux}( \text{be}, \quad \text{present_participle}, \quad \text{nonfinite} ).
\text{aux}( \text{been}, \quad \text{present_participle}, \quad \text{past_participle} ).
\text{aux}( \text{being}, \quad \text{past_participle}, \quad \text{present_participle} ).
\text{aux}( \text{am}, \quad \text{present_participle}, \quad \text{finite} ).
\text{aux}( \text{is}, \quad \text{present_participle}, \quad \text{finite} ).
\text{aux}( \text{are}, \quad \text{present_participle}, \quad \text{finite} ).
\text{aux}( \text{were}, \quad \text{present_participle}, \quad \text{finite} ).
\text{aux}( \text{was}, \quad \text{present_participle}, \quad \text{finite} ).
\text{aux}( \text{Modal verbs}, \quad \text{nonfinite}, \quad \text{finite} ).

[...]

\text{iv}( \text{halts}, \quad \text{finite} ).
\text{iv}( \text{halt}, \quad \text{nonfinite} ).
\text{iv}( \text{halting}, \quad \text{present_participle} ).
\text{iv}( \text{halted}, \quad \text{past_participle} ).

[...]

\text{tv}( \text{writes}, \quad \text{finite} ).
\text{tv}( \text{write}, \quad \text{nonfinite} ).
\text{tv}( \text{writing}, \quad \text{present_participle} ).
\text{tv}( \text{written}, \quad \text{past_participle} ).

[...]
A small problem with this parsing code is that, as can be seen in the table on page 12, the past participle of 'be,' 'been,' can, in passives, combine with another past participle. This is not handled by the code in Pereira and Shieber (1987) or Norvig (1992), and while it can be easily added, it is also necessary to limit the recursion somehow, since more than one 'been' should not be allowed.

been examined
* been been examined
* been been been examined
* [...]

There are other related problems due to the fact that 'be' occurs both in passive and progressive forms. Only 'being written' (passive), and not 'being writing,'\(^2\) (progressive) is recognised, for example. Another problem with the code on page 87, as it stands, is that the meaning of the auxiliaries is not recorded properly. The mood is returned, but the tense, aspect, and voice of the phrase are not.

The description of verb group syntax in Quirk et al. (1985, pp. 151-154; sect. 2.1.6) can be straightforwardly implemented, producing a clear and compact solution that handles all the forms listed on page 12. Using this approach, the verb group is treated as one unit, and verb complements and modifiers are analysed separately, instead of, as is usually the case, being treated as lower parts of a tree headed by the main verb.

In the system developed here, this method is used. The full parser code is listed on page 91,\(^3\) and a simplified version, without the parameters for agreement information, sub-categorisation, and semantics, is listed on page 90.

To allow for inversion (section 2.1.4), there are two forms of the top-level procedure: vp/9 takes a syntactic operator\(^4\) as parameter, while vp/8 reads the whole verb group from the token list, as in the following example.

\[
\text{?- vp(Inf1,\ldots,\ldots,\text{[will,have,been,examined]},[]).}
\]

\[
\text{Inf1 = finite([\text{modal(will)},\text{perfect},\text{passive}]) ?}
\]

When there is an operator, like 'will' in the phrase 'will certainly not have been examined,' the parser uses operator/3 to find it, then recognises the adverbials using other code,\(^5\) and finally calls vp/9, with the operator as first parameter, to analyse the rest of the verb group.

\[
\text{?- vp(will,Inf1,\ldots,\ldots,\ldots,\text{[have,been,examined]},[]).}
\]

\[
\text{Inf1 = finite([\text{modal(will)},\text{perfect},\text{passive}]) ?}
\]

The five anonymous parameters in these examples are for person (Pers), number (Num), verb phrase symbol (V), sub-categorisation information (SubCat), and the semantic representation, or logical form (Sem).

\(^2\)This construction is uncommon but not impossible (Quirk et al., 1985, p. 154n).

\(^3\)And in appendix A.1.

\(^4\)Which is either an auxiliary, or 'do,' see section 2.1.4.

\(^5\)Note that this is not fully supported in the grammar in appendix A.1.
According to Quirk et al., a verb group has four components, **modal-perfect-progressive-passive**, that each consists of two words, always come in this sequence, and overlap pairwise, if they occur. The example phrase ‘may have been examined’ (Quirk et al., 1985, p. 152) has the following structure.

- **Modal**: may
- **Auxiliary**: have
- **Infinitive**: been
- **Past participle**: examined

The **modal** unit is composed of an auxiliary followed by an infinitive, and the **perfect** unit of ‘have’, in any form, followed by a past participle. The word ‘have’ in the example is in the infinitive form and participates in both the modal and the perfect components. ‘Been’, likewise, is part of both the perfect and the passive units.

Each of the four components is implemented by one Prolog DCG rule (vp1–vp4 on page 90/91). Since any of them can be omitted, these rules have two cases each, and the second case simply calls the next rule in the sequence. The table below shows, in simplified form, the first cases of the three rules where these, and not the second case, is active in the example, together with the word types the components consist of, as in the other table above. Since the **progressive** unit is empty in this example, the second case for that rule (vp3) is active.

<table>
<thead>
<tr>
<th></th>
<th>Modal</th>
<th>Perfect</th>
<th>Passive</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>May</strong></td>
<td>Modal auxiliary</td>
<td>Auxiliary have</td>
<td>Auxiliary be</td>
</tr>
<tr>
<td><strong>Have</strong></td>
<td>Infinitive (base)</td>
<td>PAST PARTICIPLE</td>
<td>PAST PARTICIPLE</td>
</tr>
<tr>
<td><strong>Been</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Examined</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The first rule, vp1, takes the modal (**may**) as an argument (**W1**) and checks that it has the right form, using word/2 which is where the lexicon is stored. It then gets the next word (**have** from the token list (**W2**)) and checks that it is an infinitive. Finally it calls the second rule, vp2, giving it word number two.

All the rules take one word as parameter and gets one from the token list. For those components that are empty, such as progressive/vp3 in this example, the second case is activated and simply passes the input word on to the next rule in the sequence.

\[\text{vp3}(W) \rightarrow \text{vp4}(W)\]

Finally the rule vp5 gets one word, which should be the main verb of the phrase.

\[\text{vp5}(W) \rightarrow \{\text{word}(W, \verb(main(), \ldots))\}\]

The complete parser, in the simplified form used above, is shown on page 90, and the full version on page 91, and in appendix A.1. All possible forms of English verb groups are handled, with the output for each type is listed on page 92, using the same examples as in the table on page 12.
vp --
    [w].
    wp0(w).
vp(0p) --
    wp0(w).

operator(w) --
    [w].
    \{ word(w,verb(Tp,_._._._)). (Tp = modal(_; Tp = prim(_))). \}

vp0(w1) --
    \{ word(w1,verb(_.Form,_._._._)). (Form = present ; Form = past) \}.
    wp1(w1).
vp0(w1) --
    \{ word(w1,verb(_.base,_._._._)) \}.
    wp1(w1).
vp0(w1) --
    \{ word(w1,verb(_.pres_part,_._._._)) \}.
    wp1(w1).

wp1(w1) --
    \{ word(w1,verb(modal(_;_._._._)) \}.
    [w2],
    \{ word(w2,verb(_.base,_._._._)) \}.
    wp2(w2).
wp1(w) --
    wp2(w).

wp2(w2) --
    \{ word(w2,verb(prim(have),_._._._._._.)) \}.
    [w3],
    \{ word(w3,verb(_.past_part,_._._._._._._._.)) \}.
    wp3(w3).
wp2(w) --
    wp3(w).

wp3(w3) --
    \{ word(w3,verb(prim(be),_._._._._._._._.)) \}.
    [w4],
    \{ word(w4,verb(_.pres_part,_._._._._._._._.)) \}.
    wp4(w4).
wp3(w) --
    wp4(w).

wp4(w4) --
    \{ word(w4,verb(prim(be),_._._._._._._._.)) \}.
    [w5],
    \{ word(w5,verb(_.past_part,_._._._._._._._.)) \}.
    wp5(w5).
wp4(w) --
    wp5(w).

wp5(w) --
    \{ word(w,verb(main(_),_._._._._._._._.)) \}.
vp (Infl_Pers_Num, V, SubCat, [sit(V, Infl, VerbSem) | Sem]) --> 
   [W].
vp0 (W, MainVerb, Infl_Pers_Num).
   { verbsem(MainVerb, V, SubCat, VerbSem, Sem) }.
vp (Op, Infl_Pers_Num, V, SubCat, [sit(V, Infl, VerbSem) | Sem]) --> 
   vp0 (Op, MainVerb, Infl_Pers_Num).
   { verbsem(MainVerb, V, SubCat, VerbSem, Sem) }.

operator (W) -->
   [W].
   { word(W, verb(Type, ...)). (Type = modal(_); Type = prim(_)) }.

vp0 (W1, MainVerb, finite(Type), Pers, Num) -->
   { word(W1, verb(_, Form, Pers, Num)). (Form = present; Form = past) }.
   vp1 (W1, [Form], MainVerb, Type).
vp0 (W1, MainVerb, infinitive(Type), Pers, Num) -->
   { word(W1, verb(_, base, Pers, Num)) }.
   vp1 (W1, [], MainVerb, Type).
vp0 (W1, MainVerb, participle(Type), Pers, Num) -->
   { word(W1, verb(_, pres_part, Pers, Num)) }.
   vp1 (W1, [], MainVerb, Type).

vp1 (W1, MainVerb, Type) -->
   { word(W1, verb(modal(Modal), ...)) }.
   [W2], { word(W2, verb(_, base, ...)) }.
   vp2 (W2, [modal], MainVerb, Type).
vp1 (W, Form, MainVerb, Type) -->
   vp2 (W, Form, MainVerb, Type).

vp2 (W2, Form, MainVerb, Type) -->
   { word(W2, verb(pron(have), ...)) }.
   [W3], { word(W3, verb(_, past_part, ...)) }.
   vp3 (W3, [perfect|Form], MainVerb, Type).
vp2 (W, Form, MainVerb, Type) -->
   vp3 (W, Form, MainVerb, Type).

vp3 (W3, Form, MainVerb, Type) -->
   { word(W3, verb(pron(be), ...)) }.
   [W4], { word(W4, verb(_, pres_part, ...)) }.
   vp4 (W4, [progressive|Form], MainVerb, Type).
vp3 (W, Form, MainVerb, Type) -->
   vp4 (W, Form, MainVerb, Type).

vp4 (W4, Form, MainVerb, Type) -->
   { word(W4, verb(pron(he), ...)) }.
   [W5], { word(W5, verb(_, past_part, ...)) }.
   vp5 (W5, [passive|Form], MainVerb, Type).
vp4 (W, Form, MainVerb, Type) -->
   vp5 (W, Form, MainVerb, Type).

vp5 (W, Form, MainVerb, RevForm) -->
   { main_verb(W, MainVerb).
       reverse(Form, RevForm) }.
<table>
<thead>
<tr>
<th>Verb group</th>
<th>Mood/tense/aspect/voice (Inf2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>examine</td>
<td>finite([present])</td>
</tr>
<tr>
<td>examines</td>
<td>finite([present])</td>
</tr>
<tr>
<td>examined</td>
<td>finite([past])</td>
</tr>
<tr>
<td>will examine</td>
<td>finite([modal(will)])</td>
</tr>
<tr>
<td>would examine</td>
<td>finite([modal(would)])</td>
</tr>
<tr>
<td>have examined</td>
<td>finite([present, perfect])</td>
</tr>
<tr>
<td>has examined</td>
<td>finite([present, perfect])</td>
</tr>
<tr>
<td>am examining</td>
<td>finite([present, progressive])</td>
</tr>
<tr>
<td>are examining</td>
<td>finite([present, progressive])</td>
</tr>
<tr>
<td>is examining</td>
<td>finite([present, progressive])</td>
</tr>
<tr>
<td>was examining</td>
<td>finite([past, progressive])</td>
</tr>
<tr>
<td>were examining</td>
<td>finite([past, progressive])</td>
</tr>
<tr>
<td>will have examined</td>
<td>finite([modal(will), perfect])</td>
</tr>
<tr>
<td>would have examined</td>
<td>finite([modal(would), perfect])</td>
</tr>
<tr>
<td>will be examining</td>
<td>finite([modal(will), perfect])</td>
</tr>
<tr>
<td>would be examining</td>
<td>finite([modal(would), perfect])</td>
</tr>
<tr>
<td>have been examining</td>
<td>finite([present, perfect, progressive])</td>
</tr>
<tr>
<td>has been examining</td>
<td>finite([present, perfect, progressive])</td>
</tr>
<tr>
<td>had been examining</td>
<td>finite([past, perfect, progressive])</td>
</tr>
<tr>
<td>have been examined</td>
<td>finite([present, perfect, passive])</td>
</tr>
<tr>
<td>has been examined</td>
<td>finite([past, perfect, passive])</td>
</tr>
<tr>
<td>am being examined</td>
<td>finite([present, progressive, passive])</td>
</tr>
<tr>
<td>are being examined</td>
<td>finite([present, progressive, passive])</td>
</tr>
<tr>
<td>is being examined</td>
<td>finite([present, progressive, passive])</td>
</tr>
<tr>
<td>was being examined</td>
<td>finite([past, progressive, passive])</td>
</tr>
<tr>
<td>were being examined</td>
<td>finite([past, progressive, passive])</td>
</tr>
<tr>
<td>will have been examining</td>
<td>finite([modal(will), perfect, progressive])</td>
</tr>
<tr>
<td>would have been examining</td>
<td>finite([modal(would), perfect, progressive])</td>
</tr>
<tr>
<td>will be being examined</td>
<td>finite([modal(will), perfect, passive])</td>
</tr>
<tr>
<td>would be being examined</td>
<td>finite([modal(would), perfect, passive])</td>
</tr>
<tr>
<td>have been being examined</td>
<td>finite([present, perfect, progressive, passive])</td>
</tr>
<tr>
<td>has been being examined</td>
<td>finite([present, perfect, progressive, passive])</td>
</tr>
<tr>
<td>will have been being examined</td>
<td>finite([modal(will), perfect, progressive, passive])</td>
</tr>
<tr>
<td>would have been being examined</td>
<td>finite([modal(would), perfect, progressive, passive])</td>
</tr>
<tr>
<td>examining</td>
<td>infinitive()</td>
</tr>
<tr>
<td>have examined</td>
<td>infinitive([perfect])</td>
</tr>
<tr>
<td>be examining</td>
<td>infinitive([progressive])</td>
</tr>
<tr>
<td>be examined</td>
<td>infinitive([passive])</td>
</tr>
<tr>
<td>have been examining</td>
<td>infinitive([perfect, progressive])</td>
</tr>
<tr>
<td>has been examined</td>
<td>infinitive([perfect, passive])</td>
</tr>
<tr>
<td>be being examined</td>
<td>infinitive([progressive, passive])</td>
</tr>
<tr>
<td>have been being examined</td>
<td>infinitive([perfect, progressive, passive])</td>
</tr>
<tr>
<td>examining</td>
<td>participle()</td>
</tr>
<tr>
<td>have examined</td>
<td>participle([perfect])</td>
</tr>
<tr>
<td>being examined</td>
<td>participle([progressive])</td>
</tr>
<tr>
<td>been examined</td>
<td>participle([passive])</td>
</tr>
<tr>
<td>having been examined</td>
<td>participle([perfect, progressive])</td>
</tr>
<tr>
<td>having been examined</td>
<td>participle([perfect, passive])</td>
</tr>
<tr>
<td>being being examined</td>
<td>participle([progressive, passive])</td>
</tr>
<tr>
<td>having been being examined</td>
<td>participle([perfect, progressive, passive])</td>
</tr>
</tbody>
</table>
6 Formal semantics

One part of the ideological baggage that many computational theories of natural language meaning have inherited from predicate logic is the assumption that there is a simple relation between words and meaning expressions. Major content words, like verbs and nouns, in particular, are often represented directly by identically named symbols, as can be seen in the following examples, taken from recent work in the field.

Susan suffers from\(^1\) thalassemia.  
(Bailim and Wilks, 1991, pp. 89, 105)  
suffers\(_{form}(Susan,thalassemia)\)

Fido\(^2\) bit a mailman.  
(Dinsmore, 1991, p. 110)  
bite(Fido, x) & mailman(x)

Jones owns Ulysses.  
(Ramp and Reyle, 1993, pp. 60-64)  
Jones(x) & Ulysses(y) & x owns y

Thomas rang the bell.  
(Grover et al., 1994, p. 19)  
rang\(^3\) agent = Thomas  
patient = bell

The boy bought a pizza.  
(Allen, 1995, pp. 432-433)  
boy(x) & pizza(y) & buy(x,y)\(^4\)

In all of them, nouns and verbs correspond directly to identically named semantic expressions, and none have any separate encoding of tense or aspect.

6.1 States and events

The perhaps most obvious way to encode the temporal meaning of tense and modifiers is to introduce a new predicate for each unique combination. Since this scheme ignores the common core of meaning that sentences with the same verb have, additional mechanisms are needed to link the predicates together.

Fido bit the postman.  
bite\(_{past}(Fido, x)\)
Fido bit the postman playfully.  
bite\(_{playfully\_past}(Fido, x)\)
Fido bites the postman.  
bite\(_{present}(Fido, x)\)

Another possibility, apparently first suggested by the logician C.S. Peirce (Sowa, 2000, p. 206), is to refly the action using a variable, which can then be shared by many predicates, each expressing one distinct part of the verb meaning.

Fido bit the postman.  
bite(e) & tense(e, past)
Fido bit the postman playfully.  
bite(e) & tense(e, past) & playfully(e)
Fido bites the postman.  
bite(e) & tense(e, present)  
agent(e,Fido) & patient(e,x)

The use of reified events in the logical form was also suggested by Reichenbach (1947, pp. 301-310) and, most famously, Davidson (1967). A similar approach is taken in the ‘event calculus’ (Kowalski and Sergot, 1986).\(^5\)

\(^1\)Bailim and Wilks’s sentence reads ‘Susan has thalassemia.’  
\(^2\)The actual sentence is ‘Fred’s dog bit a mailman.’  
\(^3\)This is a simplification of their representation, which includes syntactic features.  
\(^4\)Allen’s formalisation also uses quantifiers.
As discussed in section 3.2, the situation type of the verb, or sentence, controls the effect on its meaning of the various tenses and aspects. To illustrate this, the following table lists the examples from above, together with the same sentences in (present) progressive.

<table>
<thead>
<tr>
<th>Traditional</th>
<th>Computational</th>
<th>Pustejovsky</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>STATE</strong></td>
<td><strong>STATE</strong></td>
<td><strong>STATE</strong></td>
</tr>
<tr>
<td><strong>ACTIVITY</strong></td>
<td><strong>EVENT/ACTIVITY</strong></td>
<td><strong>PROCESS</strong></td>
</tr>
<tr>
<td><strong>POINT</strong></td>
<td><strong>EVENT</strong></td>
<td>—</td>
</tr>
<tr>
<td><strong>ACCOMPLISHMENT</strong></td>
<td><strong>EVENT</strong></td>
<td><strong>TRANSITION followed by STATE</strong></td>
</tr>
<tr>
<td><strong>ACHIEVEMENT</strong></td>
<td><strong>EVENT</strong></td>
<td><strong>TRANSITION followed by STATE</strong></td>
</tr>
</tbody>
</table>

Of the theories discussed here and in chapter four, those that mention situation types at all, which is three out of five, only use states and events, and, in one case, activities.\(^5\) Pustejovsky (1995a, pp. 68–73; section 3.3.2) takes a modular approach, and defines accomplishments and achievements in terms of states and ‘transitions,’ which have duration and also imply a change in the state of the world.

As mentioned on page 41, Pustejovsky’s contention, that achievements necessarily involve an extended preparatory process,\(^7\) seems dubious, and is contrary to the classical view that they are instantaneous (Vendler, 1967, pp. 102–103). Unfortunately, assuming that transitions must have duration, there is no way to represent the more traditional achievements that do not have it. And since Pustejovsky’s theory already includes activities,\(^8\) with duration, a simpler system would be obtained by having momentaneous transitions, and adding a preceding activity when necessary.

Pustejovsky’s transitions, together with the ‘events’ of Allen (1984, p. 132; see page 45), which are not included in his more computational later work (Allen, 1995), are unique among the formalised or implemented situation types discussed here in that they include a change of state,\(^9\) which, of course, is necessary for defining accomplishments and achievements.

---

\(^5\)See the table on page 82.

\(^6\)These include Allen (1995), Hintzman et al. (1995), and Kämp and Reyle (1993).

\(^7\)These are called ‘transitions’ (Pustejovsky, 1995a, p. 68).

\(^8\)Labelled ‘processes’ (Pustejovsky, 1995a, p. 68).

\(^9\)They are ‘telic,’ see page 33.
With instantaneous transitions and points, we can formalise the situation types as follows.

<table>
<thead>
<tr>
<th>Traditional</th>
<th>Formalisation</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>state</td>
<td>Yes</td>
</tr>
<tr>
<td>Activity</td>
<td>activity</td>
<td>Yes</td>
</tr>
<tr>
<td>Point</td>
<td>point</td>
<td>No</td>
</tr>
<tr>
<td>Accomplishment</td>
<td>activity + transition(&lt;X,X&gt;)</td>
<td>Yes</td>
</tr>
<tr>
<td>Achievement</td>
<td>transition(&lt;X,X&gt;)</td>
<td>No</td>
</tr>
</tbody>
</table>

These situation types are used in the definitions of eventualities\(^{10}\) as values of the attribute 'sit\_type.' The variable 'X' above identifies the thing that changes, which can be either an entity, a property, or an eventuality.

```plaintext
concept creation.
  [is_a = eventuality,
   agent = Y:person,
   sit_type = activity + transition("X,(X & (X\_location = Y\_location)))").
```

And the symbol 'creation' is then used to define the lexical semantics of verbs,\(^{11}\) such as 'write.'

```plaintext
verbsem(write...[np(\_\_common(subj)...\_\_).np(\_\_common(obj)...\_\_)].
  creation(agent=X,object=Y:letter),[]).
```

This rule covers the sense of 'write' used in 'Philip wrote a letter,' and defines it as an accomplishment. The variables (X, Y) link the noun phrase referents, Philip and the letter, via the 'agent' and 'object' attributes, to the conditions of the 'transition,' with the intended meaning that the letter comes into existence at the same location as Philip.

Duration\(^{12}\) has not been given a definitive treatment here; it is only used to make a binary distinction between instantaneous events, or sequences of them, and those that are not (see also section 6.3, and section 6.6).

### 6.2 Tense and aspect

As discussed on page 2, and page 84, the meanings of tenses and aspects can be broadly divided into 'internal' and 'external,' where the former concern the temporal profile of the situation described by the verb, and the latter its relation to other times, such as the time of speech, or other situations mentioned in the discourse. In English, external time is mostly realised by tense and perfect aspect, and internal time by aspectual verbs and progressive aspect. Quite a few constructions express components of both meanings, the perfect progressives being good examples. In addition, the same meaning can often be expressed by different forms, like future, for example, by either present, progressive, or modal verbs (see section 2.2).

\(^{10}\) concept\_2 in appendix A.3.

\(^{11}\) verbsem\_5 in appendix A.3.

\(^{12}\) duration\_2 in appendix A.6.
As a first step in the formalisation of internal and external meaning, the following table summarises the meaning components that will be used. The interpretations of the external meanings to the right follow Reichenbach (1947; sect. 3.1).

<table>
<thead>
<tr>
<th>Internal</th>
<th>External</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start</td>
<td>The beginning</td>
</tr>
<tr>
<td>End</td>
<td>The end</td>
</tr>
<tr>
<td>Imperfective</td>
<td>The middle</td>
</tr>
<tr>
<td>Repetition</td>
<td>—</td>
</tr>
<tr>
<td>(Habitual)</td>
<td></td>
</tr>
</tbody>
</table>

Each of these meanings, except 'habitual' which has not been implemented, is expressed by an operator.

<table>
<thead>
<tr>
<th>Internal</th>
<th>Lexicon</th>
<th>Morph./Syntax</th>
<th>Aspectual verb</th>
</tr>
</thead>
<tbody>
<tr>
<td>start</td>
<td>imperfective</td>
<td>repetition</td>
<td>start</td>
</tr>
<tr>
<td>imperfetive</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>end</td>
<td>present</td>
<td></td>
<td>end</td>
</tr>
<tr>
<td></td>
<td>past</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>future</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>perfect</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>timeless</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Many of the tenses will, in this system, correspond to a sequence of components, such as 'past + future + perfect' for 'would.' A complete list of verb phrases together with formalisations of their temporal meaning (compare section 2.2) is given on page 106. The external meanings are discussed in section 7.1, and the internal ones in the following section.

One of the more speculative features of the approach suggested in this thesis is the use of internal aspectual operators in the lexicon (see also section 6.4). An example can be seen in the definition of 'arrive,' which is equated to the end of a movement.

`verbesm(arrive_._[np(._._._._X_._)].end(movement(agent=X))).`

Since the attribute 'sit_type' in the definition of 'movement' contains 'transition + activity + transition,' and the 'end' operator selects the final eventuality (see section 6.3 below), the situation type of a sentence such as 'Philip arrived in Paris' is transition.

```
e12: transition(x1.location=x13.x1.location=x13)
        [concept=movement.agent=x1.object=x1.target=x13.source=x27]
```

This example is from page 115. Philip is 'x1' and Paris 'x13.'

---

13 But see page 122.
14 Implemented by apply.asp/6 in appendix A.6.
15 In the procedure verbesm/5 in appendix A.3.
16 Defined by concept/2 in appendix A.3.
6.3 Internal time

Apart from repetition, the internal meanings have the effect of selecting a part of the temporal profile of the situation. Since time moves monotonically in one direction, an assertion that a non-initial portion has happened will imply that the earlier parts of the eventuality have also taken place.

<table>
<thead>
<tr>
<th>start</th>
<th>Implied</th>
<th>Asserted</th>
</tr>
</thead>
<tbody>
<tr>
<td>imperfective end</td>
<td>Beginning &amp; middle</td>
<td>End</td>
</tr>
<tr>
<td>end</td>
<td>Beginning</td>
<td>Middle</td>
</tr>
</tbody>
</table>

Because eventualities can have different duration, the definitions of these internal aspects are expressed in terms of elapsed time rather than the simpler alternative to just count the sequence. In practice, for the current implementation, it means that a group of zero-duration eventualities are treated as one unit. These operations are implemented by apply.asp/6 and other procedures in appendix A.6.

<table>
<thead>
<tr>
<th>start</th>
<th>Asserted</th>
</tr>
</thead>
<tbody>
<tr>
<td>imperfective end</td>
<td>All initial zero-duration eventualities. (start/3)</td>
</tr>
<tr>
<td>end</td>
<td>First non-zero eventuality. (imperfective/3)</td>
</tr>
<tr>
<td></td>
<td>All final zero-duration eventualities. (end/3)</td>
</tr>
</tbody>
</table>

If 'start' or 'end' are applied to a single eventuality, as in 'cease to think,' then a transition is added, and the original eventuality becomes 'implied.'

```plaintext
ev(e0, activity, [concept=contemplation])
ev(e1, transition(e0, "e0", []))
```

The above is intended to mean that 'e1' is the end of the activity 'e0.'

6.4 Lexical semantics

The representations of the meanings of verbs and nominalizations are stored in the lexicon together with information about grammatical complements. These representations link the words to eventuality concepts which define eventuality primitives. Using logical variables, the values of attributes in the concept definition are linked together, and linked with syntactic complements. Internal aspectual meanings, expressed lexically, by aspectual verbs, or by syntactic aspects, have the effect of filtering away eventualities from the set used as the final representation of the temporal meaning of the sentence. This section contains mostly examples and declarative information from the lexicon and concept definitions. For technical descriptions of the code, see the Appendix.

---

17 Repetition has not been fully implemented.
18 Here either zero or non-zero, see page 95.
19 This is essentially the same as Galton's (1995, p. 217) 'impressive' and 'aggressive' operators.
20 verbs/5 in appendix A.3.
21 concept/2 in appendix A.3.
22 Or, more correctly, instances of situation types, see the the table on page 95.
6.4.1 Main verbs

The meanings of main verbs\textsuperscript{23} are expressed in eventualities, with attributes, and possibly aspectual operators. For the verb 'trundle,' with only a subject and no complements, the following expression applies.

\[
\text{verbsem(trundle} \ldots \text{[mp(\ldots \text{common(subj)} \ldots \text{X})].}
\]

\[
\text{movement(object=X).]}\).
\]

In the definition of 'movement,'\textsuperscript{24} there are three eventualities, as values of the attribute 'sit\_type.' The value of the 'object' attribute, which is used in the lexicon entry above, is defined to be of the type 'moveable,' and is linked to the transitions via the variable 'X' in the expression below.

\[
\text{concept(movement,}
\]

\[
\text{[is\_a = eventuality,}
\]

\[
\text{agent : person,}
\]

\[
\text{object = X\_moveable,}
\]

\[
\text{source = Y\_location,}
\]

\[
\text{target = Z\_location,}
\]

\[
\text{via : location,}
\]

\[
\text{sit\_type = transition( X\_location = Y, \text{"X"_location = Y} )}
\]

\[
+ \text{activity}
\]

\[
+ \text{transition( "X"_location = Z, X\_location = Z )].}
\]

Of the three eventualities, the 'activity,' but not the 'transitions,' have duration (see page 95), so the temporal meaning is that the object (X) immediately ceases to be at the location Y, then undergoes some activity for a period of time, and finally appears at 'Z.'

Since the lexicon entry connects 'X' to the subject of the sentence, that variable in the concept definition is given a value (x0 below) explicitly, during the parsing. But the other two are not, so two implied values (x10 & x11) are created.

\[
[\text{Philip.trundled}]
\]

\[
\text{times: t0 t1 t2}
\]

\[
\text{e0 e2 e1}
\]

\[
e0: \text{transition(x0\_location=x11,\text{"x0"_location=x11)} [...]}
\]

\[
e1: \text{activity [concept=movement,object=x0,source=x11,target=x0]}
\]

\[
e2: \text{transition(\text{"x0"_location=x10,\text{\"x0\"_location=x10)} [...]}
\]

\[
x0: \text{entity [name=Philip,\text{is_a=male]}
\]

Three times (t0, t1 & t2) are also used to represent the meaning of 'Philip trundled.' The first, 't0,' is the time of speech, and the other two are when the eventualities take place, as indicated above.

6.4.2 Verb particles

In the example text\textsuperscript{25} the verb 'trundle' is used with the aspectual particle 'off,' with the meaning that the trundling started (i.e. 'ingressive' meaning, see

\textsuperscript{23}Stored in the fourth parameter of verbsem/5, appendix A.3.

\textsuperscript{24}concept/2 in appendix A.3.

\textsuperscript{25}See sections 1.1 and 4.1.
section 2.4]. This is encoded by the lexical operator ‘start,’ as shown in the lexicon entry below: 26

\[
\text{verbsem(trundle...[sp(...common(subj),...X...).p(0)].}
\text{start(movement(object=X)),[]).}
\]

The effect of the ‘start’ operator is to pick out all initial eventualities with zero duration (see section 6.3), which in our simple example means only the first ‘transition.’

\[
[\text{Philip.trundled.off]}
\text{times: t1 to e0}
\]

\[
e0: \text{transition(z0\_location=x0,} \text{z0\_location=x0)}
\]

\[
[\text{concept=movement.object=z0,source=x0, target=x0}]
\]

\[
\text{x0: entity [name=Philip,is_a=\text{male}]} \]

In a similar way, a prepositional phrase with ‘along’ as complement has been defined as imperfective. This is somewhat arbitrary, but works for the example text and is useful here for illustration.

\[
\text{verbsem(trundle...[sp(...common(subj),...X...).p(along,...Y...)]},
\]

\[
\text{imperfective(movement(object=X,via=Y)),[]).}
\]

The lexical operator ‘imperfective’ is defined to include everything up to the first non-zero eventuality. 27 Continuing with ‘Philip trundled.’ we get the following result.

\[
[\text{Philip.trundled.along.the.street}]
\text{times: t1 to e0}
\]

\[
e0: \text{transition(x1\_location=x1,} \text{x1\_location=x1)}
\]

\[
[\text{concept=movement.object=x1, source=x1, target=x1}] \]

\[
\text{x0: entity [is_a=street]}
\]

\[
\text{x1: entity [name=Philip,is_a=\text{male}]} \]

The lexical imperfective operator selects the first two eventualities, since the ‘transition’ has no duration, and the prepositional complement gives us a new entity (x0), referring to the street along which he trundles.

6.4.3 Aspectual verbs

As mentioned in section 6.2, the internal aspectual operations can be encoded by morphology/syntax and aspectual verbs as well. The aspectual verb ‘start,’ for example, has the same meaning (ingressive), in this system, as the verb/particle combination ‘trundle off.’

\[
[\text{Philip.started.trundling] / [Philip.started.to.trundle]}
\text{times: t1 to e0}
\]

26 verbsem/5 in appendix A.3.
27 See section 6.3.
\( e_0: \text{transition}(x_0\text{.location=xi1}, \"x_0\text{.location=xi1}\) \\
[\text{concept=\text{movement,object=x0,source=xi1,target=xi0}}] \\
x_0: \text{entity [name=Philip,is_a=male]} \\
\)

Some major simplifications have been made in the analysis of aspeuctal verbs, see page 127.

### 6.4.4 Progressive aspect

The imperfective meaning of the progressive aspect is also encoded by the `imperfective` operator,\(^{28}\) which selects all eventualities up to, and including, the first with non-zero duration.

\[
[\text{Philip,is,writing,a,letter}] \\
times: t_0 \\
\hspace{1cm} e_0 \\
\hspace{2cm} e_0: \text{activity [concept=\text{creation,agent=xi1,object=x0}}] \\
x_0: \text{entity [is_a=document]} \\
x_1: \text{entity [name=Philip,is_a=male]} \\
\]

As shown on page 95, ‘write’ has two eventualities, an ‘activity’ and a ‘transition.’ The latter is not included in the imperfective version, but a perfective sense, such as simple past, has both.

\[
[\text{Philip,wrote,a,letter}] \\
times: t_1 \hspace{3mm} t_2 \hspace{3mm} t_0 \\
\hspace{1cm} e_0 \hspace{1cm} e_1 \\
\hspace{2cm} e_0: \text{activity [concept=\text{creation,agent=xi1,object=x0}}] \\
\hspace{3cm} e_1: \text{transition(\text{\"x_0,\hspace{1mm}x_0\&(\hspace{1mm}x_0\text{.location=xi1}\text{.location)}\)}} \\
\hspace{4.5cm} [\text{concept=\text{creation,agent=xi1,object=x0}}] \\
x_0: \text{entity [is_a=document]} \\
x_1: \text{entity [name=Philip,is_a=male]} \\
\]

The final eventuality, the ‘transition,’ is meant to encode that the letter \(x_0\) comes into being, and this only happens if the writing is finished.

As Pustejovsky (1995a, p. 82; sect. 3.3.2) points out, this formalisation of the imperfective meaning goes some way towards explaining the ‘imperfective paradox.’\(^{29}\)

### 6.4.5 Nominalizations

The system also handles nominalizations (see page 18). Only one exists in the lexicon,\(^{30}\) ‘arrival.’

\[
\text{word(ARRIVAL, nom(ARRIVE, singular, count, common(\_))))}. \\
\]

A link to the verb meaning\(^{31}\) is provided by ‘arrive’ inside the ‘nom’ expression.

---

\(^{28}\)See section 6.3.

\(^{29}\)See page 42.

\(^{30}\)Word/2 in appendix A.2.

\(^{31}\)Verbsem/5 in appendix A.3.
verbsem(arrive_{_},np(_\ldots through\ldots_{_}).pp(in_{_\ldots_{_}Y_{_\ldots}})].
end(movement(agent=X.object=X.target=Y)),[]).

The complement list above includes a prepositional phrase, ‘pp(in_{_\ldots_{_}Y_{_\ldots}}),’ which is matched with the noun modifier ‘in Paris’ in the example below.

[he.thought.of.her.before.Philip’s.arrival.in.Paris]
times: t1 t2 t0
e0 e1
e2
e3
e0: activity [concept=contemplation,agent=xp1.object=xp0]
e1: transition(x1=location=x10.x1=location=x10) [...]
e2: activity [concept=movement,agent=x1.object=x1.target=x0.source=x10]
e3: transition(“x1=location=x0.x1=location=x0) [...]
xp0: entity [is_a=female]
xp1: entity [is_a=male]
x0: entity [name=Paris.is_a=city]
x1: entity [name=Philip.is_a=male]

The added complexity of a verb phrase and a temporal preposition is necessary to make a proper sentence. Temporal prepositions and conjunctions are discussed in section 7.3.

6.4.6 Eventive nouns

Besides nominalizations, temporal adjuncts can modify eventive noun phrases, such as ‘tea’ as a social event. The method used here to handle this is similar to what Pustejovsky (1995a, sect. 7.1) calls ‘coercion,’ where there is only one lexical entry for the noun, and the derivation of the special sense is triggered by the type of object the noun is combined with. In the noun phrase ‘tea on the following day,’ for example, the temporal connective ‘on’ is only allowed to combine with an event, which means ‘tea’ must be converted to one. The pre-conditions for a noun to be eligible for coercion are that the lexicon entry\textsuperscript{32} contains ‘+ev,’ and that the concept definition\textsuperscript{33} includes an attribute ‘event,’ indicating the type of eventuality created.

\texttt{word(tea.noun(singular\ldots common\ldots_{_}).X.+_{_}7).[],[is_a(X.tea)])].}
\texttt{concept(tea,}
\texttt{[event=tea_event]).}

The concept ‘tea_event’ is then defined separately.

\texttt{concept(tea_event,}
\texttt{[is_a = social_event,}
\texttt{sit_type = activity ]].}

If ‘tea’ is used as a normal concrete noun, the ‘event’ attribute is ignored.

\textsuperscript{32}word/2, appendix A.2.
\textsuperscript{33}concept/2, appendix A.3.
[he.had.the.tea]
times: t1 t0
   e0
   e0: state [concept=containment,owner=xp0,object=x0]
xp0: entity [is_a=male]
x0: entity [is_a=tea]

But modification by a time adjunct triggers the 'coercion.' This sentence is from the example text.\textsuperscript{34}

[he.had.in.his.pocket.an.invitation.to.tea.on.the.following.day]
times: t0
e0
e0: activity [r_eq=x2,concept=tea_event]
times: t1 t0
e0
   e0: state [concept=containment,owner=xp0,container=x1.object=x0]
xp0: entity [is_a=male]
xp1: entity [is_a=male.pos=x1]
x0: entity [is_a=invitation.event=e0]
x1: entity [poss_by=xp1,is_a=pocket]
x2: entity [seq=next,is_a=day]

Since the eventive noun 'tea' is timeless, there is only one time point (t0) for the first proposition. The adjunct, 'x2,' provides the temporal location, but it is not interpreted further in the current implementation.

\section{6.5 Perfect aspect}

What sets perfect aspect apart from the other 'external' time specifications is the 'current relevance' that it expresses (see section 2.2). Note that the perfect meaning need not be included in all uses of a particular syntactic form. The auxiliary 'had' followed by past participle, for example, is ambiguous\textsuperscript{35} between 'past + perfect' and 'past + past.' This mapping is implemented by the procedure inf1/3 in appendix A.5.

Conrie (1976, chap. 3) distinguishes four different types of perfect meaning, based on cross-linguistic evidence. In English they are all possible, with the same syntactic form; this is not always the case in other languages (Conrie, 1976, p. 56).

\textsuperscript{34}Sections 1.1 and 4.1.
\textsuperscript{35}See page 14.
Result
A current state is the result of a past situation, as in 'He has gone to London.'

Experiential
The situation has occurred, at least once, before the present. 'I have been to London.'

Persistent situation
The situation started in the past and continues into the present. For example: 'She has lived in London ever since.'

Recent past
A situation which does not fall under any of the other three cases, but is recent, e.g. 'I have finished the book.'

It is not clear if this classification is immediately useful for an implementation, and, for simplicity, it is not supported here. Adding it, and investigating the behaviour of perfect in combination with time adjuncts (sections 2.3.4 and 2.2.1), is a topic for further research.

The approach used here for handling perfect meaning is simply to explicitly represent the reference time, which is the time when the fact that the event happened, is relevant. In the example below, that is 't0,' and the brackets indicate that the events (e0 etc) did not occur but are relevant at that point.

[Philip,had.trundled]
times: t2 t3 t0 t1
e0 e2 (e6)
e1 (e1)
(e2)
e0: transition(x0"location=x11."x0"location=x11) [...]
e1: activity [concept=movement,object=x0,source=x11,target=x10]
e2: transition("x0"location=x10.x0"location=x10) [...]x0: entity [name=Philip,is_a=male]

A slight problem is evident in this example. Since there are three eventualities (e0, e1, e2), there are also three separate expressions encoding the 'perfect' relevance. To some extent, this is counter-intuitive, because there is no explicit indication in the semantic representation that those three expressions correspond to the same linguistic item. On the other hand, this is not necessarily a practical problem, and the representation can easily be changed.

Of the theories discussed in chapter 4, only Dinsmore (1991; page 70) represents both the reference and event time of perfect, as here. Allen (1995) treats it strictly as a tense, and his semantic encoding does not indicate the reference point (see the picture on page 81). Kamp and Reyle (1993) and Hitzeman et al. (1995) ignores the event time and places the whole situation at the reference time. Kamp and Reyle also treats any perfect, regardless of actual situation type, as a state (page 49).

30Compare the restrictions on the use of time adjuncts with present perfect, page 14.
31This is the 'past perfect' meaning, not 'past of past,' see page 14 and the procedure inf1/3 in appendix A.5.
32It is implemented by the procedure relauus/3 in appendix A.6.
6.6 Formal representation of time

The range of meanings expressible by time adjuncts (section 2.3) can essentially be reduced to the following cases, where the time specifications (X, Y, and Z) can take many forms, including cardinal and ordinal times, and comparisons with event descriptions.

<table>
<thead>
<tr>
<th>Meaning</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>at X</td>
</tr>
<tr>
<td>Location in span</td>
<td>o -&gt; o</td>
</tr>
<tr>
<td></td>
<td>between Y and Z</td>
</tr>
<tr>
<td></td>
<td>o -&gt; after Y</td>
</tr>
<tr>
<td></td>
<td>-&gt; o before Z</td>
</tr>
<tr>
<td>Duration (span)</td>
<td>o -&gt; o</td>
</tr>
<tr>
<td></td>
<td>from Y to Z</td>
</tr>
<tr>
<td></td>
<td>o -&gt; since Y</td>
</tr>
<tr>
<td></td>
<td>-&gt; o until Z</td>
</tr>
<tr>
<td>Repetition and frequency</td>
<td>--</td>
</tr>
</tbody>
</table>

Prepositions and conjunctions are often ambiguous between several of these meanings (tables on page 23), and they can have non-temporal senses as well.

The lengths of spans, or durations, are not represented in the current implementation. To support the situation type computations described in section 6.3 it is necessary to distinguish between momentaneous and extended periods, and this is done but nothing more. Temporal descriptions are not interpreted, and the times are only represented symbolically. Four types of data are used, in this system, to encode temporal information.

Sequence       One time point is earlier than another
Events         An event takes place at a certain time
Relevance      The event is relevant (see section 6.5) at the time
Duration       Length between two times; only linked to a description

How this works is illustrated below using the somewhat contrived example ‘He has arrived a moment before tea.’ The arrival is encoded by the eventuality ‘e3’ which takes place at ‘t2’ and is considered relevant at ‘t1.’ The tea event, ‘e0,’ happens at ‘t0.’

<table>
<thead>
<tr>
<th>segment 1: 1 interpretation(s)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[present, perfect]</td>
</tr>
<tr>
<td>times:</td>
<td>t2 t1 t0</td>
</tr>
<tr>
<td>e1 (e3) e0</td>
<td></td>
</tr>
<tr>
<td>e2</td>
<td></td>
</tr>
<tr>
<td>e3</td>
<td></td>
</tr>
<tr>
<td>duration from t2 to t0 is x0</td>
<td></td>
</tr>
</tbody>
</table>

The times are ordered as in the list above (t2, t1, t0) and ‘x0’ refers to the description ‘a moment.’

Only a few temporal prepositions/conjunctions are supported by the system. These are ‘before,’ ‘after,’ and the synonymous ‘on,’ ‘at,’ ‘when,’ and ‘while.’

39 The symbols are explained on page 22 (and on page 130).
40 See page 95, and duration/2 in appendix A.6.
41 The database format is described on page 178.
42 See check_temp_rel1/3 in appendix A.4, and page 169.
7 Discourse interpretation

Reichenbach's time points (1947; section 3.1 and 4.2) are the obvious foundation for analysis of tense, or 'external' time, in discourse. His system is accepted with little modification by Dinsmore (1991; section 4.6), Kamp & Reyle (1993; section 4.4), and Allen (1995; section 4.7). Cutrer's (Cutrer, 1995; section 4.6) time space configurations use the same basic vocabulary (speech time, reference time, event time) but with the refinement that she looks for plausible formalisations of the possible verb group meanings, rather than plausible verb groups for a given set of formalisations, as Reichenbach seems to have done.

As discussed in chapter 4 (page 46), there are a few other factors that contribute to the inter-sentential temporal meaning of a discourse: time adjuncts, default progression (section 4.3), and conceptual relations. Of the non-temporal information, entities (noun phrases, roughly) will have to be tracked, since they provide global continuity, and are presumably the linguistic items of highest interest to readers. It seems a reasonable assumption that entities and events, including relations and modifiers, are the vast majority of expressions. There is a peripheral mist of other potential items, such as 'rhetorical relations' and 'cue phrases,' but their place in a general theory of discourse interpretation, as this work aspires to be, is not obvious, they would appear to be more common in specialised language such as argumentation, than, for example, narrative, and none occur in the example text, so they have not been included.

When the final semantic representation of a new sentence is added to the discourse, attempts are made to identify those entities that co-refer with existing entities from earlier sentences, using an algorithm loosely based on the one described in Allen (1995, chap. 14 & sect. 16.3).

7.1 External time

Cutrer's (1995) innovation, to list the possible meanings of each verb group, has been duplicated, and the resulting catalogue of formalisations is shown in the table on page 106 (see also the procedure inf1/3 in appendix A.5). The entries, whose format is described in section 6.2, for the 'external' column in that table, correspond to Reichenbach's 'fundamental forms' (Reichenbach, 1947, p. 296; listed in the table on page 28), and are the input to the discourse interpretation algorithm.

Although Reichenbach (1947, p. 293) does discuss temporal conjunctions, his time point system cannot, without extensions, account for sequences of sentences in general (section 4.2). Kamp and Reyle (1993, p. 594) points out, with a contrived example, that in a series of perfect tenses in a text, the events can form their own narrative progression, separate from (and earlier than) the 'reference time,' which, it is assumed, lies in the main progression of the discourse. From this they deduce the need for two separate reference points (section 4.4). But

---

1 See sections 1.1 and 4.1.
2 Implemented by entities/3 in appendix A.6.
<table>
<thead>
<tr>
<th>Verb group form</th>
<th>Internal</th>
<th>External</th>
</tr>
</thead>
<tbody>
<tr>
<td>finite([present])</td>
<td>Habitual</td>
<td>Timeless</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Present</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Near) past</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Near) future</td>
</tr>
<tr>
<td>finite([past])</td>
<td>Habitual</td>
<td>Past</td>
</tr>
<tr>
<td>finite([present, perfect])</td>
<td>Habitual (temporary)</td>
<td>Future</td>
</tr>
<tr>
<td></td>
<td>Repetition</td>
<td>-</td>
</tr>
<tr>
<td>finite([present, progressive])</td>
<td>Imperfective</td>
<td>Future + perfect</td>
</tr>
<tr>
<td></td>
<td>Habitual (temporary)</td>
<td>Past</td>
</tr>
<tr>
<td></td>
<td>Repetition</td>
<td>Perfect</td>
</tr>
<tr>
<td>finite([past, progressive])</td>
<td>Imperfective</td>
<td>Past + perfect</td>
</tr>
<tr>
<td>finite([modal(will)])</td>
<td>Imperfective</td>
<td>Future of past</td>
</tr>
<tr>
<td></td>
<td>Habitual</td>
<td>Future of past</td>
</tr>
<tr>
<td></td>
<td>Repetition</td>
<td>Future of past</td>
</tr>
<tr>
<td>finite([modal(would)])</td>
<td>Imperfective</td>
<td>Future</td>
</tr>
<tr>
<td></td>
<td>Habitual</td>
<td>Future</td>
</tr>
<tr>
<td></td>
<td>Repetition</td>
<td>Future</td>
</tr>
<tr>
<td>finite([modal(would), perfect])</td>
<td>Imperfective</td>
<td>Future of past + perfect</td>
</tr>
<tr>
<td></td>
<td>Habitual</td>
<td>Future of past + perfect</td>
</tr>
<tr>
<td></td>
<td>Repetition</td>
<td>Future of past + perfect</td>
</tr>
<tr>
<td>finite([modal(would), progressive])</td>
<td>Imperfective</td>
<td>Future of past + perfect</td>
</tr>
<tr>
<td></td>
<td>Habitual</td>
<td>Future of past + perfect</td>
</tr>
<tr>
<td></td>
<td>Repetition</td>
<td>Future of past + perfect</td>
</tr>
<tr>
<td>participle([])</td>
<td>Imperfective</td>
<td>Future of past</td>
</tr>
<tr>
<td></td>
<td>Habitual</td>
<td>Future of past</td>
</tr>
<tr>
<td></td>
<td>Repetition</td>
<td>Future of past</td>
</tr>
<tr>
<td></td>
<td>Simultaneous with head</td>
<td>Future (head?)</td>
</tr>
<tr>
<td>participle([perfect])</td>
<td>Imperfective</td>
<td>Prior to head</td>
</tr>
<tr>
<td></td>
<td>Habitual</td>
<td>Prior to head + future</td>
</tr>
<tr>
<td></td>
<td>Repetition</td>
<td>Prior to head + future</td>
</tr>
<tr>
<td>infinitive([])</td>
<td>Imperfective</td>
<td>Timeless</td>
</tr>
<tr>
<td></td>
<td>Habitual</td>
<td>Timeless</td>
</tr>
<tr>
<td></td>
<td>Repetition</td>
<td>Timeless</td>
</tr>
<tr>
<td>infinitive([perfect])</td>
<td>Imperfective</td>
<td>Prior to head</td>
</tr>
<tr>
<td></td>
<td>Habitual</td>
<td>Prior to head</td>
</tr>
<tr>
<td></td>
<td>Repetition</td>
<td>Prior to head</td>
</tr>
<tr>
<td>infinitive([progressive])</td>
<td>Imperfective</td>
<td>Future</td>
</tr>
<tr>
<td></td>
<td>Habitual</td>
<td>Future</td>
</tr>
<tr>
<td></td>
<td>Repetition</td>
<td>Future</td>
</tr>
<tr>
<td>infinitive([perfect, progressive])</td>
<td>Imperfective</td>
<td>Future of past + perfect</td>
</tr>
<tr>
<td></td>
<td>Habitual</td>
<td>Future of past + perfect</td>
</tr>
<tr>
<td></td>
<td>Repetition</td>
<td>Future of past + perfect</td>
</tr>
<tr>
<td>nominalization</td>
<td>Timeless</td>
<td>Timeless</td>
</tr>
</tbody>
</table>
there is no limit, in principle, to the number of separate progressions.3
Webber (1988) proposes a system, later used by (Allen, 1995, chap. 16), with
a ‘temporal focus’ point, which can serve as Reichenbachian reference point or
event point (section 3.1) and also encodes the narrative progression (Webber,
1988, p. 67), together with a stack where these time points are kept available
for later reference. The partitioned representationalists (section 4.6.3), finally,
use Reichenbach’s system, but with slightly different names.

<table>
<thead>
<tr>
<th>Reichenbach</th>
<th>Webber &amp; Allen</th>
<th>Kamp &amp; Reyle</th>
<th>Dinsmore &amp; Cutrer</th>
</tr>
</thead>
<tbody>
<tr>
<td>P. of speech</td>
<td>P. of speech</td>
<td>Reference point</td>
<td>Base space</td>
</tr>
<tr>
<td>P. of reference</td>
<td>Temporal focus</td>
<td>Narrative point</td>
<td>Focus space</td>
</tr>
<tr>
<td>P. of event</td>
<td>P. of event</td>
<td>Event space</td>
<td></td>
</tr>
</tbody>
</table>

Armed with the concept of focus time, we can now introduce the complete list
of ‘external’ temporal verb group meanings in English (see section 6.2 and
the table on page 106). The table below replaces Reichenbach’s (page 28), which
is also used by Dinsmore (page 68), Kamp & Reyle (page 50), and Allen (page 74).

<table>
<thead>
<tr>
<th></th>
<th>Old focus</th>
<th>New focus</th>
<th>Times</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>[present]</td>
<td>( T )</td>
<td>( T + D )</td>
<td>( T = \text{Base} )</td>
<td>( T : \text{Events} )</td>
</tr>
<tr>
<td>[past]</td>
<td>( T )</td>
<td>( T + D )</td>
<td>( T &lt; \text{Base} )</td>
<td>( T : \text{Events} )</td>
</tr>
<tr>
<td>[present, future]</td>
<td>( T_1 )</td>
<td>( T_3 + D )</td>
<td>( T_2 &gt; T_1 )</td>
<td>( T_3 : \text{Events} )</td>
</tr>
<tr>
<td>[past, future]</td>
<td>( T_1 )</td>
<td>( T_3 &gt; T_1 )</td>
<td>( T_2 : \text{Events} )</td>
<td></td>
</tr>
<tr>
<td>[past, past]</td>
<td>( T_1 )</td>
<td>( T_3 &gt; T_1 )</td>
<td>( T_2 &lt; T_1 )</td>
<td>( T_3 : \text{Events} )</td>
</tr>
<tr>
<td>[present, future, past]</td>
<td>( T_3 )</td>
<td>( T_3 &lt; T_2 )</td>
<td>( T_2 &gt; T_1 )</td>
<td>( T_3 : \text{Events} )</td>
</tr>
<tr>
<td>[past, perfect]</td>
<td>( T_1 )</td>
<td>( T_3 &lt; T_2 )</td>
<td>( T_2 &lt; T_1 )</td>
<td>( T_3 : \text{Events} )</td>
</tr>
<tr>
<td>[present, perfect]</td>
<td>( T_1 )</td>
<td>( T_3 &lt; T_2 )</td>
<td>( T_2 : \text{Events} )</td>
<td></td>
</tr>
</tbody>
</table>

Any number of focus times can be available when the processing starts, and the
one that is selected, and removed, gets the label from the ‘old focus’ column.
The time points in the ‘new focus’ column are added when the sentence has

3 Or ‘threads’ (Kunstyan et al., 1993; Hitzeman et al., 1995, p. 256), ‘segments’ (Allen,
been processed, and narrative progression (section 4.3) is accounted for by the time difference $D$ that is added.\footnote{Not fully implemented.}

Perfect aspect introduces an ambiguity, noted by Dinmore (1991, p. 230) and Allen (1995, p. 524), in the default progression. The second sentence in the first pair below has a different form but refers to the same time as the first sentence. In the second pair, the time referred to by the second sentence is the reference time in the perfect sentence, which is what Reichenbach’s (1947) system predicts.

<table>
<thead>
<tr>
<th>I had been there before.</th>
<th>PAST PERFECT</th>
<th>(event &amp; reference time in past)</th>
</tr>
</thead>
<tbody>
<tr>
<td>I went as a child.</td>
<td>SIMPLE PAST</td>
<td>(event time)</td>
</tr>
<tr>
<td>I had been there before.</td>
<td>PAST PERFECT</td>
<td>(event &amp; reference time in past)</td>
</tr>
<tr>
<td>I remembered it well.</td>
<td>SIMPLE PAST</td>
<td>(reference time)</td>
</tr>
</tbody>
</table>

Allen handles this by allowing a shift from perfect aspect to the same tense without perfect, as in the first example above, without changing segment (see page 75). In the second example above the time shifts, so the two sentences would be in different segments. Here, the solution is to introduce both time points (event and reference) as new focus times.

The expression ‘$T :$ Events’ in the table of external meanings (page 107) is equivalent to identifying $T$ as event point in Reichenbach’s system — it is where the content of the sentence is placed — and ‘Events relevant’ is the meaning of perfect aspect (section 6.5).

‘Base.’ Dinmore’s (1991) name for the point of speech, has been extended slightly to accommodate non-finite verbs, which none of the other theories under discussion handle (see the table on page 82). For a finite phrase it is the speech time, and for a non-finite phrase, the time of the governing verb (for the rationale, see page 16).

The constraints on, and meanings of, the ‘external’ tense/aspect combinations displayed in the table on page 107 can be stated more succinctly by making a couple of generalisations. The item ‘present,’ if it is part of the expression, must be first, and the item ‘perfect,’ if included, must be last. Given this, the table can be replaced by two rules which are applied on the items in the expression sequentially.

<table>
<thead>
<tr>
<th>Initial</th>
<th>present</th>
<th>$T = \text{Base}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>past</td>
<td>$T &lt; \text{Base}$</td>
<td></td>
</tr>
<tr>
<td>Non-initial</td>
<td>past</td>
<td>$T_y &lt; T_x$</td>
</tr>
<tr>
<td></td>
<td>future</td>
<td>$T_y &gt; T_x$</td>
</tr>
<tr>
<td></td>
<td>perfect</td>
<td>$T_y &lt; T_x$; $T_x$ : Events relevant; Add $T_y$ to focus list</td>
</tr>
</tbody>
</table>

These rules, for interpreting the external tense/aspect, are implemented by times/8 in appendix A.6.
7.2 Simple sentences

If the representation of a new sentence is simply appended to the earlier representations in the discourse, some eventualities might be duplicated. In a sentence such as 'He trundled off slowly through the gay streets and along the narrow ways,' which, once the ellipsis has been expanded, corresponds to the representations below, the first eventualities in each sentence (e0 & e1 below) are actually the same, since only one trundling is taking place.

[Philip, trundled.off, slowly, through, the.gay.streets]
times: t1 t0

e0: transition(x1",location=x3,"x1",location=x3) 
    [concept=movement.object=x1, via=x0, source=x3, target=x2]

[he, trundled, along, the.narrow, ways]
times: t3 t2

e1: transition(x4",location=x7,"x4",location=x7) [...]
e2: activity [concept=movement.object=x4, via=x5, source=x7, target=x6]

This problem is neatly solved with the help of the distinction, discussed in section 6.3, between ‘asserted’ and ‘implied’ parts of an eventuality. The part of the table of lexical aspectual operators (page 97) that is relevant for the current example is reproduced here.

<table>
<thead>
<tr>
<th>Aspectual operator</th>
<th>Implied</th>
<th>Asserted</th>
</tr>
</thead>
<tbody>
<tr>
<td>trundle off</td>
<td>start</td>
<td></td>
</tr>
<tr>
<td>trundle along</td>
<td>imperfective</td>
<td>Beginning</td>
</tr>
</tbody>
</table>

Since the ‘transition’ in the second sentence is the beginning, it is ‘implied.’ This means that the problem is reduced to checking, when adding a sentence to the discourse, whether the implied parts of it overlap with existing eventualities at, or just before, the ‘current’ time. This is done by times/8 in appendix A.6.

Worth noting is that the problem of overlapping event descriptions is not limited to aspectual operators, as can be seen from examples like ‘he ran and ran,’ where repetition is used for emphasis, and there is only one single event.

7.3 Complex sentences

Three different types of complex sentence can be distinguished. First, there are those with prepositions and conjunctions, which are formalised in the same way. This includes sentences with only one verb group and a temporal adjunct. Non-temporal prepositions and conjunctions are, with the exception of ‘and,’ ignored in the current implementation. The second type of complex sentence has a non-finite phrase as complement, and the third are sentences with more than one temporal segment.

\[\text{Only the eventualities are shown here.}\]
\[\text{Can also be written as semicolon.}\]
Between the classes of prepositions and conjunctions there is a large overlap, and when a word belongs to both it often has the same core of meaning in the two cases. For this reason, the same semantic representation is used for both, and they are referred to simply as 'temporal relations.' The parameters of these relations are either verb phrases, eventive noun phrases,⁷ or temporal but non-eventive noun phrases such as time specifications. The latter can only occur in prepositional phrases.

<table>
<thead>
<tr>
<th>LHS</th>
<th>RHS</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verb phrase</td>
<td>Conj.</td>
<td>She left before he arrived.</td>
</tr>
<tr>
<td>Verb phrase</td>
<td>Prep.</td>
<td>He arrived before lunch.</td>
</tr>
<tr>
<td>Verb phrase</td>
<td>Prep.</td>
<td>He arrived at ten.</td>
</tr>
<tr>
<td>Eventive NP</td>
<td>Prep.</td>
<td>[…] tea when she arrived.</td>
</tr>
<tr>
<td>Eventive NP</td>
<td>Prep.</td>
<td>[…] a run before breakfast.</td>
</tr>
<tr>
<td>Eventive NP</td>
<td>Prep.</td>
<td>[…] her arrival at ten.</td>
</tr>
<tr>
<td>Non-ev. NP</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Non-ev. NP</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Non-ev. NP</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

In the first case here, when the relation is expressed by a conjunction, not all possible combinations of tenses and aspects of the two verb phrases are valid. The typical case is perhaps when they are the same.

Philip left when dawn broke.  

But it is not always so, as the first sentence in the example⁸ shows.

But Philip ceased to think of her a moment after he had settled down in his carriage.

The second verb group here has been analysed as past of past rather than past perfect,⁹ since any use of perfect on the right hand side of a conjunction seems odd. The time indicated by a perfect verb group is the Reichenbachian (1947; section 3.1) reference point, not the time of the event. This means that in the sentence above, if the second verb group were a perfect, the settling down would have happened at some unspecified time in the past and there would be another, more recent, time (the reference time, ‘t’ below) when that fact is relevant.

<table>
<thead>
<tr>
<th>LHS</th>
<th>RHS</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>But Philip ceased to think of her (PAST)</td>
<td>a moment after (PAST PERFECT)</td>
<td>he had settled down in his carriage.</td>
</tr>
<tr>
<td>Event time = t + one moment</td>
<td>Event time = before t Reference time = t</td>
<td></td>
</tr>
</tbody>
</table>

But there is no indication here of why the event is relevant at ‘t,’ and no specific information about when that is. If the second phrase is taken to be past of past, then the comparison is with an event directly, which means there is a specific time, even if it is not known.

⁷Both nominalizations, and nouns such as 'tea' that refer to an event and can be modified by a temporal prepositional phrase.

⁸Sections 1.1 and 4.1.

⁹See page 14, and section 6.5.
In a sentence with a non-finite verb phrase complement, the tense of the subordinate phrase is relative to the governing phrase (see page 16).

I expected him to finish. (yesterday)
I expect him to finish. (today)
I will expect him to finish. (tomorrow)

One of the sentences in the example text,10 'When he arrived in Paris he had his luggage put on a cab,' contains both a temporal conjunction (when) and a non-finite complement (put on a cab). Since both indicate simultaneousness, all three propositions in that sentence take place at the same time.

\[
\text{[when, he arrived, in, Paris, he, had, his, luggage, put, on, a, cab]}
\]

times: t1 t3 t2 t0
\[
e_0 \ e_6
\]
e1
e2
e3
e4
e5

\[
e_0: \text{transition(xp1"location=x11","xp1"location=x11)} [...]
\]
e1: activity [concept=movement,agent=xp1,object=xp1,target=x2,source=x11]
e2: transition("xp1"location=x2,xp1"location=x2) [...] 
e3: point [concept=cssativity,agent=xp0,object=x1]
e4: transition(xp0"location=x10,"xp0"location=x10) [...] 
e5: activity [concept=movement,object=xp0,target=x0,source=x10]
e6: transition("xp0"location=x0,xp0"location=x0) [...] 

The third type of complex sentence has two, or more, syntactically linked verb phrases that are temporally unrelated, in which case they can be paraphrased by two separate sentences.

He visited the house where Mozart was born.

He visited house X.

Mozart was born in house X.

Note that there is no direct temporal connection between the two propositions. The default assumption is perhaps that the birth occurred before the visit, but a sentence like ‘In 1749, he visited the house where Mozart was born in 1756.’ is perfectly acceptable.

---

10Sections 1.1 and 4.1.
7.4 Details of the example

This section describes the analysis of the example (section 1.1, also shown below) produced by the implementation under discussion, which is included in the Appendix. Similar output for the theories reviewed in chapter 4 can be found in sections 4.4.4 (discourse representation theory), 4.5.2 (discourse grammar), and 4.7.5 (segments and stack).

But Philip ceased to think of her a moment after he had settled down in his carriage. He thought only of the future. He had written to Mrs Otter, the masseur to whom Hayward had given him an introduction, and had in his pocket an invitation to tea on the following day. When he arrived in Paris he had his luggage put on a cab and trundled off slowly through the gay streets, over the bridge, and along the narrow ways of the Latin Quarter. He had taken a room at the Hotel des Deux Ecoles, which was in a shabby street off the Boulevard du Montparnasse; it was convenient for Amitrano’s school at which he was going to work.

The system has been designed specifically for this example, and for the open word classes, the lexicon (appendix A.2) contains only those used in the text, also listed here.

<table>
<thead>
<tr>
<th>Verbs</th>
<th>Nouns</th>
<th>Adjectives</th>
<th>Adverbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>arrive</td>
<td>bridge</td>
<td>masseur</td>
<td>convenient</td>
</tr>
<tr>
<td>give</td>
<td>cab</td>
<td>moment</td>
<td>following</td>
</tr>
<tr>
<td>put</td>
<td>carriage</td>
<td>pocket</td>
<td>gay</td>
</tr>
<tr>
<td>settled</td>
<td>day</td>
<td>room</td>
<td>narrow</td>
</tr>
<tr>
<td>take</td>
<td>future</td>
<td>school</td>
<td>shabby</td>
</tr>
<tr>
<td>think</td>
<td>introduction</td>
<td>street</td>
<td></td>
</tr>
<tr>
<td>trundle</td>
<td>invitation</td>
<td>tea</td>
<td></td>
</tr>
<tr>
<td>work</td>
<td>luggage</td>
<td>ways</td>
<td></td>
</tr>
<tr>
<td>write</td>
<td>arrival</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The concept definitions (concept/2 in appendix A.3) are likewise limited to what is needed for the example. Some standard items, like logical quantifiers, have been omitted. With these exceptions, the system design is intended to be general.

To obtain the example output discussed in this section, consult the files and give the command ‘discourse.’

```
SICStus (Version)
| ?- [tears].
(Messages)

   yes
   | ?- discourse.
```

The input is processed sentence by sentence, and the output consists of times (tN), eventualities (eN), and entities (εN), which are added monotonically to the discourse representation.

11 There is nothing to stop these from being used, but they were not needed for the example.
12 The code is available online, see page 131.
'Philip ceased to think of her a moment after he had settled down in his carriage.'

The number of interpretations of each sentence depends mainly on the 'external' temporal meaning of the verb group, which is shown in the table on page 106. Although the clause on the right hand side of the conjunction here is in past perfect, which has two possible readings (page 106), only 'past of past' is possible here since 'perfect' is not allowed in this position (see page 110).

[Philip ceased to think of her, a moment after he had settled down in his carriage]

sentence has 1 segment(s)

segment 1: 1 interpretation(s)
[past] + [past,past]
times: t1 t0 now
   e2 e0
   e3 e1
   e4
duration from t1 to t0 is x3

   e0: activity [concept=contemplation,agent=x1,object=x0]
   e1: transition(e0, "e0"
   e2: transition(x1location=x4,"x1"location=x4)
       [concept=location,agent=x1,object=x1,source=x4]
   e3: activity [concept=location,agent=x1,object=x1,target=x2,source=x4]
   e4: transition("x1"location=x2,x1"location=x2)
       [concept=location,agent=x1,object=x1,target=x2,source=x4]

focus time(s): [t0,t1]

x0: entity [is_a=female]
x1: entity [name=Philip, is_a=male]
x2: entity [poss_by=x1, is_a=vehicle]
x3: entity [is_a=moment]
x4: entity [is_a=location]
centres: [x1]

The construction 'cease to think' is analysed as the end of an activity, meaning two eventualities (e0, e1). The other three (e2, e3, e4) encode the settling down, which is considered a 'movement.' The location 'x4' is where Philip (x1) was before he settled down in the carriage (x2).

'He thought only of the future.'

This sentence is relatively uncomplicated and has only one sense.

[he,thought,only,of,the,future]

sentence has 1 segment(s)

segment 1: 1 interpretation(s)
[past]
times: t0 now
   e5
e5: activity [concept=contemplation,agent=x1,object=x5]

13 The actual values used in the program can be found in inf1/3, appendix A.5.
focus time(s): [t0]

x5: entity [is_a=future]
centres: [x1]

The adverb 'only' is ignored.

'He had written to Mrs Otter,
the massière to whom Hayward had given him an introduction,
and had in his pocket an invitation to tea on the following day.'

In this sentence there are four segments: writing (e7, e8, e9), giving (e10),
having in pocket (e11), and tea (e6), all of which are temporally independent,
in the narrow sense used to define 'segments.'

[he, had, written, to, Mrs Otter,
the, massière, to, whom, Hayward, had, given, him, an, introduction,
and, had, in, his, pocket, an, invitation, to, tea, on, the, following, day]
sentence has 4 segment(s)

segment 1: 1 interpretation(s)
[]
times: t0
e6: activity [r_eq=x11, concept=tea_event]
focus time(s): [t0]

segment 2: 2 interpretation(s)
[past, perfect]
times: t2 t3 t0 now
e7 e8 (e7)
e9 (e8)
(e9)
e7: activity [concept=creation, agent=x1, object=x12]
e8: transition("x12, x12&{x12\"location=x1\"location}")
[e8: concept=creation, agent=x1, object=x12]
e9: transition(x1\"owns=x12, x6\"owns=x12)
[e9: concept=transfer, source=x1, object=x12, target=x6]
focus time(s): [t0]

segment 3: 2 interpretation(s)
[past, perfect]
times: t4 t0 now
e10 (e10)
e10: transition(x7\"owns=x8, x1\"owns=x8)
[e10: concept=transfer, object=x8, source=x7, target=x1]
focus time(s): [t0]

segment 4: 1 interpretation(s)
[past]
times: t0 now
e11
e11: state [concept=containment, owner=x1, container=x10, object=x9]

\textsuperscript{14}See page 111.
**focus time(s):** [t0]

x6: entity [name=Mrs Otter, is_a=female, is_a=mammal, is_a=person]

x7: entity [name=Hayward, is_a=male]

x8: entity [is_a=introduction, patient=x6]

x9: entity [is_a=invitation, event=e6]

x10: entity [poss_by=x1, is_a=pocket]

x11: entity [seq=next, is_a=day]

x12: entity [is_a=letter]

centres: [x1]

Both ‘write’ (segment 2) and ‘give’ (segment 3) are in past perfect, so there is
a reference time, indicated by brackets (see section 6.5).

The time adjunct ‘on the following day’ (x11) is not interpreted, so the time of
the tea event (e6), described by a timeless nominalization, is the same as the
focus time, ‘t0.’

‘When he arrived in Paris he had his luggage put on a cab
and trundled off slowly through the gay streets, over the bridge,
and along the narrow ways of the Latin Quarter.’

In this sentence there are also four segments. The first one consists of three
propositions: ‘arrive’ (e12), ‘have’ (e13), and ‘put’ (e14, e15, e16), and two rela-
tions: the temporal identity relation (when) between the first two propositions,
and subordination (see pages 16 and 111) between the last two.

The three initial eventualities in each proposition are simultaneous, but since
‘e15’ is an ‘activity,’ with duration, the last eventuality in the putting (a
‘movement’ with three eventualities) takes place at a later time (t5) than the
others.

The other three segments, ‘trundle off through,’ ‘trundle over,’ and ‘trundle along,’
respectively, illustrate the effects of ‘internal’ aspect (section 6.3). As can be
verified in appendix A.3, they have the following lexical definitions.

`verbsem(trundle...[mp(..., common(subj)..., X)..., p(off), pp(through,..., Y...)].
start(movement(object=X, via=Y)).[])`.

`verbsem(trundle...[mp(..., common(subj)..., X)..., pp(over,..., Y...)].
movement(object=X, via=Y)).[])`.

`verbsem(trundle...[mp(..., common(subj)..., X)..., pp(along,..., Y...)].
imperfective(movement(object=X, via=Y)).[])`.

In the middle one, which is an unmodified ‘movement,’ all three eventualities15
are retained (e18, e19, e20), but in the other two, which are modified by ‘start’
and ‘imperfective’ (see the table on page 97), only one eventuality remains in
each.

`[when, he, arrived, in, Paris, he, had, his, luggage, put, on, a, cab,
and, trundled, off, slowly, through, the, gay, streets, over, the, bridge,
and, along, the, narrow, ways, of, the, Latin, Quarter]`

15These are defined by concept/2 in appendix A.3, as values of the attribute ‘situate’ for
‘movement.’
sentence has 4 segment(s)

segment 1: 2 interpretation(s)
[past] + [past] + [present]
times: t0 t5 now
e12 e16
e13
e14
e15
e12: transition (x1\"location=x13,x1\"location=x13)
   [concept=movement.agent=x1,object=x1,target=x13,source=x27]
e13: point [concept=causativity.agent=x1,object=x14]
e14: transition (x1\"location=x26,x1\"location=x26)
   [concept=movement.object=x1,target=x15,source=x26]
e15: activity [concept=movement.object=x1,target=x15,source=x26]
e16: transition (x1\"location=x15,x1\"location=x15)
   [concept=movement.object=x1,target=x15,source=x26]
focus time(s): [t0,t0,t5]

segment 2: 1 interpretation(s)
[past]
times: t0 now
e17
e17: transition (x1\"location=x25,x1\"location=x25)
   [concept=movement.object=x1,via=x16,source=x25,target=x24]
focus time(s): [t0]

segment 3: 1 interpretation(s)
[past]
times: t0 t6 now
e18 e20
e19
e18: transition (x1\"location=x23,x1\"location=x23)
   [concept=movement.object=x1,via=x17,source=x23,target=x22]
e19: activity [concept=movement.object=x1,via=x17,source=x23,target=x22]
e20: transition (x1\"location=x22,x1\"location=x22)
   [concept=movement.object=x1,via=x17,source=x23,target=x22]
focus time(s): [t6]

segment 4: 1 interpretation(s)
[past]
times: t6 now
e21
e21: activity [concept=movement.object=x1,via=x18,source=x21,target=x20]
focus time(s): [t6]

x13: entity [name=Paris.is_a=city]
x14: entity [pos_by=x1.is_a=luggage]
x15: entity [is_a=vehicle]
x16: entity [mood=qay,is_a=street]
x17: entity [is_a=bridge]
x18: entity [width=narrow,is_a=street,location=x19]
x19: entity [name=L. Q.,is_a=city_part]
x20: entity [is_a=location]
x21: entity [is_a=location]
x22: entity [is_a=location]
x23: entity [is_a=location]
x24: entity [is_a=location]
x25: entity [is_a=location]
x26: entity [is_a=location]
x27: entity [is_a=location]
centres: [x1]

All the ‘location’ entities are the start and end points of the movements, and
are not mentioned explicitly in the sentence but added during the interpretation
of the concept definitions.

‘He had taken a room at the Hotel des Deux Ecoles,
which was in a shabby street off the Boulevard du Montparnasse;
it was convenient for Amitrano’s school at which he was going to work.’

In the final sentence, the two relative clauses (which . . .) introduce separate
segments, so the total number of those for the sentence is four.
The analysis of hotel room booking used here is a bit simplistic: the room (x30)
is transferred to Philip (x1) from the hotel (x29).

[he,had,taken,a,room,at,the,Hotel des Deux Ecoles,
which,was,in,a,shabby,street,off,the,Boulevard du Montparnasse,;,
it,was,convenient,for,Amitrano’s,school,at,which,he,was,going,to,work]
sentence has 4 segment(s)

segment 1: 2 interpretation(s)
[past,perfect]
times: t7 t6 now
  e22 (e22)
e22: transition (x29 "own=x30,x1" "owns=x30")
  [concept=transfer,object=x30,source=x29,target=x1]
focus time(s): [t6]

segment 2: 1 interpretation(s)
[past]
times: t6 t8 now
  e23 e24
e23: state [concept=containment,object=x29,container=x32]
e24: state [concept=proximity,object=x29,reference=x31]
focus time(s): [t8]

segment 3: 1 interpretation(s)
[past]
times: t8 now
  e25
e25: state [concept=proximity,object=x28,reference=x33]
focus time(s): [t8]

segment 4: 1 interpretation(s)
[past,future]
times: t8 t9 now
  e26
e26: activity [concept=employment.agent=x1.place=x33]
focus time(s): [t9]

x28: entity [is_a=object]
x29: entity [name=H. des D. E.,is_a=hotel]
x30: entity [is_a=room]
x31: entity [name=B. du M.,is_a=street]
x32: entity [appearance=shabby,is_a=street]
x33: entity [poss_by=x34.is_a=school,is_a=object]
x34: entity [poss=x33.name=Amitrano,is_a=person]

centres: [x1]

As was pointed out repeatedly in chapter 4, the hotel’s properties, being in a
certain street (e23 & e24), and being convenient for the school (e25), are not
merely non-instantaneous, as here, but persist throughout the discourse.

In the last proposition, the verb group ‘was going to’ is ‘past future’,\(^\text{10}\) so the
reference time (t8) is anterior to both the present (now) and the event time
(t9).

\(^{10}\)See the table on page 106.
A summary of the times for the whole discourse\(^\text{17}\) is shown below.

<table>
<thead>
<tr>
<th>times: t1 t2 t3 t4 t0 t5 t7 t6 t8 t9 now</th>
</tr>
</thead>
<tbody>
<tr>
<td>(e_2) e7 e8 e10 e0 e16 e22 e20 e24 e26</td>
</tr>
<tr>
<td>(e_3) e9 e1 e21 e25</td>
</tr>
<tr>
<td>(e_4) e5 e6 (e22)</td>
</tr>
<tr>
<td>e11</td>
</tr>
<tr>
<td>e12</td>
</tr>
<tr>
<td>e13</td>
</tr>
<tr>
<td>e14</td>
</tr>
<tr>
<td>e15</td>
</tr>
<tr>
<td>e17</td>
</tr>
<tr>
<td>e18</td>
</tr>
<tr>
<td>e19</td>
</tr>
<tr>
<td>(e7)</td>
</tr>
<tr>
<td>(e8)</td>
</tr>
<tr>
<td>(e9)</td>
</tr>
<tr>
<td>(e10)</td>
</tr>
</tbody>
</table>

\(\text{duration from } t_1 \text{ to } t_0 \text{ is } x_3\)

The following table lists the eventive words and the corresponding eventualities.

<table>
<thead>
<tr>
<th>cease</th>
<th>e1</th>
</tr>
</thead>
<tbody>
<tr>
<td>think (of her)</td>
<td>e0</td>
</tr>
<tr>
<td>settle down</td>
<td>e2, e3, e4</td>
</tr>
<tr>
<td>think (future)</td>
<td>e5</td>
</tr>
<tr>
<td>write</td>
<td>e7, e8, e9</td>
</tr>
<tr>
<td>give</td>
<td>e10</td>
</tr>
<tr>
<td>have (in pocket)</td>
<td>e11</td>
</tr>
<tr>
<td>tea</td>
<td>e6</td>
</tr>
<tr>
<td>arrive</td>
<td>e12</td>
</tr>
<tr>
<td>have (luggage)</td>
<td>e13</td>
</tr>
<tr>
<td>put</td>
<td>e14, e15, e16</td>
</tr>
<tr>
<td>trundle off through</td>
<td>e17</td>
</tr>
<tr>
<td>trundle over</td>
<td>e18, e19, e20</td>
</tr>
<tr>
<td>trundle along</td>
<td>e21</td>
</tr>
<tr>
<td>take (room)</td>
<td>e22</td>
</tr>
<tr>
<td>be in</td>
<td>e23</td>
</tr>
<tr>
<td>be off</td>
<td>e24</td>
</tr>
<tr>
<td>be convenient</td>
<td>e25</td>
</tr>
<tr>
<td>work</td>
<td>e26</td>
</tr>
</tbody>
</table>

Some amusing computerisms can be found here. In the moment between the settling down (t1) and the ceasing to think (t0), the program has inserted the writing of the letter (e7, e8, e9) and the giving of the introduction (e10). Since the lengths of events are not represented, there is no indication of how many or few letters can be written in one moment. And presumably Philip received the introduction (e10/t4) before writing the letter, but this is not encoded anywhere, so the program puts them in the order they occur in the text (t2 and t3 before t4). Another consequence of the incomplete handling of duration is that a sizeable bit of the cab ride (e17, e18, e19) have occurred before the putting of the luggage is finished (e16). The code also concludes that Philip books the room (e22/t7) at some point during the ride in the cab (t0 & t6), which seems unlikely but is perhaps not completely impossible.

\(^{17}\text{Obtained by the command 'list discourse,' after running 'discourse.'}\)
Conclusions

The discourse interpretation system described in this thesis is superior, in all respects to do with time, to other such systems in the literature, for several reasons. First of all, it is complete and self-contained. The few other systems with these qualities that are available, like Norvig's (1992), do not handle temporal information beyond reporting the syntactic form of verb phrases, and those systems that attempt to do more, like Allen's (1995), Hitzeman et al.'s (1995), and the various, most of them partial, implementations of discourse representation theory (Kamp and Reyle, 1993), are often parts of larger projects and not easily available for separate use.

Secondly, the solutions offered here to the various technical problems of interpreting time in discourse are in most cases equal to, or better than, what has been suggested earlier in the literature. Compared with the systems discussed in chapter 4, the implementation provided here makes more extensive use of basic techniques from computational linguistics and knowledge representation, and it handles a wider range of temporally related phenomena.

- The verb group parser described in chapter 5 (page 91) is simple and compact, and recognises all the phrase types listed in the table on page 12.

- As discussed in section 2.2, the mapping between syntactic form and verb group meaning is many-to-many. Simple present, for example, can express past, present, or future meaning, and the latter can also be expressed by present progressive and by modal verbs. Of the theories reviewed in chapter 4, only Cutrer (1995; section 4.6), whose work is nor formal, makes a serious attempt to list all the possibilities. Allen (1995) and Kamp and Reyle (1993) mention the problem but their formalisations only include a few examples. Here, this mapping is implemented by a table (page 106), which is reasonably complete. The major exception is that the 'politeness' meanings of simple past, progressive, and 'would' (section 2.2) are not included.

- Reichenbach's (1947) famous table of verb phrase types and time points (page 28, section 3.1) is used with little or no change by Dinsmore (1991; page 68), Kamp & Reyle (1993; page 50), and Allen (1995; page 74), among those reviewed in chapter 4. The only real alternative that has been suggested is the 'tense trees' of Hwang & Schubert (Hwang and Schubert, 1992; section 4.7.1), also discussed by Allen (1995), but they seem to achieve the same result, and are much less clear than Reichenbach's table.

In the current system, the time point table has been improved significantly (page 107), by replacing the syntactic verb group forms that Reichenbach and the others have in the first column, with the 'external' meanings (section 7.1) discussed above. This has two major advantages: it allows a greater range of verb group meanings to be correctly encoded, and it makes the interpretation much easier, since the table can be reduced to two simple rules (shown on page 108).

\(^1\)In the sense that it addresses all the major problem areas, and takes real text as input.

\(^2\)Not all the information in this table is used in the code (inf1/3 in appendix A.3).
Like Dinsmore (1991; page 70), the current implementation encodes both the reference time and event time of perfect aspect. Allen (1995) uses the event time only, and Kamp and Reyle (1993) and Hitzeman et al. (1995) use only the reference time. For further discussion of this, see section 6.5.

The various ways to express ‘internal’ aspectual meaning (section 6.2) in English, which include syntax (e.g., progressive aspect), aspectual verbs (page 8), and verb particles (section 2.4), have all been formalised in the same way (section 6.3). None of the theories in chapter 4 handles internal meaning.

The formalisation of situation types (section 3.2) used in the current system (see the table on page 95) is basic, but it is more sophisticated than any of those used in the theories in chapter 4 (see tables on pages 94 and 82). Pustejovsky’s (1995a; section 3.3.2) formalisation is similar but slightly less expressive and Moens’ (1987; section 3.3.1) is more expressive in that it includes repetition and habituals, but perhaps unnecessarily complicated in that it includes achievements and accomplishments which in the table on page 95 are composed out of simpler primitives.

Nominalizations (page 18) are fully supported in the current system, and their semantics is specified by the same clauses as for verbs. None of the systems reviewed in chapter 4 support nominalizations.

Eventive noun phrases, like ‘tea,’ which are not nominalizations, are also handled, by converting them from entities to eventualities (see section 6.4.6). This process is similar to what Pustejovsky (1995a, sect. 7.1) calls ‘coercion.’

Non-finite verb phrase complements (page 16) are also supported here (page 111), and not by any of the systems discussed in chapter 4. The relativity of their temporal meaning to the superordinate verb phrase is achieved by letting the base of the non-finite phrase be the same as the focus of the superordinate phrase.

Last but not least, the system described here allows complex sentences with subordinate clauses that are temporally unrelated to the superordinate phrase (see page 111). This problem is not mentioned by any of the theories in chapter 4, and it is handled by putting the two propositions into separate segments, as if they were expressed by two separate sentences.

All these features are significant, and taken together they mean that the system described here is, for interpreting time in discourse, superior to any other in the literature.

---

3By ‘perfect’ here is meant the ‘external’ meaning (sections 6.2 and 6.5).
4This is discussed on page 94.
5verbs/5 in appendix A.3.
6As a social event.
7Dinsmore (1991) mentions it briefly.
8See section 7.1.
Further work

While it is better than others in the literature, the system described in this thesis does fall somewhat short of being a complete model of the interpretation and representation of time in natural language. As a first step in detailing the shortcomings, the picture below shows the types of information that a complete system would need to contain, and the connections between them.

Lexicon, open classes

- Eventuality concepts
  (Verbs & eventive nouns)
  - Situation type ontology
  - Connections situations - times

- Temporal object concepts
  (Cardinal/ordinal times, temporal nouns and proper names)

Times
(Duration)

- Partitions
  (Timeless meaning, habitual meaning, modalities)
- Temporal relations
  (Prepositions and conjunctions etc.)

Recursive decomposition
(Repetition, combinations of internal aspects)

Only the major dependencies are shown here. Verbs and nouns in the lexicon are formalised using concepts from the database, and the numbers of all these are in principle unlimited. The format used to represent times is central. In the current system they are purely symbolic, but some way to encode duration is essential. The nature of these times determines the representation of temporal objects, such as dates and proper nouns like 'Christmas,' temporal relations, and the connections to situations. Some form of partitions are a natural way to
separate the normal narrative time line from other modalities, including timeless and habitual meanings. Recursive decomposition, finally, is the mechanism at work when internal meanings are combined, as in ‘begin to arrive,’ if that is taken to be the start of the end of a movement. The table below gives a summary of how these things are currently handled, and where some suggestions for improvements can be found.

<table>
<thead>
<tr>
<th>Current solution</th>
<th>Relevant literature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Times</td>
<td>Symbols, duration</td>
</tr>
<tr>
<td></td>
<td>not encoded</td>
</tr>
<tr>
<td>Connections to situations</td>
<td>One relation</td>
</tr>
<tr>
<td>Situation type ontology</td>
<td>Four primitives</td>
</tr>
<tr>
<td></td>
<td>(Table on page 95)</td>
</tr>
<tr>
<td>Temporal objects</td>
<td>Only a few</td>
</tr>
<tr>
<td></td>
<td>(Table on page 169)</td>
</tr>
<tr>
<td>Partitions</td>
<td></td>
</tr>
<tr>
<td>Recursive decomposition</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Allen (1984), Galton (1990), see page 44</td>
</tr>
<tr>
<td></td>
<td>Galton's (1990) nine relations, see page 45</td>
</tr>
<tr>
<td></td>
<td>Most other suggestions are worse, see section 6.1</td>
</tr>
<tr>
<td></td>
<td>Galion (1995)</td>
</tr>
<tr>
<td></td>
<td>Dinsmore (1991), Cutrer (1996), see section 4.6</td>
</tr>
<tr>
<td></td>
<td>Moens and Steedman (1988)</td>
</tr>
</tbody>
</table>

Regarding the representation of times, the major issue seems to be whether to use instant points, intervals, or both. This is discussed on page 44, and Galton (1995, pp. 200–210) provides a useful summary. The ‘temporal objects’ are typically part of adjuncts (section 2.3), which can take a wide variety of syntactic forms, but meaning-wise these objects really only express two things: amount of time (either position, duration, or both) and precision. When it comes to temporal relations, there are two diametrically opposed, as it were, starting points. The relations used in the current system are derived from prepositions and conjunctions (table on page 23). Only the simplest ones are included, and much more cannot be done without a way to encode duration. To see this, consider ‘She arrived during the day,’ where a punctual achievement (arrival) takes place at some point in the span of time somewhat imprecisely defined by the temporal object ‘the day.’ In the current system, both these times are just symbols and there is no way to express the relation between them. On the bright side, the set of temporal relations needed to represent the meanings of prepositions and conjunctions in English appears to be quite limited (section 6.6). Allen (1984) approaches the problem from the other direction, by considering the possible combinations of intervals, including when they overlap. He suggests a set of thirteen basic relations, and, according to Galton (1995, p. 205), five more are needed for handling instants. Whether the two varieties of relations are conceptually distinct, or they can be unified, is an open question.

Parsing and interpretation

There are few, if any, syntactic categories that cannot contain temporal information, and for some kinds of temporal meaning, the objects especially, syntactic analysis appears to be the major part of interpretation. Since this is comparatively well understood, and implementing it involves much work of practical
rather than intellectual nature, little space has been devoted to it in this thesis. While semantically uncomplicated, time adjuncts (section 2.3) can be quite complex, since it is possible to combine several types of information in the same expression, as in ‘for a short while every day or so in January’ (Quirk et al., 1985, p. 551). For the other classes of temporal meaning, the table below lists some possible syntactic realisations, and indicates which of them are supported here.¹

<table>
<thead>
<tr>
<th>Type of meaning</th>
<th>Realisation</th>
<th>Current solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>External (present/past/future/perfect)</td>
<td>Tense/perfect aspect</td>
<td>Verb group (ch. 5)</td>
</tr>
<tr>
<td>Internal (start/end/imperfective)</td>
<td>Progressive aspect&lt;br&gt;Aspectual verbs&lt;br&gt;Lexical semantics</td>
<td>Combinations not allowed²</td>
</tr>
<tr>
<td>Timeless meaning</td>
<td>Simple present/will&lt;br&gt;Non-finite phrases&lt;br&gt;when</td>
<td>Verb group&lt;br&gt;Verb group&lt;br&gt;—</td>
</tr>
<tr>
<td>Repetition/frequency</td>
<td>Progressive aspect&lt;br&gt;Adjuncts (definite)</td>
<td>Verb group&lt;br&gt;—&lt;br&gt;—</td>
</tr>
<tr>
<td>Habitual meaning</td>
<td>Simple present/will&lt;br&gt;use to&lt;br&gt;Adjuncts (indefinite)</td>
<td>Verb group&lt;br&gt;—&lt;br&gt;—</td>
</tr>
</tbody>
</table>

Frequency adjuncts (Quirk et al., 1985, pp. 541–550) can be divided into two major groups: definite ones (every Monday, on Mondays, weekly, two times, twice) which indicate repetition, and indefinite ones (usually, occasionally, a few times, always, all year round) whose meaning is more akin to habitual. It is not clear, however, if this distinction is general and reliable enough so that they can be separated already by the parser. Similarly, how to distinguish in practice between, for example, the ‘normal,’ timeless, and habitual meanings of simple present, is an open problem.

The same holds for type-shifting (section 3.3), for which relatively straightforward solutions for simple examples can be implemented in the current system (see page 150), while the general principles await somewhere beyond the present reach of our understanding.

Other problems for the sentential interpretation stage that are not currently handled include back-shift (section 2.2.4) and the ‘politeness’ meanings of simple past, progressive, and ‘would’ (sections 2.2.1 and 2.2.2).

Partitions

As discussed in chapter 4, the use of partitions has been suggested, for representing time. There is some disagreement, however, about the technical details.

<table>
<thead>
<tr>
<th>Source</th>
<th>Term</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dinsmore (1991, p. 126; section. 4.6)</td>
<td>‘partitions’</td>
<td>One per verb phrase.</td>
</tr>
<tr>
<td>Cutler (1995, section 4.6)</td>
<td>‘spaces’</td>
<td>New ones introduced</td>
</tr>
</tbody>
</table>

Apart from Allen (1995), none of these theories are anywhere near implemented.

¹Note that this concerns only the processing that is done separately for each sentence.
²See page 177.
While it seems natural to keep situations with different modality, e.g. ‘factual’ versus ‘hypothetical,’ separate, situations that only differ in tense (including perfect aspect) need to somehow be accessible together for time point calculations. Allen requires that all those in one ‘segment’ have the same tense, except that certain shifts in perfect aspect are allowed (see page 75), but he admits that this is too restrictive (Allen, 1995, pp. 523–524). Given that a partition containing only situations that are co-temporal can be constructed dynamically, if that is needed for application of an inference procedure for example, it seems reasonable to keep situations, that only differ in the ‘external’ meaning, together.

Timeless and habitual meaning, on the other hand, are good candidates for separate partitioning. As an example of the former, consider the last sentence in this passage from H.P. Lovecraft’s The Book.

Then came the first scratching and fumbling at the dormer window that looked out high above the other roofs of the city. It came as I drooped aloud the ninth verse of that primal lay, and I knew amidst my shudders what it meant. For he who passes the gateways always wins a shadow, and never again can he be alone.

The following situations are involved.

| The first scratching and fumbling came | POINT | Activity |
| I drooped aloud the ninth verse | "State" | Narrative |
| I knew amidst my shudders what it meant | | |

| The window looked out above the other roofs | STATE | Persistent |
| The who passes the gateways | ACHIEVEMENT | Timeless |
| always wins a shadow | ACHIEVEMENT | |
| Never again can he be alone | STATE | |

Those three situations that are listed first all happen to be simultaneous, which is indicated by the conjunction ‘as’ and the default principle that states overlap with the preceding situation. The statement about the window is in a separate temporal segment, and common sense tells us that is true for roughly the period that the house exists. That the final sentence is timeless is clear to a human reader, but the reasons for this interpretation are perhaps not. It cannot be the adverbial ‘always,’ since the sentence ‘For my brother always forgets his key’ is syntactically similar but gives habitual meaning. One solution is for the initial analysis to treat the timeless and habitual senses in the same way, and leave the disambiguation for a later stage of processing. But in the case of present tense and ‘will,’ those two still have to be separated from the non-partitional senses.

The automatic construction of partitioned representations during interpretation appears to be an unsolved problem. Fauconnier (1985, 1997) and Dinsmore (1991) suggest that certain linguistic items trigger this construction directly, but this is probably an overly simplistic view, because of the general ambiguity of language. In any case, there do not seem to have been any implementations.

---

3. Sections 3.1, 4.2, and 7.1.
4. This is discussed in section 4.6.3.
5. See section 4.3.
6. *I.e.* it has the pattern ‘Subject [always] predicate.’
7. The ‘space builders,’ see section 4.6.1.
Recursive decomposition

Aspectual verbs, and other ‘internal’ meanings (section 6.3), are currently handled by selecting parts of the eventualities corresponding to the unmodified meaning of the main verb, if there are several, or adding a transition, if there is only one. Movement is defined by three eventualities, and ‘arrive’ as the end of a movement (see page 96), which means only the final eventuality is kept.

\[
\begin{align*}
\text{move} & \rightarrow e0:\text{transition} + e1:\text{activity} + e2:\text{transition} \\
\text{arrive} & \rightarrow e0:\text{transition} \\
\end{align*}
\]

When the unmodified meaning is a single eventuality, as for ‘think’ (page 97), an internal operator has the effect of adding a transition that refers to the original eventuality, which is made ‘implied.’

\[
\begin{align*}
\text{think} & \rightarrow e0:\text{activity} \\
\text{case to think} & \rightarrow [e0:\text{activity}] + e1:\text{transition}(e0, "e0") \\
\end{align*}
\]

But if there is more than one internal meaning, this method is insufficient. The verb group ‘finish trundling off,’ for example, where ‘trundle off’ is equivalent to ‘start trundling,’ cannot be handled.

\[
\begin{align*}
\text{trundle} & \rightarrow e0:\text{activity} \\
\text{trundle off} & \rightarrow e0:\text{transition}("e1,e1") + [e1:\text{activity}] \\
\end{align*}
\]

A naïve approach would be to add a second transition, but since they are instantaneous (see the table on page 95), that would not capture the right meaning, which is that the ‘trundling off’ is seen not as punctual but as an extended event (an accomplishment), and that it has been completed.

\[
\begin{align*}
\text{finish trundling off} & \rightarrow e0:\text{activity} + e1:\text{transition}("e2,e2") + [e2:\text{activity}] \\
\end{align*}
\]

Here, ‘e0’ is the activity of starting the journey, and ‘e2’ the travelling itself.

As Moens and Steedman (1988, pp. 18–19; see also section 3.3.1) points out, there is an element of recursiveness here, and in principle any number of internal meanings can be combined.\(^8\)

In the example above, the start of the trundling, which is initially seen as a punctual transition, is decomposed into the activity and transition of an achievement.

\[
\begin{align*}
\text{finish trundling off} & \rightarrow e0:\text{transition}("e2,e2") + [e2:\text{activity}] \\
& \quad + e0a:\text{activity} + e0b:\text{transition}("e2,e2") \\
\end{align*}
\]

It might be useful to retain the recursive structure, i.e. not ‘flatten’ the representation, in order to attach various information to the right place. If it is stated, for example, that somebody ‘saw him trundle off,’ then that information should perhaps be connected to ‘e0’ rather than ‘e0a’ or ‘e0b’ above.

There are clear similarities between these operations and type-shifting (section 3.3), and the transitions of Moens and Steedman (1988, section 3.3.1), which include imperfective meaning but not aspectual verbs, are probably among the more reliable hints towards a solution, in the literature.

\(^8\)See Binnick’s (1991) examples on page 26.
In an analysis inspired by Moens and Steedman (1988), the operator ‘end,’ corresponding to the aspactual verb ‘cease,’ would expect to be combined with (say) an achievement. If the main verb is an activity, as in ‘cease to think,’ then a way to shift the situation type to achievement would have to be found.

The mechanism of recursive decomposition, operating in the other direction, could perhaps also be used for handling repetition, by adding a ‘repetition’ operator around the eventualities. Assuming that a nod, like a movement, is composed of two transitions and an activity, some possible meanings can be formalised as follows.

<table>
<thead>
<tr>
<th>nod</th>
<th>e0:point</th>
</tr>
</thead>
<tbody>
<tr>
<td>is nodding (Imperfective)</td>
<td>e0:point</td>
</tr>
<tr>
<td>is nodding (Repetition)</td>
<td>e0:activity + e0:transition + e0:transition</td>
</tr>
</tbody>
</table>

In the imperfective case, the first transition is implied, and the second one is selected away, since the event is not completed. This leaves the activity as the asserted meaning. Repetition, in the bottom case, is an activity composed of the point iterated (Moens and Steedman, 1988, p. 18).

Returning to aspactual verbs, ‘stop,’ ‘cease,’ and ‘finish’ have in the current implementation been given exactly the same semantics, which is a simplification. A better approximation of their true meaning is provided by the following table.

<table>
<thead>
<tr>
<th>verb</th>
<th>meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>stop</td>
<td>Temporary termination; can be resumed.</td>
</tr>
<tr>
<td>cease/abort</td>
<td>Terminated but not completed.</td>
</tr>
<tr>
<td>end/finish</td>
<td>Completed.</td>
</tr>
</tbody>
</table>

But implementing this would require a way to determine if a situation is completed, and whether one act is the continuation of another, neither of which the present code provides. It is interesting to note that the aspactual verbs that refer to the initial part of a situation, such as ‘start’ and ‘begin,’ do not exhibit any similar ambiguity.
Summary

Interpretation of temporal information in discourse is the topic of this thesis. The parts of the English language that are related to time are described, and the most important philosophical and linguistic theories on the subject, such as situation types and Reichenbach’s time points, are reviewed. A survey of computational approaches is made, with emphasis on recent attempts to model time in discourse.

A complete system has been implemented, that reads a discourse, sentence by sentence, identifies entities and events, and keeps track of the time points involved. The input is parsed by a Prolog definite clause grammar, that includes a complete syntax for English verb groups, into a logistic form, where logical variables are used to connect the expressions. This is converted into attribute-value structures representing events and entities, the latter chosen from a small set of primitives.

All the major time-related constructions in English are handled by the system. Syntactically, all possible tense and aspect forms are recognised, and there is a wide, but possibly incomplete, range of meanings for them. Temporal prepositions and conjunctions are supported, as are aspectual verbs and verb particles. The lexicon is limited to those words that occur in the example text used, and a few more chiefly from closed classes.

This discourse interpretation system is superior, in all respects to do with time, to other such systems in the literature. It makes more extensive use of basic techniques from computational linguistics and knowledge representation, and it handles a wider range of temporally related phenomena.

Theoretical advances include implementation of the ‘internal’ meaning of tense and aspect, which has been ignored by other computational theories, an improvement of Reichenbach’s time point table, and the notion of syntactically linked but temporally unrelated ‘segments,’ which is not found in previous published work.
Acknowledgements

A great debt I owe to my supervisor, Yorick Wilks, for generous support and knowledgeable criticism.

The examiners, Mark Steedman and Mark Hepple, gave me very helpful advice.

I am also indebted to the writers of textbooks in computational linguistics. In particular James Allen, John Dinsmore, and Peter Norvig.
Glossary

A few words have special, technical, meaning and are defined here.

**Situation type** For a general definition, see section 3.2. In discussions of the implementation, the term normally refers to the primitives listed in the table on page 95, which are defined in appendix A.6.

**Internal time/meaning** Modification of the temporal extent or profile of a situation. Section 6.3.

**External time/meaning** Temporal relations between a situation and other contextually given times. Section 7.1

**Eventuality** This term is used in a technical sense, to refer to those concepts (see concept/2 in appendix A.3) that have the attribute ‘is_a = eventuality’ as opposed to the things verbs refer to, which are ‘events’ and can be composed of many eventualities. To complicate matters further, an eventuality can contain several ‘situation type primitives’ (see the table on page 95), and in the final output the ‘eventualities’ listed are really these primitives, with extra information added.

**Segment** The technical meaning of this term, when discussing the implementation, is temporally unrelated parts of the same sentence (see page 111). This is similar to, but more restricted than, Allen’s (1995; section 4.7) use of the term to mean the shared reality of a set of sentences with the same tense, aspect, and mood.

The following symbols are used in front of sentences.

* Ill-formed.
? Questionable.
† Well-formed but not true, or implausible.

And these are used as operators in expressions.

- $X > Y$ Greater than. For times, $X$ is later than $Y$.
- $X < Y$ Less than. For times, $X$ precedes $Y$.
- $X \in Y$ $X$ is part of $Y$. $Y$ includes $X$. $X$ overlaps $Y$.
- $X \ni Y$ $Y$ is part of $X$.

The mathematical symbols above are used for brevity rather than precision, and should not be interpreted with excessive strictness.

In sections 2.3 and 6.6, the following symbols are used to indicate type of time span.

- $\rightarrow$ Up to the end point.
- $\circ \rightarrow$ From the start point onwards.

They are also explained on page 22.
Appendix

All the code, written in Sicstus Prolog 3.6, is listed in this Appendix, and it is also available online from the following web page.

http://www.basun.net/homepages/tomas/thesis/

The electronic distribution includes instructions and test data, which are not listed here.

Overview

This system analyses natural language discourse, with an emphasis on the interpretation of tense and aspect. Inputs are a sentence, represented as a list of words, and a discourse state, stored in the Prolog database. Modifications to this state are the outputs. The parser is a Prolog definite clause grammar with a hand-coded lexicon, and it produces a semantic representation, or logistic form, that identifies the situations and entities referred to, plus quantifiers, relations, names and other properties, and instances of concepts. The latter are drawn from a hand-coded knowledge base which is organised hierarchically and expressed in object-attribute-value format.

The logistic form is converted from the original list of expressions to attributes and values, with situations and entities kept separate and identified by symbolic names. Then the situations are divided into temporal segments, which are interpreted individually. Finally, the segments and entities are matched against the discourse state to identify co-reference and find the correct temporal locations of the new situations. During interpretation, the situations are modified by aspectual operators and replaced by primitives, and the relations are used to constrain the identification of temporal locations. The times used in the discourse database are symbolic.

To sum up, the system consists of the following modules.

- Parser (Prolog definite clause grammar)
- Semantic interpretation: logistic form $\rightarrow$ eventualities/entities/relations
- Segmentation
- Discourse interpretation: each segment processed separately

In the database, the eventualities are connected with symbolic times, and these are ordered temporally. There is no other temporal information. In particular, the lengths of durations are not calculated.
Sentence, as a list of words \( \rightarrow \) \( s/4 \) (Definite clause grammar)

\[ \text{Logicistic form (Sem)} \]

\[ \text{collect/6} \]

\[ \text{Eventualities (Evs/Verbs)} \]
\[ \text{Relations (Rela)} \]
\[ \text{Non-pronoun entities (NonPs)} \]

\[ \text{segments/5} \]

\[ \text{Segments (Segs)} \]

\[ \text{times/8} \]

\[ \text{Events (Es)} \]

\[ \text{Database} \]

\[ \text{entities/5} \]

\[ \text{New entities (NewEs)} \]
\[ \text{Centres (Centres)} \]

The graph above shows the data-flow, with the Prolog variable names used in the code, and the various modules are described in the following sections.

<table>
<thead>
<tr>
<th>Module</th>
<th>Code</th>
<th>Sections</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parser</td>
<td>( s/4 )</td>
<td>A.1, A.2</td>
</tr>
<tr>
<td>Semantic interpretation</td>
<td>( \text{collect/6} )</td>
<td>A.4</td>
</tr>
<tr>
<td>Segmentation</td>
<td>( \text{segments/5} )</td>
<td>A.5</td>
</tr>
<tr>
<td>Discourse interpretation</td>
<td>( \text{times/8, entities/5} )</td>
<td>A.6</td>
</tr>
<tr>
<td>Database format</td>
<td></td>
<td>A.6</td>
</tr>
</tbody>
</table>

Section 7.4 contains a full exposition of the analysis produced, by this system, of the example (sections 1.1 and 4.1).

\textsuperscript{1}These are passed to \textit{segments/5} to be available for 'coercion.'

\textsuperscript{2}Entities that occur in the eventuality specification (\textit{concept/2}, appendix A.3) but are not mentioned explicitly in the sentence.

\textsuperscript{3}Note that \textit{segments/5} returns a list, while \textit{times/8} takes one segment, and can succeed more than once.

\textsuperscript{4}Focus times are used in the interpretation of external meaning.

\textsuperscript{5}Centres are used for anaphora resolution.
A.1 Grammar

The parser is a definite clause grammar based on Norvig (1992, chap. 21), but with the verb group parser described in chapter five (page 91). It supports the following types of simple sentence (Quirk et al., 1985, p. 803; Norvig, 1992, pp. 725–726).

<table>
<thead>
<tr>
<th>Sentence type</th>
<th>Example</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Declarative</td>
<td>Kim likes Lee</td>
<td>declarative(Sym, Tense)</td>
</tr>
<tr>
<td>Yes/no-interrogative</td>
<td>Did you see him?</td>
<td>yq(Sym)</td>
</tr>
<tr>
<td>Wh-interrogative</td>
<td>Who did you see?</td>
<td>whq(Sym)</td>
</tr>
<tr>
<td>(NP question)</td>
<td>Who likes Lee?</td>
<td>—</td>
</tr>
<tr>
<td>(Echo question)</td>
<td>Kim likes who?</td>
<td>—</td>
</tr>
<tr>
<td>Imperative</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Exclamative</td>
<td>—</td>
<td>—</td>
</tr>
</tbody>
</table>

Several forms of complex sentences are also accepted, as well as a couple of forms of ellipsis, on an ad hoc basis. The non-terminal ‘vp’ only covers the verb group itself; complements and adjuncts are consumed by ‘modifiers.’

grammar.pl

% Copyright 2001 Tomas Ny
% http://www.cs.bnm.edu/homepages/tomas/thesis/
% Copyrighted freeware. All rights reserved.

:- use_module(library(lists), [append, reverse/2]).
s(declarative(Y, Inf1, Inf2, Sen) -->
  { empty_f1(FL0) },
  x_p(Constituent, *Vb, FL0, TopSen),
  { gappable(Constituent) },
  { f2_head(Constituent, FL0, FL) },
  clause(normal, Inf1, *FL, ClauseSen),
  { append(TopiSen, ClauseSen, Sen) }).
s(yq(S, Inf1, Inf2, Sen) -->
  { empty_f1(FL0) },
  clause(aux_inv_s, Inf1, S, FL0, Sen).
s(whq(S, Inf1, Inf2, Sen) -->
  { empty_f1(FL0) },
  x_p(Constituent, *Vb, FL0, SubjSen),
  { f2_head(Constituent, FL0, FL) },
  clause(aux_inv_s, Inf1, S, FL, ClauseSen),
  { append(SubjSen, ClauseSen, Sen) }).
s(S, Sen) -->
  word(c_conj),
s(S, Sen),
s(coord(Y1, Inf1, Y2, Inf2, Sen) -->
  s(declarative(Y1, Inf1), Sen1),
  word(c_conj),
s(declarative(Y2, Inf2), Sen2),
  { append(Sen1, Sen2, Sen) }).
s(subord(Y1, Inf1, Y2, Inf2, Sen) -->
  s(declarative(Y1, Inf1), Sen1),
  subj_conj(Y1, Y2, ConjSen),
s(declarative(Y2, Inf2), Sen2),
  { append(Sen1, ConjSen, Temp),
    append(Temp, Sen2, Sen) }).
s(subord(Y1, Inf1, Y2, Inf2, Sen) -->
  subj_conj(Y1, Y2, ConjSen),
s(declarative(Y1, Inf1), Sen1),
s(declarative(Y2, Inf2), Sen2),
  { append(Sen1, ConjSen, Temp),
    append(Temp, Sen2, Sen) }).
s(subord(Y1, Inf1, Y2, Inf2, Y3, Inf3, Sen) -->
  subj_conj(Y1, Y2, ConjSen),
  s(declarative(Y3, Inf3), Sen3).
s(declarative(V1, Inf11), Sem1),
s(subject_ellipsis(V2, Inf12, V3, Inf13), Sem2),
{ append(Sem1, Conjunction, Temp),
  append(Temp, Sem2, Sem) }.

s(subject_ellipsis(V1, Inf11, V2, Inf12), Sem) -->
{ empty_fl(FL0) },
xp(Constituent, VB, FL0, Topic, Sem),
{ gappable(Constituent),
  FL_head(Constituent, FL0, FL),
  clause(normal, Infli, V1, X, GL, C0, Sem),
  [and]
  clause(no_subj, Infli2, V2, X, FL0, C0, Sem),
  append(C0, SemC0, C0, C0, Class Sem, Sem) }.

clause(normal, Infli, V1, X, FL, Sem) -->
subject(P, P, SubjectSlot, FL1, SubjSem),
{ concat_fl(FL1, FL2, Temp),
  concat_fl(Temp, FL0, FL) },
modifiers(pre, c, verb(Y), [], FL0, PreSem),

\[ P (Inf1, P, Y, [SubjectSlot [Slots], VerbSem],
\{ subj_pred_agreed(P, P, Inf1) \}),

modifiers(post, c, verb(Y), Slots, FL0, PostSem),
  \{ append(PreSem, VerbSem, Temp),
  append(Temp, PostSem, Sem) \}).

clause(no_subj, Infli, V1, X, FL, Sem) -->
{ empty_fl(FL0) },
modifiers(pre, c, verb(Y), [], FL0, PreSem),

\[ P (Inf1, Y, [GhostSubject [Slots], VerbSem],
\{ GhostSubject = xp(\ldots \ldots X, \ldots) \}),

modifiers(post, c, verb(Y), Slots, FL0, PostSem),
  \{ append(PreSem, VerbSem, Temp),
  append(Temp, PostSem, Sem) \}).

clause(normal, Infli, V1, X, FL, Sem) -->
subject(P, P, SubjectSlot, FL1, SubjSem),
{ empty_fl(FL0) },
modifiers(pre, c, verb(Y), [], FL0, PreSem),
tokens(VP),

\[ P (Inf1, P, Y, [SubjectSlot [Slots1], VerbSem1, VP], []),
\{ subj_pred_agreed(P, P, Inf1) \}),

modifiers(post, c, verb(Y), Slots1, FL0, PostSem),
  \{ append(VerbSem1, PostSem1, Sem),
  \{ append(VerbSem1, PostSem1, VP),
  \{ subj_pred_agreed(P, P, Inf1) \}) \}],

','])

modifiers(post, c, verb(Y), Slots2, FL0, PostSem2),

\[ P (Inf1, P, Y, [SubjectSlot [Slots2], VerbSem2, VP], []),
\{ subj_pred_agreed(P, P, Inf1) \}),

', ' ,']

modifiers(post, c, verb(Y), Slots3, FL0, PostSem3),

\[ P (Inf1, P, Y, [SubjectSlot [Slots3], VerbSem3, VP], []),
\{ subj_pred_agreed(P, P, Inf1) \}),

', ' ,']

clause(no_subj, Infli, V1, X, FL, Sem) -->
{ empty_fl(FL0) },
modifiers(pre, c, verb(Y), [], FL0, PreSem),
tokens(VP),

null
nullifiers(PrePost, CatInfo, Slots, FL, Sen) --> 
  adjunct(PrePost, CatInfo, FL, AdjSen),
  { concat_fl(FL1,FL2,FL) },
nullifiers(PrePost, CatInfo, Slots, FL, NodSen),
  { append(AdjSen,NodSen,Sen) }.
nullifiers(_, [], FLO, []) --> 
  { empty_fl(FLO) }.

complement(Constituent, FL, Sen) --> 
  xp(Constituent, FL, Sen),
  { fl_one(Constituent, FL) }.

xp(sp(P, N, Case, Wh, Def, X, Sen), Wh, FL, Sen) --> 
  xp(P, N, Case, Def, X, FL, Sen).

xp(pp(Vord, Wh, S, SP, Sen), Wh, FL, Sen) --> 
  pp(Vord, Wh, S, SP, FL, Sen).

xp(adjp(V), 'wh', FLO, Sen) --> 
  { empty_fl(FLO) },
  advp(Y, Sen).

xp(adj(Vord, X), 'wh', FLO, Sen) --> 
  { empty_fl(FLO) },
  { word(Vord, adj(X, Sen)) }.

xp(p(V), 'wh', FLO, []) --> 
  { empty_fl(FLO) },
  word(particle(W)).

xp(that(Tp, Y, X), 'wh', FL, Sen) --> 
  word(that),
  clause(normal_finite(Tp), Y, X, FL, Sen).

xp(part(Tp, Y, X), 'wh', FL, Sen) --> 
  clause(no_subj, participle(Tp), Y, X, FL, Sen).

xp(binf(Tp, Y, X), 'wh', FL, Sen) --> 
  clause(no_subj, infinitive(Tp), Y, X, FL, Sen).

xp(inf(Tp, Y, X), 'wh', FL, Sen) --> 
  word(to),
  clause(no_subj, infinitive(Tp), Y, X, FL, Sen).

gappable(sp(-----)) .
gappable(c(-----)) .
gappable(pp(-----)) .
gappable(adjp(-----)) .
gappable(part(-----)) .

adjunct(pre, c_noun(X,), FL, Sen) --> 
  { empty_fl(FLO) },
  word(adj(X, Sen)) .

adjunct(post, c_verb(Y), FL, Sen) --> 
  { empty_fl(FLO) },
  advp(Y, Sen) .

adjunct(pre, c_noun(Y2), FL, [temp_rel(Y1, Y2, relative)]) --> 
  { fl_one(part_c_Y1, Y2, FL) } .

adjunct(post, c_noun(Y, 'ever'), FL, Sen) --> 
  { time_prep(Prep, S, NP, RelSen) },
  pp(Prep, S, NP, FPPSen),
  { append(RelSen, FPPSen) }.

adjunct(post, c_noun(Y, 'ever'), FL, Sen) --> 
  { empty_fl(FLO) },
  sp(_, common(obj), 'wh', X, FLO, NPSen),
  { time_prep(Prep, S, X, NP, RelSen) },
  pp(Prep, S, NP, FPPSen),
  { append(SPPSen, RelSen, Temp) },
  append(Temp, FPPSen, Sen) .

adjunct(post, c_verb(Y), FL, Sen) -->
The last few predicates here are used for implementing gap threading (Pereira and Shieber, 1987, pp. 127–129; Norvig, 1992, pp. 702–704; Allen, 1990, pp. 148–150), which is used to handle topicalization and other long-range dependencies.
A.2 Lexicon

The lexicon is stored in a single Prolog predicate word/2, whose first parameter is the word, and second has the following format, for the different word classes.

- 'be,' 'have,' 'do' verb(prin(Verb), Form, Person, Number)
- Modal verb verb(modal(Verb), Form, Person, Number)
- Main verb verb(main(Verb), Form, Person, Number)
- Proper noun noun(Case, Sym, Sem)
- Common noun noun(Number, Count, Case, Sym, Sem)
- Nominalization nom(Verb, Number, Count, Case)
- Article adj(Number, Wh, Def, Quant)
- Pronoun pron(Person, Number, Wh, Case, Sym, Sem)
- Adjective adj(Sym, Sem)
- Adverb adv(Sym, Sem)
- Preposition prep(Word)
- Conjunction conj(Sym, Sym, Sem)
- sx_conj(Sym, Sym, Sym, Sem)

The components can take these values.

<table>
<thead>
<tr>
<th>Form</th>
<th>Person</th>
<th>Number</th>
<th>Count</th>
<th>Case</th>
<th>Def</th>
</tr>
</thead>
<tbody>
<tr>
<td>base</td>
<td>1</td>
<td>singular</td>
<td>count</td>
<td>common(subjective)</td>
<td>def</td>
</tr>
<tr>
<td>present</td>
<td>2</td>
<td>plural</td>
<td>noncount</td>
<td>common(objective)</td>
<td>indef</td>
</tr>
<tr>
<td>past</td>
<td>3</td>
<td></td>
<td></td>
<td>genitive</td>
<td></td>
</tr>
<tr>
<td>pres_part</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>past_part</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For verbs and nominalizations, the Verb indexes a verbgem/5 clause containing the complements and semantics (see page 148).

`lexclosed.pl`

% Copyright 2001 Tomas By
% http://www.hans.net/homepages/tomas/thesis/
% Copyrighted freeware, all rights reserved.
:- multifile word/2.

word( be, verb(prin(be), base, _)).
word( an, verb(prin(be), present, 1,singular)).
word( are, verb(prin(be), present, 2,singular)).
word( am, verb(prin(be), present, 3,singular)).
word( are, verb(prin(be), present, 1,plural)).
word( are, verb(prin(be), present, 2,plural)).
word( was, verb(prin(be), past, 1,singular)).
word( were, verb(prin(be), past, 2,singular)).
word( was, verb(prin(be), past, 3,singular)).
word( were, verb(prin(be), past, 1,plural)).
word( were, verb(prin(be), past, 2,plural)).
word( were, verb(prin(be), past, 3,plural)).
word( has, verb(prin(be), pres_part, _)).
word( has, verb(prin(be), past_part, _)).
word( be, verb(main(be), base, _)).
word( am, verb(main(be), present, 1,singular)).
word( are, verb(main(be), present, 2,singular)).
word( are, verb(main(be), present, 3,singular)).
word( are, verb(main(be), present, 1,plural)).
word( are, verb(main(be), present, 2,plural)).
word( finished, verb(main(finish), past, 1, plural)).
word( finished, verb(main(finish), past, 2, plural)).
word( finished, verb(main(finish), past, 3, plural)).
word( finished, verb(main(finish), pres_part, ...)).
word( finished, verb(main(finish), past_part, ...)).
word( start, verb(main(start), base, ...)).
word( start, verb(main(start), present, 1, singular)).
word( start, verb(main(start), present, 2, singular)).
word( start, verb(main(start), present, 3, singular)).
word( started, verb(main(start), past, 1, singular)).
word( started, verb(main(start), past, 2, singular)).
word( started, verb(main(start), past, 3, singular)).
word( started, verb(main(start), past, 1, plural)).
word( started, verb(main(start), past, 2, plural)).
word( started, verb(main(start), past, 3, plural)).
word( starting, verb(main(start), pres_part, ...)).
word( started, verb(main(start), past_part, ...)).
word( stop, verb(main(stop), base, ...)).
word( stop, verb(main(stop), present, 1, singular)).
word( stop, verb(main(stop), present, 2, singular)).
word( stop, verb(main(stop), present, 3, singular)).
word( stopped, verb(main(stop), past, 1, singular)).
word( stopped, verb(main(stop), past, 2, singular)).
word( stopped, verb(main(stop), past, 3, singular)).
word( stopped, verb(main(stop), past, 1, plural)).
word( stopped, verb(main(stop), past, 2, plural)).
word( stopped, verb(main(stop), past, 3, plural)).
word( stopping, verb(main(stop), pres_part, ...)).
word( stopped, verb(main(stop), past_part, ...)).
word( down, particle(down)).
word( off, particle(off)).
word( the, art( singular, ^-v\h\s, def, one )).
The semi-auxiliary ‘be going to’ (Quirk et al., 1985, p. 143–144, 236) is treated as an auxiliary with the same meaning as ‘will,’ which is a simplification. A consequence of this is that a combination such as ‘will be going to’ is not allowed.

1 ex.open.pl

%; Copyright 2001 Tomas By
%; http://www.baswn.net/homepages/tomas/thesis/
%; Copyrighted freeware. All rights reserved.
:- multifile word/2.
word( will, verb(modal(will),present, _)).
word( would, verb(modal(will),past, _)).

word( be going to, verb(modal(will),present, 1, singular)).
word( are going to, verb(modal(will),present, 3, singular)).
word( 'm going to, verb(modal(will),present, 2, singular)).

word( have going to, verb(modal(have),past, 1, singular)).

word( arrive, verb(main(arrive), base, _)).
word( arrives, verb(main(arrive), present, 1, singular)).
word( arrive, verb(main(arrive), present, 2, singular)).
word( arrived, verb(main(arrive), past, 1, singular)).
word( arrive, verb(main(arrive), present, 3, singular)).
word( arrives, verb(main(arrive), present, 3, plural)).
word( arrived, verb(main(arrive), past, 3, singular)).
word( arrive, verb(main(arrive), present, 3, plural)).
word( arrives, verb(main(arrive), present, 3, plural)).
word( arriving, verb(main(arrive), pres_part, _)).
word( arrived, verb(main(arrive), past_part, _)).

word( give, verb(main(give), base, _)).
word( gives, verb(main(give), present, 1, singular)).
word( give, verb(main(give), present, 2, singular)).
word( gives, verb(main(give), present, 3, singular)).
word( give, verb(main(give), present, 1, plural)).
word( settle,  verb(main(settle), base, ˓→)).
word( settle,  verb(main(settle), present, 1,singular )).}
word( settle,  verb(main(settle), present, 2,singular )).}
word( settle,  verb(main(settle), present, 3,singular )).}
word( settle,  verb(main(settle), present, 1,plural )).}
word( settle,  verb(main(settle), present, 2,plural )).}
word( settle,  verb(main(settle), present, 3,plural )).}
word( settled, verb(main(settle), past, 1,singular )).}
word( settled, verb(main(settle), past, 2,singular )).}
word( settled, verb(main(settle), past, 3,singular )).}
word( settled, verb(main(settle), past, 1,plural )).}
word( settled, verb(main(settle), past, 2,plural )).}
word( settled, verb(main(settle), past, 3,plural )).}
word( setting, verb(main(settle), pres_part, ˓→)).
word( settled, verb(main(settle), past_part, ˓→)).
word( work, verb(main(work), base, ˓→)).
word( work, verb(main(work), present, 1,singular )).}
word( work, verb(main(work), present, 2,singular )).}
word( work, verb(main(work), present, 3,singular )).}
word( work, verb(main(work), present, 1,plural )).}
word( work, verb(main(work), present, 2,plural )).}
word( work, verb(main(work), present, 3,plural )).}
word( worked, verb(main(work), past, 1,singular )).}
word( worked, verb(main(work), past, 2,singular )).}
word( worked, verb(main(work), past, 3,singular )).}
word( worked, verb(main(work), past, 1,plural )).}
word( worked, verb(main(work), past, 2,plural )).}
word( worked, verb(main(work), past, 3,plural )).}
word( working, verb(main(work), pres_part, ˓→)).
word( worked, verb(main(work), past_part, ˓→)).
word( write, verb(main(write), base, ˓→)).
word( write, verb(main(write), present, 1,singular )).}
word( write, verb(main(write), present, 2,singular )).}
word( write, verb(main(write), present, 3,singular )).}
word( write, verb(main(write), present, 1,plural )).}
word( write, verb(main(write), present, 2,plural )).}
word( write, verb(main(write), present, 3,plural )).}
word( wrote, verb(main(write), past, 1,singular )).}
word( wrote, verb(main(write), past, 2,singular )).}
word( wrote, verb(main(write), past, 3,singular )).}
word( wrote, verb(main(write), past, 1,plural )).}
word( wrote, verb(main(write), past, 2,plural )).}
word( wrote, verb(main(write), past, 3,plural )).}
word( writing, verb(main(write), pres_part, ˓→)).
word( written, verb(main(write), past_part, ˓→)).

word( 'Philip', psnoun (common(_),X, [name(X,'Philip'), is_a(X,male)]).)
word( 'Philip', psnoun (genitive,X, [name(X,'Philip'), is_a(X,male)]).)
word( 'Mrs._Butter', psnoun (common(_),X, [name(X,'Mrs._Butter'), is_a(X,female)]).)
word( 'Hayward', psnoun (common(_),X, [name(X,'Hayward'), is_a(X,male)]).)
word( 'Paris', psnoun (common(_),X, [name(X,'Paris'), is_a(X,city)]).)
word( 'Anitran's', psnoun (genitive, X, [name(X,'Anitran'), is_a(X,person)].)
word( 'Latin_Quarter', psnoun (common(_),X, [name(X,'Latin_Quarter'), is_a(X,city_part)]).)
word( 'Hotel_wes_wex_choles', psnoun (common(_),X, [name(X,'Hotel_wes_wex_choles'), is_a(X,hotel)]).)
word( 'Boisg heard de Montparnasse', psnoun (common(_),X, [name(X,'Boisg heard de Montparnasse'), is_a(X,street)]).)

word( 'Monday', psnoun (common(_),X, [is_a(X,day), prop(X,week_day,1)]).)
word( 'Tuesday', psnoun (common(_),X, [is_a(X,day), prop(X,week_day,2)]).)
word( 'Wednesday', psnoun (common(_),X, [is_a(X,day), prop(X,week_day,3)]).)
word( 'Thursday', psnoun (common(_),X, [is_a(X,day), prop(X,week_day,4)]).)
word( 'Friday', psnoun (common(_),X, [is_a(X,day), prop(X,week_day,5)]).)
word( 'Saturday', psnoun (common(_),X, [is_a(X,day), prop(X,week_day,6)]).)
word( 'Sunday', psnoun (common(_),X, [is_a(X,day), prop(X,week_day,7)]).)

word( bridge, noun(singular, count, common(_),X,'-er', []).)
word( car, noun(singular, count, common(_),X,'-er', []).)
word( introduction, noun(singular, count, common(_),X,'-er', []).)

word( 'Tears in the Rain', 144).
[pp(to,...Y...), [is_a(X, introduction), prop(X, patient,Y)] ]).
word( invitation, noun( singular, count, common(,_),X, 'er' ),
[pp(to,...Y...), [is_a(X, invitation), prop(X, event,Y)] ]).
word( luggage, noun( singular, non count, common(,_),X, 'er' ),
[pp(to,...Y...), [is_a(X, luggage), prop(X, event,Y)] ]).
word( nassiere, noun( singular, count, common(,_),X, 'er' ),
[pp(to,...Y...), [is_a(X, nassiere), prop(X, event,Y)] ]).
word( pocket, noun( singular, count, common(,_),X, 'er' ),
[pp(to,...Y...), [is_a(X, pocket), prop(X, event,Y)] ]).
word( room, noun( singular, count, common(,_),X, 'er' ),
[pp(to,...Y...), [is_a(X, room), prop(X, event,Y)] ]).
word( school, noun( singular, count, common(,_),X, 'er' ),
[pp(to,...Y...), [is_a(X, school), prop(X, event,Y)] ]).
word( street, noun( singular, count, common(,_),X, 'er' ),
[pp(to,...Y...), [is_a(X, street), prop(X, event,Y)] ]).
word( streets, noun( plural, count, common(,_),X, 'er' ),
[pp(to,...Y...), [is_a(X, streets), prop(X, event,Y)] ]).
word( ten, noun( singular, common(,_),X, 'er' ),
[pp(to,...Y...), [is_a(X, ten), prop(X, event,Y)] ]).
word( ways, noun( plural, count, common(,_),X, 'er' ),
[pp(of,...Y...), [is_a(X, street), prop(X, location,Y)] ]).
word( moment, noun( singular, count, common(,_),X, 'er' ),
[pp(of,...Y...), [is_a(X, moment), prop(X, event,Y)] ]).
word( day, noun( singular, count, common(,_),X, 'er' ),
[pp(of,...Y...), [is_a(X, day), prop(X, event,Y)] ]).
word( future, noun( singular, common(,_),X, 'er' ),
[pp(of,...Y...), [is_a(X, future), prop(X, event,Y)] ]).
word( hotel, noun( singular, count, common(,_),X, 'er' ),
[pp(of,...Y...), [is_a(X, hotel), prop(X, event,Y)] ]).
word( letter, noun( singular, count, common(,_),X, 'er' ),
[pp(of,...Y...), [is_a(X, letter), prop(X, event,Y)] ]).
word( arrival, noun( arrive, singular, count, common(,_)) ).
word( convenient, adj( _ _ ) ).
word( following, adj( X, [prop(X, seq, next)] ) ).
word( narrow, adj( X, [prop(X, wide, narrow)] ) ).
word( shabby, adj( X, [prop(X, appearance, shabby)] ) ).
word( slowly, adv(X,[prop(Y, speed, low)] )).
word( only, adv(_ _ ) )

Some proper names that consist of more than one word have been coded as single symbols, to make things easier.
A.3 Meaning representation

The semantic representation is divided into a logistic form, that is returned by the parser, and a concept hierarchy, part of which consists of eventualities, defined using the situation type primitives in the table on page 95. How these are related is illustrated by the picture below.

\[
\begin{array}{ccc}
\text{Situation type} & \text{ontology}^1 \\
\downarrow & \\
\text{Eventualities} & \text{Concepts} \\
(concept/2)^2 & (concept/2) \\
\downarrow & \downarrow & \downarrow \\
\text{Logic form:} & \text{sit/4} & \text{ref/4} & \text{temp.rel/3} \\
\downarrow & \downarrow & \downarrow & \\
\text{Eventualities} & \text{Entities} & \text{Relations} \\
(vb/3) & (de/3) & (rel/3)
\end{array}
\]

The logistic form is a flat list of expressions, and it is converted, by collect/6, into eventualities, entities, and relations (see the graph on page 132). Many of the rules in the grammar and lexicon return pieces of the logistic form; for verbs and nominalizations they are stored in verbsam/5.

**Logistic form**

In addition to the three types of expressions shown above, which correspond to grammatical constituents, there are five lesser types.

\[
\begin{array}{ll}
\text{sit} (SymV, SymNP, refl, Events) & \text{Verb phrases} & \text{&} & \text{Nominalizations} \\
\text{ref} (SymNP, Pron, Wh, Ev) & \text{Noun phrases} & \\
\text{temp.rel} (Sym, RelType, Sym) & \text{Prepositions} & \text{&} & \text{Conjunctions} \\
\text{quant} (Type, DeJ, SymNP) & \text{Quantifier} \\
\text{name} (Sym, Atom) & \text{Proper name} \\
\text{is.a} (Sym, Concept) & \text{Concept instance} \\
\text{prop} (Sym, Attribute, Value) & \text{Property} \\
\text{pos} (SymNP, SymNP) & \text{Possession (e.g. Genitive)}
\end{array}
\]

The Sym's are uninstantiated Prolog variables which are used to link the expressions together. The other symbols are explained on the next page. There are no definite rules for how the five minor expression types are distributed in the grammar, and it has mostly been decided by trial and error.

---

1 See the table on page 95.

2 Eventualities are distinguished from other concepts by two attributes: ‘is.a – eventuality’ and ‘sit.type,’ with an event structure as value.
This table shows the content of the expressions in the logistic form.

<table>
<thead>
<tr>
<th>Infl</th>
<th>Tense and aspect, see section A.4.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Events</td>
<td>Event structure (fourth parameter in verbsem/5)</td>
</tr>
<tr>
<td>Pron</td>
<td>–pron / +pron (Identifies pronouns)</td>
</tr>
<tr>
<td>Wh</td>
<td>–wh / +wh</td>
</tr>
<tr>
<td>Ev</td>
<td>–ev / +ev (Identifies eventive noun phrases)</td>
</tr>
<tr>
<td>Redtype</td>
<td>before / before(Sym) / after / after(Sym) / when / while / since / until / for / at / in / on / relative / and / but</td>
</tr>
<tr>
<td>Type</td>
<td>one / some / all (Quantifier type)</td>
</tr>
<tr>
<td>Def</td>
<td>def / indef</td>
</tr>
<tr>
<td>Concept</td>
<td>Atom that identifies a concept (concept/2)</td>
</tr>
</tbody>
</table>

There are a number of common sense restrictions on these expressions, such as: a variable should not identify more than one eventuality (sit) or entity (ref); there should not be more than one quantifier per variable; the two variables in ‘temp:rel’ and ‘poss’ should be distinct; and so on. None of these are verified.

**Verb lexical semantics**

As mentioned earlier, the lexical semantics of verbs are stored in a separate predicate, with the form `verbsem(Verb, SymV, Subcat, Events, Sem)` where `Events` is the event structure (see below). `Sem` is logistic form, and `Subcat` is the sub-categorisation specification, which is a list of expressions from the following table, indicating the types of the complements. Note that the sub-cat. list should always start with an ‘np’ for the grammatical subject.

<table>
<thead>
<tr>
<th>Category</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun phrase</td>
<td>np(Person, Number, Case, Wh, Def, SymNP, Sem)</td>
</tr>
<tr>
<td>Adjective</td>
<td>adj(Sym)</td>
</tr>
<tr>
<td>Adverbal phrase</td>
<td>advp(SymV, Sym)</td>
</tr>
<tr>
<td>Prepositional phrase</td>
<td>pp(Word, Wh, SymH, SymNP, Sem)</td>
</tr>
<tr>
<td>Particle</td>
<td>p(Word)</td>
</tr>
<tr>
<td>That-clause</td>
<td>that(Ext, Sym, Sym)</td>
</tr>
<tr>
<td>Participle clause</td>
<td>part(Ext, Sym, Sym)</td>
</tr>
<tr>
<td>Bare infinitive</td>
<td>binf(Ext, Sym, Sym)</td>
</tr>
<tr>
<td>Inflitive</td>
<td>infl(Ext, Sym, Sym)</td>
</tr>
</tbody>
</table>

For further explanations of the symbols, see the table on page 139. The `Ext` in the non-finite phrases is a list of ‘external’ meanings, see section 6.2 and the table on page 92. Prolog variables (the `Sym`’s) are used as links between the complements and the information in the event structure, since collect/6 preserves the identity of variables through the conversion. The parsing of complements is implemented by complement/6 in appendix A.1.

The event structure contains one or more eventuality symbols, possibly with

---

3Used only for additional information, besides the sit/4 expression.  
4That is, a noun phrase, for example, and the corresponding entity will be identified by the same variable.  
5Referring to eventualities (concept/2).
attributes and aspectual operators. It is stored in the sit/4 expression of the logical form, and is converted, by collect/6, into situation type primitives.

The BNF below describes the form of the event structure. The meaning of ':' is that the value, which need not be given, should be an instance of the concept.

All attributes used in the event structure must also occur in the definition of the corresponding eventuality (concept/2).

<events> ::= <situations> |
  <asp. op.> "(" <situations> ")"
<asp. op.> ::= "start" |
  "end" |
  "imperfective"
<situations> ::= <sit. spec.> { ":+" <sit. spec.> }
<sit. spec.> ::= <eventuality> |
  <eventuality> ":=" (<tuple> { ":." <tuple> }* ")"
	<tuple> ::= <attribute> ":=" <value> |
  <attribute> ":=" <variable> |
  <attribute> ":=" <concept> |
  <attribute> ":=" <variable> ":=" <concept>
Here follow the lexical semantics of the verbs. The event structure is the fourth parameter and the sub-categorisation list number three.

verbsem.pl

Copyright 2001 Tomas By
http://www.sasun.net/homepages/tomas/thesis/

Copyrighted freeware. All rights reserved.

verbsem(he, [sp(____, common(subj),____, X, ...), adj(common X), pp(for,____, Y, ...)], proximity(object=X, reference=close, []).
verbsem(he, [sp(____, common(subj),____, X, ...), adj(common X)], proximity(object=X, distance=close, [1]).
verbsem(he, [sp(____, common(subj),____, X, ...)], pp(in,____, Y, ...), pp(off,____, X, ...),
  proximity(object=X, container=Y) * proximity(object=reference, distance=close, reference=2), [1]).
verbsem(he, [sp(____, common(subj),____, X, ...)], pp(in,____, Y, ...), pp(off,____, X, ...),
  proximity(object=X, container=Y), [1]).
verbsem(he, [sp(____, common(subj),____, X, ...)],
  pp(in,____, Y, ...), sp(____, common(obj),____, X, ...),
  proximity(object=X, distance=close, reference=Y), [3]).
verbsem(have, V1, [sp(____, common(subj),____, X, ...),
  sp(____, common(obj),____, Y, ...),
  causativity(agent=object=Y), temp_rel(V1, V2, relative)]).
verbsem(have, [sp(____, common(subj),____, X, ...),
  pp(in,____, Y, ...), sp(____, common(obj),____, X, ...),
  proximity(object=object=Y, container=X), [1]).
verbsem(have, [sp(____, common(subj),____, X, ...),
  sp(____, common(obj),____, Y, ...),
  proximity(object=object=Y, object=X), [1]).
verbsem(verbsem, [sp(____, common(subj),____, X, ...),
  sp(____, common(obj),____, Y, ...)],
  causativity(agent=object=Y), temp_rel(V1, V2, relative)]).
verbsem(verbsem, [sp(____, common(subj),____, X, ...),
  sp(____, common(obj),____, Y, ...),
  proximity(object=object=Y, container=X), [1]).
verbsem(verbsem, [sp(____, common(subj),____, X, ...),
  sp(____, common(obj),____, Y, ...),
  proximity(object=object=Y, object=X), [1]).
verbsem(verbsem, [sp(____, common(subj),____, X, ...),
  sp(____, common(obj),____, Y, ...),
  proximity(object=object=Y, container=X), [1]).
verbsem(verbsem, [sp(____, common(subj),____, X, ...),
  sp(____, common(obj),____, Y, ...),
  proximity(object=object=Y, object=X), [1]).
Some examples of different verb phrases and their semantics can be found in section 6.4.
Type-shifting

Type-shifting (section 3.3) refers to systematic variations in the situation type of a verb sense depending on the presence and properties of the subject, objects, and adjuncts. In this implementation, no such systematic mechanisms exist, but these effects can be achieved for individual words by providing several `verbsem` clauses. The phrasal verb 'trundle off,' for example, differs from 'trundle' in that it refers to the start of the movement only (see section 6.4.2).

```
verbsem(trundle...
  [np(_...common(subj)_...X_...)]
  movement(object=X),[]).
verbsem(trundle...
  [np(_...common(subj)_...X_...).p(off)]
  start(movement(object=X)),[]).
```

In a similar way, the type-shifting examples from section 3.3 can be handled. All the event structures below are labelled '???' to indicate that they have not actually been implemented.

Optional verb complements and adjuncts can be easily detected.

```
Paavo is running.                        ACTIVITY
Paavo is running a mile.                ACCOMPLISHMENT
Paavo is running to the hills.          ACCOMPLISHMENT

verbsem(run...
  [np(_...common(subj)_...X_...)]
  ???(agent=X),[]).
verbsem(run...
  [np(_...common(subj)_...X_...).np(_...common(obj)_...Y_...)]
  ???(agent=X, distance=Y),[]).
verbsem(run...
  [np(_...common(subj)_...X_...).pp(to,...Y,...)]
  ???(agent=X, target=Y),[]).
```

The second element in the 'np/?' terms indicates number, and the fifth whether the noun phrase is definite (def) or not (indef).

```
Wilhelm ate the apples.                  ACCOMPLISHMENT
Wilhelm ate apples.                     ACTIVITY

verbsem(ate...
  [np(_...common(subj)_...X_...).np(_...common(obj)_...def,Y_...)]
  ???(agent=X, patient=Y),[]).
verbsem(ate...
  [np(_...common(subj)_...X_...).np(_...plural.common(obj)_...indef,Y_...)]
  ???(agent=X, patient=Y),[]).
```

The student scored well on the test.     ACCOMPLISHMENT
Students scored well on the test.         ACTIVITY

```
verbsem(score...
  [np(_...common(subj)_...def,X_...).pp(on,...Y_...)]
  ???(agent=X, instrument=Y),[]).
verbsem(score...
  [np(_...plural.common(subj)_...indef,X_...).pp(on,...Y_...)]
  ???(agent=X, instrument=Y),[]).
```

Since this approach, however, involves separate implementations for each verb of what appears to be much more general mechanisms, there is some potential for improvement. This is discussed on page 124.
Concepts and eventualities

Concepts (concept/2) are defined using attributes and values. The only attribute, in the non-eventuality concepts, that is used by the code is ‘is_a,’ which has its usual meaning of hyponymy. For eventualities, the ‘sit_type’ attribute and the ‘thematic roles’ are also used. The syntax of the concept definitions is specified by the following BNF.

```
<concept> ::= "concept(" name "," <tuple> { "," <tuple> }* ")"
<tuple> ::= <attribute> "=" <value>       | <attribute> "=" <variable>       | <attribute> ":" <concept>       | <attribute> "=" <variable> ":" <concept>       | "sit_type" "=" <type spec.> { "+" <type spec.> }*       | "restriction" "=" <restriction>
<type spec.> ::= <sit. type>       | <sit. type> "(" <statement> { "," <statement> }* ")"
<sit. type> ::= "state"       | "point"       | "transition"       | "activity"
<statement> ::= <variable>       | <attr_spec> "=" <variable>       | <attr_spec> "=" <attr_spec>       | "=" <statement>       | <statement> "&" <statement>
<restriction> ::= <condition> { "&" <condition> }*
<condition> ::= <attr_spec> <comp_op> <attr_spec>
<attr_spec> ::= <variable> "+" <attribute>
<comp_op> ::= ">" | "<<" | ">=\" | "<<\" | ">=\" | "<<\"
```

The mathematical operators\(^7\) have not been implemented, beyond being accepted syntactically, but the following operators are used.

\[
= \quad \text{Assignment/equality} \\
\text{:\quad Same as the} \text{'is_a'} \text{attribute: the value is an instance}\(^8\) \text{of the concept} \\
+ \quad \text{Sequence} \\
- \quad \text{Indirection: refers to the value of the attribute in the concept that the} \text{variable indicates}
\]

Much, if not most, of the data in the concepts are not used for any constructive purposes by the program, but they are checked for consistency, as described in

\(^6\)These are ‘agent,’ ‘object,’ ‘source,’ ‘target,’ and a few more.
\(^7\)Negation (‘\text{~}’), conjunction (‘&’), and the comparison operators (compare).
\(^8\)There is no formal distinction between concrete objects and abstract concepts.
section A.4, and they are useful for testing the robustness of the code.

```
concepts.pl
% Copyright 2001 Tomas By
% http://www.basun.net/homepages/tomas/thesis/
% Copyrighted freeware. All rights reserved.
:- multiset concept/2.
concept(noveable,
  [location : location]).
concept(location,
  [location_size : number]).
concept(object, []).
concept(city,
  [has_part = city_part]).
concept(city_part,
  [has_part = street,
    has_part = bridge,
    has_part = building]).
concept(street,
  [is_a = location,
    location_size = 3]).
concept(bridge,
  [is_a = location,
    location_size = 3]).
concept(building,
  [has_part = room]).
concept(room,
  [is_a = location,
    location_size = 2]).
concept(person,
  [is_a = noveable,
    has_part = pocket,
    noveable_size = 2,
    owns : object]).
concept(male,
  [is_a = person]).
concept(female,
  [is_a = person]).
concept(pocket,
  [is_a = location,
    location_size = 1]).
concept(vehicle,
  [is_a = noveable,
    is_a = location,
    noveable_size = 3,
    location_size = 2]).
concept(luggage,
  [is_a = noveable,
    noveable_size = 2]).
concept(document,
  [is_a = noveable,
    noveable_size = 1]).
concept(letter,
  [is_a = document]).
```
concept(invitation,  
  [is_a = document]).

concept(introduction,  
  [is_a = document]).

concept(tea,  
  [event=tea_event]).

concept(hotel,  
  [is_a = building]).

concept(school,  
  [is_a = building]).

concept(massiere,  
  [is_a = person]).

concept(future, []).

concept(day, []).

concept(moment, []).

Since the mathematics has not been implemented, the eventualities also contain data, including for example the 'restriction' attributes, that are meaningless.

events.pl

% Copyright 2001 Tomas Ry
% http://www.hun.net/homepages/tomas/thesis/
% Copyrighted freeware. All rights reserved.

:- multifile concept/2.
:- op(900,ry,[ ]).
:- op(600,xfy,[ ]).

concept(movement,  
  [is_a = eventuality,  
    agent : person,  
    object = X:novelse,  
    source = Y:location,  
    target = Z:location,  
    via : location,  
    sit_type = transition( X:location = Y, X:location = Y )  
      + activity  
      + transition ( Y:location = Z, Y:location = Z ),  
    restriction = ( X:novelse <= Y:location_size )  
      & ( Z:novelse <= Z:location_size ) ]).

concept(transfer,  
  [is_a = eventuality,  
    object = X,  
    source = Y:person,  
    target = Z:person,  
    sit_type = transition( Y:owns = X, Z:owns = X ) ]).

concept(creation,  
  [is_a = eventuality,  
    object = X,  
    agent = Y:person,  
    sit_type = activity + transition( X, (X:location = Y:location) ) ]).

concept(employment,  
  [is_a = eventuality,  
    agent : person,  
    place : building,  
    sit_type = activity ]).

concept(social_event,  
  [is_a = eventuality,  
    host : person,  
    guest : person,  
    sit_type = activity ]).


concept(tea_event,
    [is_a = social_event,
     sit_type = activity]).

counting(reflection,
    [is_a = eventuality,
     agent : person,
     object : person,
     sit_type = activity]).

counting(containment,
    [is_a = eventuality,
     owner : person,
     object : object,
     container : object,
     sit_type = state]).

counting(proximity,
    [is_a = eventuality,
     object : building,
     reference : building,
     distance,
     sit_type = state]).

counting(causativity,
    [is_a = eventuality,
     agent : person,
     object : object,
     action : eventuality,
     sit_type = point]).

The value of the 'sit_type' attribute should be a sequence of situation type primitives (see the table on page 95), combined using the '+' operator.
A.4 Semantic interpretation

The procedure collect/6 takes the logistico form (see page 146) and returns situation type primitives, entities, and temporal relations. At first sight, this might seem like a simple operation, but there are some complications.

- If there is more than one eventuality in the event structure (see the BNF on page 148), they cannot be converted to primitives individually as they might share variables.

- When an eventuality from the event structure is expanded with its definition (concept/2), the two lists of attributes must be merged, which involves various forms of consistency checking.

- Both the lexical semantics (verbsem/5) and concept definitions (concept/2) contain variables that sometimes do not refer to entities mentioned in the sentence. In that case, ‘implied entities’ are created, unless the variable occurs only once. To determine this, it is necessary to assemble all the variables in the logistico form of the sentence, and count them.

Since common nouns, but not pronouns, are to be made available later for ‘coercion’, and the implied entities are not used for co-reference resolution, collect/6 returns three separate lists of entities. The following variable names and data types are used for these, and for the eventualities and relations.

<table>
<thead>
<tr>
<th>Evs2</th>
<th>amp(Sym, AV, I1, I2, Struct)</th>
<th>Verb groups w. asp. verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evs3</td>
<td>vb(Sym, I, Struct)</td>
<td>Verb groups without it</td>
</tr>
<tr>
<td>Es</td>
<td>de(Sym, Prom, Ev, As)</td>
<td>All entities</td>
</tr>
<tr>
<td>Prons/Ps</td>
<td>de(Sym, Ev, As)</td>
<td>Pronouns</td>
</tr>
<tr>
<td>NomPs</td>
<td>de(Sym, Ev, As)</td>
<td>Common nouns; names</td>
</tr>
<tr>
<td>ImplEs/ImpEs</td>
<td>de(Sym, Ev, As)</td>
<td>—</td>
</tr>
<tr>
<td>Rs</td>
<td>rel(Res, Sym1, Sym2)</td>
<td>Prepositions; conjunctions</td>
</tr>
</tbody>
</table>

The table below lists the possible values of the elements.

<table>
<thead>
<tr>
<th>AV</th>
<th>I1/I2</th>
<th>Asp</th>
<th>Ops</th>
<th>Pron</th>
<th>Ev</th>
</tr>
</thead>
<tbody>
<tr>
<td>start</td>
<td>finite(...)</td>
<td>vb(I, Ops)</td>
<td>start</td>
<td>-prom</td>
<td>-ev</td>
</tr>
<tr>
<td>stop</td>
<td>infinitive(...)</td>
<td>amp(AV, I1, I2, Ops)</td>
<td>end</td>
<td>*prom</td>
<td>*ev</td>
</tr>
<tr>
<td>cease</td>
<td>participle(...)</td>
<td>imperfective</td>
<td>imperfective</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

‘Struct’ in the ‘vb/3’ clause is the event structure from verbsem/5 (see page 148) and ‘Events’ the corresponding list of situation type primitives. The attributes lists (As) are discussed on page 157 and the relations (Rel) in section A.5.

1 Section 6A.6.
2 In collect_expr/6 and as input to collect_evs/8.
3 Output of collect_evs/8 and collect/6.
4 Full list in the table on page 92.
5 Lexical operators from verbsem/5. Note that ‘Ops’ is a list of these.
6 These are listed in the table on page 95.
The algorithms are shown first in pseudo-code, which is simplified and does not include things like error checking, and then in Prolog on page 160. Variable names are capitalised, as in Prolog, and as far as possible the same names are used here and in the real Prolog code. The organisation differs somewhat but the Prolog procedure names are printed, to the right, to indicate corresponding sections of the pseudo-code.

\[\text{COLLECT}(\text{LF})\]  
\[\begin{align*} \text{begin} & \quad (\text{collect/6}) \\
& \quad \text{Evs, Es, Rs, Vs} \leftarrow \text{empty lists} \\
& \quad \text{repeat} \quad (\text{collect_expr/6}) \\
& \quad \quad \text{if two connected situations found in LF then} \\
& \quad \quad \quad \text{Evs} \leftarrow \text{Evs + asp/5} \\
& \quad \quad \text{else if one situation found in LF then} \\
& \quad \quad \quad \text{Evs} \leftarrow \text{Evs + vb/3} \\
& \quad \quad \text{else if an entity found in LF then} \\
& \quad \quad \quad \quad \text{if ENTITY(entity data,LF) succeeds then} \\
& \quad \quad \quad \quad \quad \text{Es} \leftarrow \text{Es + Entity (de/4)} \\
& \quad \quad \quad \quad \quad \text{Vs} \leftarrow \text{Vs + variables found in the entity} \\
& \quad \quad \text{end if} \\
& \quad \quad \text{else if a relation found in LF then} \\
& \quad \quad \quad \text{Rs} \leftarrow \text{Rs + rel/3} \\
& \quad \quad \quad \text{Vs} \leftarrow \text{Vs + variables found in the relation} \\
& \quad \quad \text{end if} \\
& \quad \text{LF} \leftarrow \text{LF - the thing that was found} \\
& \quad \text{until nothing found} \\
& \quad \text{NewEvs, NewEs, NewImpEs} \leftarrow \text{empty lists} \\
& \quad \text{while Evs not empty do} \quad (\text{collect_evs/8}) \\
& \quad \quad \text{Struct} \leftarrow \text{event structure of head of Evs} \\
& \quad \quad \text{Evs} \leftarrow \text{tail of Evs} \\
& \quad \quad \text{Primitives, Vars} \leftarrow \text{COLLECT_VARIABLES(Struct)} \\
& \quad \quad \text{NewEs, NewImpEs} \leftarrow \text{NewEs, NewImpEs + CHECK_VARIABLES(Vars, Es)} \\
& \quad \quad \text{NewEvs} \leftarrow \text{NewEvs + vb/3, including Primitives} \\
& \quad \text{end while} \\
& \quad \text{for each V in Vs do} \quad (\text{linked_entities/5}) \\
& \quad \quad \text{if V \in Es but V \notin NewEvs or NewEs then} \\
& \quad \quad \quad \text{NewEs} \leftarrow \text{NewEs + V's entity} \\
& \quad \quad \text{end if} \\
& \quad \text{end for} \\
& \quad \text{Ps} \leftarrow \text{pronouns from NewEs} \quad (\text{pronouns/3}) \\
& \quad \text{NonPs} \leftarrow \text{non-pronouns from NewEs} \\
& \quad \text{return Evs, Ps, NonPs, NewImpEs, Rs} \\
& \end{align*}\]

There are three loops here. The first picks out the major elements of the logistic form: eventualities, entities, and relations. The second loop converts the eventualities into primitives, and the third checks that all variables that occur in entity attributes and relations in the logistic form (Vs) also refer to an entity or eventuality that is returned.

\[1\text{i.e. aspectual verb and main verb group.}\]
The representation of entities in the logicistic form (see the table on page 146) is spread out over several expressions, united by a common variable. Between these and the attributes used in the semantic representation there is a straightforward mapping.

<table>
<thead>
<tr>
<th>Logicistic form</th>
<th>Semantic representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ref ( (X_1, \ldots, X_n) )</td>
<td>( X )</td>
</tr>
<tr>
<td>is_a ( (X, \text{Concept}) )</td>
<td>is_a = Concept</td>
</tr>
<tr>
<td>name ( (X, \text{Atom}) )</td>
<td>name = Atom</td>
</tr>
<tr>
<td>poss ( (X, Y) )</td>
<td>poss = Y</td>
</tr>
<tr>
<td>poss ( (Y, X) )</td>
<td>poss_by = Y</td>
</tr>
<tr>
<td>prop ( (X, \text{Attr}, \text{Val}) )</td>
<td>Attr = Val</td>
</tr>
<tr>
<td>quant ( (\ldots, X) )</td>
<td>—</td>
</tr>
</tbody>
</table>

The conversion procedure takes all the remaining

The conversion procedure takes all the remaining expressions in the logicistic form of the sentence and converts them according to the table above.

ENTITY \( (X, \text{Pron}, \text{Ev}, \text{LF}) \) (entity/6)
begin
Variables, Attributes \( \leftarrow \) empty lists
for each expression in LF do (expressions/6)
if the expression is ‘is_a\( (X, \text{Concept}) \)’ then
if Concept is defined then
Attributes \( \leftarrow \) Attributes + ‘is_a = \text{Concept}’
end if
else if the expression contains \( X \) then
Attributes \( \leftarrow \) Attributes + semantic representation as per the table
Variables \( \leftarrow \) Variables + any variables except \( X \) in the expression
end if
end for
End \( \leftarrow \) de \( (X, \text{Pron}, \text{Ev}, \text{Attributes}) \)
return Entity, Variables
end

As an example of how the procedure works, the following call occurs during the analysis of ‘Philip’s shabby room was convenient.’

\[
\begin{align*}
\text{E} &= \text{de}(X_1, \text{'pron'}, \text{'-ev'}, \text{[poss=X2,name='Philip',is_a=male]}, \text{prop}(X_2, \text{appearance}, \text{shabby}), \text{ref}(X_2, \text{'-pron'}, \text{'-ev'}, \text{'-ev'}), \text{is_a}(X_2, \text{room}), \text{Vs}, \text{E}).
\end{align*}
\]

Since ‘X2,’ the room, is linked to Philip (X1), via the ‘poss’ attribute, it is included in the list of variables. Only those expressions in the logicistic form that mention Philip have been converted and included in the entity.

8That is, all except those sit’s, ref’s, and temp ref’s that have been removed.
The procedure \texttt{collect\_variables/6} converts the event structure (see page 148) of a single situation to situation type primitives. Data about all the variables in the eventualities are returned and used later for creating `implied entities'.

\begin{verbatim}
COLLECT\_VARIABLES(Struct) (collect\_variables/6)
begin
Primitives, Vars ← empty lists
for each expression X in Struct do
    Concept ← the functor of X (collect\_spec/4)
    Xs ← the rest of X (attributes)
    As ← attributes from the definition of Concept
    ST ← value of the attribute `sit\_type' in As
    As, Vars ← \texttt{COMBINE\_ATTRIBUTES(Xs,As)}
    Types ← empty list
    for each expression Y in ST do (collect\_sit\_type/5)
        Vars ← Vars + variables in Y
        Types ← Types + Y
    end for
    VarsAs ← attributes in As that refer to variables
    for each T in Types do
        Primitives ← Primitives + ev/3 (Concept, VarsAs)
    end for
end for
return Primitives, Vars
end
\end{verbatim}

Combining the two attribute lists, from the lexicon (\texttt{verbsem/5}) and the definition (\texttt{concept/2}), for a single eventuality is fairly straightforward. All attribute names that occur in the lexicon specification should also be in the definition, and there should be no attribute `sit\_type' in either list.

\begin{verbatim}
COMBINE\_ATTRIBUTES(Xs,As) (combine\_attributes/6)
begin
Attr, Vars ← empty lists
for each tuple X in lexicon specification (Xs) do
    find the tuple Y with same attribute name in concept definition (As)
    Attr ← Attrs + (X \cup Y) (combine\_valspec/8)
    Vars ← Vars + variables in the tuples
end for
for each remaining tuple A in As do (combine\_attributes/5)
    Attr ← Attrs + A (combine\_valspec/8)
    Vars ← Vars + any variable in A
end for
return Attrs, Vars
end
\end{verbatim}

All variables that occur in the attributes are returned separately, and the Prolog code also contains error checking.
Data about all variables in the event structure of a situation is returned by `collect:variables/6`, in the following format.

<table>
<thead>
<tr>
<th>Variable data (Vars)</th>
<th>Syn-data (Num, Concepts, Attributes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syn</td>
<td>The variable in question</td>
</tr>
<tr>
<td>Num</td>
<td>Number of times it occurs</td>
</tr>
<tr>
<td>Concepts</td>
<td>All concepts that the variable is declared (i.e. an instance of</td>
</tr>
<tr>
<td>Attributes</td>
<td>All attributes (names) that the variable is the value of</td>
</tr>
</tbody>
</table>

From the above information, plus a list of all entities in the logistic form (Es), the procedure below finds those entities that occur explicitly (N), and creates representations for those that do not (NI).

```
CHECK_VARIABLES(Vars,Es) (check:variables/6)
begin
  N, NI ← empty lists
  for each variable pair in Vars do
    V0 ← variable in current pair
    D ← data in current pair
    if Es contains entity with same variable as V0 then
      if that entity is compatible with D then (compatible-entity/3)
        N ← N + entity
      end if
    else
      if V0 occurs more than once then (new-entity/6)
        create new entity from V0 & D
      NI ← NI + new entity
      end if
    end if
  end for
  return N, NI
end
```

To see how the transformation into primitives works in practice, we consider the sentence 'He had written to Mrs. Otter.' The verb, 'write,' has the following definition, for this particular sense and sub-categorisation.

```
verbsem(write,...,[np(...,common(subj)....,X...),pp(to,...,Y...)].
  creation(agent=X,object=Z:letter)
  + transfer(source=X,object=Z,target=Y).[]).
```

Two eventualities, 'creation' and 'transfer,' are used. The first has this form.

```
concept(creation,
  [is_a = eventuality,
   object = X,
   agent = Y:person,
   sit_type = activity + transition( "X, (X & (X<location = Y<location)) ")].
```

Note that the variable names do not match here. In the lexicon entry, 'X' is the value of the 'agent' attribute; in the definition it is the value of 'object.' Prolog's built-in unification handles this transparently.
The second eventuality, ‘transfer,’ has the following definition.

```
concept(transfer,  
    [is_a = eventuality,  
      object = X,  
      source = Y:person,  
      target = Z:person,  
      sit_type = transition( Y'owns = X, Z'owns = X )]).
```

In the value of the ‘sit_type’ attributes above are three primitive situation types, an activity and two transitions. These, with attributes from the concept definitions added, are the final eventualities.

```
| ?- s(_, LF, [De, had, written, to, Mrs Utter', []]), collect(LF, Evs, Ps, NonPs, ImpEs, Rs).

LF = [ref(X1, 'pron', 'vb', 'ev'), is_a(X1, male),  
      sit(X2, finite([past, perfect]), creation(agent=X1, object=X3:letter)),  
      ref(X4, 'pron', 'ev'), name(X4, 'Mrs Utter'), is_a(X4, female)],

Evs = [vb(X2).vb(finite([past, perfect]), []),  
       ev(_, activity, [concept=creation, agent=X1, object=X3]),  
       ev(_, transition(X3, X3e(X3'location=X1'location))),  
       ev(_, transition(X1'owns=X3, X4'owns=X3),  
           [concept=creation, source=X1, object=X3, target=X4])],

Ps = [de(X1, 'ev', [is_a=male])],  
NonPs = [de(X4, 'ev', ['Mrs Utter', is_a=female])],
ImpEs = [de(X3, 'ev', [is_a=letter])],
Rs = [] ?
```

And here follows the Prolog code that implements the semantic interpretation.

```
collect.pl
```

9See the table on page 95.
TEARS IN THE RAIN

161

collect_expr (Isi, Evs, Es, Rs, Vs, Exprs) 

; select (temp_rel (Vi, V2, Rel), Is0, Isi) ->
( check_temp_rel (Rel, NewRel, RelVs) ->
  append (Fs, RelVs, NewRel);
  NewVs = [Rel (NewRel, V1, V2) | Rs]

; format ('collect : unknown temp rel : u \u201cm', [Rel])
; NewVs = Vs;
  NewVs = Rs );
collect_expr (Isi, Evs, Es, NewVs, NewEvs, Exprs)

; collect_expr (Evs, Es, [], NewEvs, [], NewEs0, [], New IMPLIED Es),
linked_entities (Vs, NewEvs, Es, NewEs0, NewEvs),
list_to_ord_set (Es, OldSet),
list_to_ord_set (NewEvs, NewSet),
ord_subtract (OldSet, NewSet, Set),
\$error_collect (Set),
append (NewEvs, Set, NewEvs),
reverse (Rs, RevRs),
{ ( Is0 = [] ) -> true 
\$ format ('collect : unused expressions : \u201cu \u201cm', [Is0]) })
;
propositions (NewEvs, Prons, NonProns),
Exprs = exprs (NewEvs, Prons, NonProns, New IMPLIED Es, RevRs) .

error_collect ([de(Y, P, Ev, As) | DeEs]) :=
\$ format ('collect : entity u \u201cm \u201cm \u201cm \u201cm connected \u201cm', [Y, P, Ev, As]),
error_collect (DeEs).

error_collect ([]).

pronouns ([de (X, \u2018pros \u2018Ev, As) | Es], [de (X, Ev, As) | Ps], Rs) :=
pronouns (Es, Ps, Rs).
pronouns ([de (X, \u2018pros \u2018Ev, As) | Es], [de (X, Ev, As) | Ps]) :=
pronouns (Es, Ps).

pronouns ([], [], []). 

linked_entities ([VO | Vs], Es, NewEs0, RevRs) :=
( member (rb (V, W, Z), Vs), Y \u2014 VO ->
  NewEvs = RevEvs
; member (de (V, W, Z), Vs), Y \u2014 \u2014 VO ->
  NewEvs = RevEvs

; member (de (Y, P, Ev, As), Es), Y \u2014 VO ->
\$ format ('collect : connected entity : u \u201cm', [VO, As]),
append (NewEvs, de (Y, P, Ev, As) | NewEvs)

; format ('collect : undefined reference : u \u201cm', [VO]),
NewEvs = RevEvs
)

linked_entities ([], Vs | Es, NewEvs, NewEvs).
linked_entities ([], [], [], []). 

check_temp_rel (where, r_eq, []). 

check_temp_rel (before, r_before, []).

check_temp_rel (after, r_after, []). 

check_temp_rel (since, r_under, []). 

check_temp_rel (until, r_under, []). 

check_temp_rel (before (X), r_before (X), []). 

check_temp_rel (after (X), r_after (X), []). 

check_temp_rel (on, r_eq, []). 

check_temp_rel (at, r_eq, []). 

check_temp_rel (in, r_under, []). 

check_temp_rel (for, r_under, []). 

check_temp_rel (relative, r_relative, []). 

entity (XO, Pron, Ev, Ps, Is, Entity) :=
\$ format ('create entity : u \u201cm', [XO, Ps]),
expressions (Ps, XO, [], Is, [], As).
Entity = de (XO, Pron, Ev, As).

expressions ([P | Ps], YO, XO, Is, ASo, As) :=
( P = is_a (Y, Concept), Y \u2014 YO ->
  ( concept (Concept) ) ->
  As = [is_a (Concept) | ASo]

; format ('entity: u \u201cm \u201cm concept \u201cm', [Concept]),
As = ASo ).

Isi = XSo 
; P = name (V, Name), Y \u2014 YO ->
  Isi = XSo .
Axi = [name=Name[Aso]
; P = prop(Y, Attr, Val), Y == Y0 ->
( var(Val) -> Axi = [Val][Aso]; Axi = Aso ),
Axi = [Attr,Val][Aso]
; P = poss(X, Y), Y == Y0 ->
Xsi = [X][Aso],
Axi = [pos, byx][Aso]
; P = poss(X, Y), Y == Y0 ->
Xsi = [X][Aso],
Axi = [pos, ax][Aso]
; Xsi = Xso,
Axi = Aso ),
expressions(Expr, Y0, X1, X2, Axi, As).
expressions([..], Xs, Xs, Exprs, As) :- reverse(Exprs, As).
collect_exprs(Exprs, Exprs0, Exprs1, Exprs2, Exprs3, Exprs4) :-
( X = vb(Y, I, Spec) -> E = vb(Y, vb((I, Ops), Eventualities))
; X = asp(Y, AV, I1, I2, Eventualities) -> E = vb(Y, asp(Y, AV, I1, I2, Ops), Eventualities)
; format(1, collect_exprs(Exprs, Exprs0, Exprs1, Exprs2, Exprs3, Exprs4)),
collect_variables(Spec, Eventualities, [], NewOps, [], Vars),
reverse(Exprs, Ops),
check_variables(Exprs, Exprs0, Exprs1, Exprs2, Exprs3, Exprs4),
collect_exprs(Exprs0, Exprs1, Exprs2, Exprs3, Exprs4).
collect_variables(Spec, Exprs, Ops0, Ops, Exprs0, Exprs) :-
( Spec = Ss + E ->
collect_variables(Ss, Exprs0, Ops0, Ops, Exprs0, Exprs1),
collect_spec(Ss, Ys, Vars1, Vars),
append(Xs, Ys, Exprs)
; Spec = .. [AspOp, X], asp_op(AspOp) ->
collect_variables(X, Exprs0, Ops, Exprs, Ops, Vars0, Vars),
collect_spec(Spec, Exprs, Vars0, Vars), Ops = Ops0 ).
asp_op(start).
asp_op(end).
asp_op(imperfect).
collect_spec(X, Exprs, Vars0, Vars) :-
X = .. [Concept[Xs]],
( concept(Concept, As0) ->
( select(sit_type=SitType, As0, Axi) ->
combine_attributes(Xs, Asi, [], Axi, Vars0, Vars1),
collect_sit_type(SitType, [], Types, Vars1, Vars),
combine_attributes(Xs, Axi, [], TempVars),
variable_vars(TempVars, Vs),
filter_atts(As, Vs, [], FinalAs),
map_exprs(Types, Concept, FinalAs, Exprs)
; format(1, collect_spec(Exprs0, [sit_type], [ definitio ns], [,,, ], [Concept]),
Exprs = [], Vars = Vars0 )
; format(1, collect_spec(Exprs0, [sit_type], [ definitio ns], [,,, ], [Concept]),
Exprs = [], Vars = Vars0 )
)
map_exprs([Tp|Tps], Concept, As, [Ev|Evrs]) :-
Expr = expr(_, ..., [concept(Concept|As)]),
map_exprs(Tps, Concept, As, Evs).
map_exprs([,, ...]),
filter_atts(Xs, Vs, As0, As) :-
( X = (Asy) ->
Axi = As0
; var(Y) ->
( member(Z, Vs), Z == Y ->
Axi = [Z][As0]
; Axi = As0
; A => restriction -> I Y Y
Axi = As0
; format(1, filter_atts(,,, illegal value : w-n], [Y]),
Axi = As0
; format(1, filter_atts(,,, illegal value : w-n], [Y]),
Axi = As0
)
filter_atts(As, Vs, Asi, As).

Tears in the Rain
162
filter_attributes([], As, RevAs) :- reverse(As, RevAs).

combine_attributes([X|Xs], As0, Attrs0, Attrs, Vars0, Vars) :-
  ( X = Vars0, var(Var) ->
    write('combine_attributes::attribute_values_variable'), nl,
    Asi = As0, Attrsi = Attrs0, Varsi = Vars0
  ; valspec(X, Name, V1i, Vri, Cni) ->
    ( select(Var, As0, Asi), valspec(Y, Name, Y12, Yr2, Cn2) ->
      append(V1i, V12, Vli), append(Vri, Yr2, Yri), append(Cni, Cn2, Cni)
    ; format('combine_attributes::-[x]type=unknown, X'), [Name]).
    Asi = As0, V1i = V1i, Vri = Vri, Cni = Cni
  ),
  combine_valspec(Name, V1, Vr, Cn, Attrs0, Attrs0, Vars0, Vars1),
  X = (sit_type=.) ->
  write('combine_attributes::[sit_type]=unknown'), nl,
  Asi = As0, Attrsi = Attrs0, Varsi = Vars0
  ; (restriction=)
    write('combine_attributes::[restriction]=unknown'), nl,
    Asi = As0, Attrsi = Attrs0, Varsi = Vars0
  ; format('combine_attributes::unknown X', [X]),
  combine_attributes(As, Attrs, Attrs1, Vars, Vars),
  combine_attributes([], As0, Attrs0, Attrs0, Vars0, Vars1).

combine_attributes([A|As], Attrs0, Attrs, Vars0, Vars) :-
  ( valspec(A, Name, V1, Vr, Cn) ->
    combine_valspec(Name, V1, Vr, Cn, Attrs0, Attrs0, Vars0, Vars1),
    A = (sit_type=.) ->
    write('combine_attributes::[sit_type]=unknown'), nl,
    Attrsi = [A|Attrsi], Varsi = Vars0
  ; (restriction=)
    write('combine_attributes::[restriction]=unknown'), nl,
    Attrsi = [A|Attrsi], Varsi = Vars0
  ; format('combine_attributes::unknown X', [A]),
    combine_attributes(As, Attrs, Attrs1, Vars, Vars),
    reverse(Attrsi, RevAttrs)
  ),

combine_valspec(Name, V1, Vr, Cn, Attrs0, Attrs, Vars0, Vars) :-
  ( V1 = [] ->
    V1 = [X] ->
      check_value(V1, Cn, Name, X, Attrs0, Attrs),
      check_value(V1, Name, Value, Attrs0, Attrs)
  ; format('combine_valspec::more than two values: X', [V1]),
    Attrs = Attrs0, Vars = Vars0
  ),

check_value(Vr, Cn, Name, Value, Attrs0, Attrs) :-
  ( Vr = [], Cn = [] ->
    Attrs = [Name=Value|Attrs0]
  ; format('combine_valspec::both variables X and variables/ concepts X: Y', [Value, Vr, Cn]),
    Attrs = Attrs0 )

unify_vars([], Y) :-
  unify_vars(Names, Y).

unify_vars([X|Xs], Y) :-
  unify_vars(Names, Y),
  unify_vars(Xs, Y).

valspec(A, Attr, [], [], []) :-
  atom(A),
  A = Attr.
  attr(Attr).
val spec(Attr=Value, Attr.[Value].[[],[[]]] :-
  attr(Attr),
  aton(Value).
val spec(Attr=Var, Attr.[Var], [[],[[]]] :-
  attr(Attr),
  var(Var).
val spec(Attr : Concept, Attr.[Concept], [[],[[Concept]]] :-
  attr(Attr),
  aton(Concept).
val spec(Attr=Var:Concept, Attr.[Var,:Concept], [[],[[Concept]]] :-
  attr(Attr),
  var(Var),
  aton(Concept).
attr(X) :-
  aton(X), \ ( X = "sit_type" ; X = "restriction" ).
collect sit_type(SitType, Ts0, Ts, Yars0, Yars) :-
( SitType = Xs + X ->
collect one sit_type(Xs, Yars0, Yars1),
collect sit_type(SitType[Ts0], Ts, Yars1, Yars)
; collect one sit_type(SitType, Yars0, Yars),
  Ts = [SitType[Ts0] ] ).
collect one sit_type(Spec, Yars0, Yars) :-
  ( aton(Spec) ->
    Yars = Yars0
  ; Spec = ..[Si,S2] ->
    check statement(Si,Yars0, Yars1),
    check statement(S2, Yars1, Yars)
  ; check condition(Si,Yars0, Yars) ).
check restriction(S, Yars0, Yars) :-
  ( S = C & Cs ->
    check condition(C, Yars0, Yars1),
    check restriction(Cs, Yars1, Yars)
  ; check condition(S, Yars0, Yars) ).
check condition(X, Yars0, Yars) :-
  ( X = ..[X1,X2] ->
    check attr spec(X1, Yars0, Yars1),
    check attr spec(X2, Yars1, Yars)
  ; check condition(X, Yars0, Yars) ).
check statement(S, Yars0, Yars) :-
  ( var(S) ->
    Yars = Yars0
  ; S = (AS1/V), var(V) ->
    check attr spec(AS1, Yars0, Yars1),
    add variable(Yars1, Y, Yars)
  ; S = (AS1=AS2) ->
    check attr spec(AS1, Yars0, Yars1),
    check attr spec(AS2, Yars1, Yars)
  ; S = (X) ->
    check statement(X, Yars0, Yars)
  ; S = ([X1&X2] ->
    check statement(X1, Yars0, Yars1),
    check statement(X2, Yars1, Yars)
  ; format('check condition :illegal statement :\"w-\"s', [S]),
    Yars = Yars0 ).
check attr spec(AS, Yars0, Yars) :-
  ( AS = (Y\^A), var(Y) ->
    add variable attribute(Yars0, Y, Y, Yars)
  ; format('check attr spec :illegal attr spec :\"w-\"s', [AS]),
    Yars = Yars0 ).
add variable(Yars0, Y0, Yars) :-
( select(Y-data([0], Cn, Ass), Yars0, Yars1), Y = Y0 ->
  Y1 = Y0 + 1,
  Yars = [Y0-data([Y1,Cn, Ass])|Yars1],
  Yars = [Y0-data([1,1,1])|Yars0] ).
add_variable_concepts(Yvars0, V0, Cns, Vars) :-
  list_to_ord_set(Cns, CnsSet),
  ( select(V-data(0, Cns, As), Vvars0, Vars1), V == V0 ->
    N1 is N0 + 1,
    ord_union(Cns0, CnsSet, NewCns)
  ; N1 = 1, NewCns = CnsSet, As = [], Vars1 = Vars0 ),
  Vars = [V-data(N1, NewCns, As)|Vars1].

del_variable_attribute(Yvars0, Y0, A, Yvars) :-
  ( select(V-data(0, Cns, As), Yvars0, Yvars1), V == Y0 ->
    N1 is N0 + 1,
    ord_add_element(As, A, Nevars)
  ; N1 = 1, Cns = [], Nevars == [A], Yvars1 = Yvars0 ),
  Yvars = [V-data(N1, Cns, Nevars)|Yvars1].

check_variables([V-data(Sum, Cns, VAs)|Vars], Es, N0, N10, N1) :-
  ( member(De(V, P, E, Es), Es), V == Y0 ->
    ( compatible_entity(Eas, Cns, VAs) ->
      ord_add_element(S0, de(V, P, E, Es), N1),
      N1 = N10
    ; format('check_variables: bad entity: \wedge E \wedge E \wedge E \wedge E \wedge E \wedge E', [Y0, Eas, Cns, VAs]),
      N1 = N0, N10 = N10
    ),
    check_variables(Vars, Es, N1, N10, N1),
    check_variables([], Es, N1, N10, N1).)

variable_vars([Y-V Yvars], [Y Yvars]) :-
  variable_vars(Vars, Vars). variable_vars([], []).

compatible_entity([], [], []).% 
compatible_entity([Eas, Cns, VAs]) :-%
  format('compatible_entity: \wedge E \wedge E \wedge E \wedge E \wedge E \wedge E', [Eas, Cns, VAs]).

new_entity(V, Sum, Cns, As, N10, N1) :-
  format('new_entity: \wedge E \wedge E \wedge E \wedge E \wedge E \wedge E', [Sum, Cns, As]),
  ( Sum > 1 ->
    new_isa(Cns, Nevars1),
    new_atts(As, Cns, [I, Nevars2]),
    append(Nevars1, Nevars2, Nevars),
    E == de(V, \wedge E, Nevars),
    ord_add_element(S0, E, N1),
    N1 = N10
  ;
    new_isa([Cns | Cns], [is_a=Cns | As]) :-
    new_isa(Cns, As),
    new_isa([], []).

new_atts([A | As], Cns, As0, As) :-
  ( attribute(A, Cns) ->
    As1 is As0
  ; format('new_entity: \wedge E \wedge E \wedge E \wedge E \wedge E \wedge E', [A, Cns]),
    As1 is As0
  ),
  new_atts(A, Cns, As1, As),
  new_atts([], [], As1, As1).

new_atts([I | Nevars], As) :-
  reverse(Nevars, As),
  attribute(Spec, Cns) :-
    member(Cns, Cns),
    concept(As, As),
    attr_member(Spec, As),
  attr_member(At, As) :-
    member(At, As),
  attr_member(At, As) :-
    member(At, As),
  attr_member(Spec, As) :-
    member(is_a=Spec, As),
    concept(Nevars, Nevars),
    attr_member(Spec, Nevars).
The analysis of the sentence ‘He arrived in Paris before tea’ throws light on some more obscure details of the interpretation. The verb has this definition.

\[
\text{verbsem} \text{arrive,}_{\ldots} \text{np} \text{\{\ldots common (mbj),\ldots X,\ldots \}}, \text{pp (in, \ldots Y, \ldots)}],
\text{end (movement (agent=X, object=Y, target=Y)),} [\text{[\ldots]}].
\]

And the eventuality concept ‘movement’ the following.

\[
\text{concept (movement,}
\begin{align*}
\text{[is_a = eventuality,} \\
\text{agent : person,} \\
\text{object = X:moveable,} \\
\text{source = Y:location,} \\
\text{target = Z:location,} \\
\text{via : location,} \\
\text{sit type = transition (X:location = Y, X:location = Y),} \\
\text{+ activity} \\
\text{+ transition (X:location = Z, X:location = Z),} \\
\text{restriction = (X:moveable.size <= Y:location.size) \\
& (X:moveable.size <= Z:location.size))].}
\end{align*}
\]

The attribute ‘restriction’ is ignored since the comparison operators have not been implemented. Here follows the output of collect/6 for this example.

\[
| ?- s(_.LF, [de, arrived, in, 'Paris', before, tea], []).collect(Sem, Evs, Ps, NonPs, ImpPs, Rs).
\]

\[
\begin{align*}
\text{LF} &= \text{[ref(X1, 'prom', '-vb', '-ev'), is_a(X1, male),} \\
& \text{sit(X2, finite([past])), end (movement (agent=X1, object=X1, target=X5))),} \\
& \text{ref(X5, '-prom', '-vb', name(X5, 'Paris'), is_a(X5, city),} \\
& \text{temp_rel(X2, X3, before),} \\
& \text{ref(X3, '-prom', '-ev'), is_a(X3, tea)]},
\end{align*}
\]

\[
\begin{align*}
\text{Evs} &= \text{[vb(X2, vb(finite([past])), [end]),} \\
& \text{ev(_transition(X1:location=X4, X1:location=X4),} \\
& \text{[concept (movement, agent=X1, object=X1, target=X5), source=X4)],} \\
& \text{ev(_activity, [concept (movement, agent=X1, object=X1, target=X5), source=X4)],} \\
& \text{ev(_transition (X1:location=X5, X1:location=X5),} \\
& \text{[concept (movement, agent=X1, object=X1, target=X5, source=X4)]])},
\end{align*}
\]

\[
\begin{align*}
\text{Ps} &= \text{[de(X1, '-ev', [is_a=male])],} \\
\text{NonPs} &= \text{[de(X5, '-ev', [name='Paris', is_a=city]),} \\
& \text{de(X3, '-ev', [is_a=tea])],} \\
\text{ImpPs} &= \text{[de(X4, '-ev', [is_a=location])],} \\
\text{Rs} &= \text{[rel(r, before, X2, X3)]} ?
\end{align*}
\]

\[
\text{yes}
\]

| ?- |

Of particular interest here is the lexical aspectual operator ‘end’ in the eventuality clause, that the ‘via’ attribute has not been included, because there is no variable connected with it, and that the evocative noun phrase ‘tea’ is still considered an entity: it has not yet been ‘coerced’ into an eventuality (section 6.4.6).
A.5 Segmentation

When a sentence contains more than one verb phrase, there are, from a temporal point of view, three cases (section 7.3).

- First, a non-finite phrase can be complement to the main (finite) verb phrase. In this type of construction the tense of the non-finite phrase is relative to the governing verb phrase (see page 16).
- The second case is when there is a temporal conjunction or preposition indicating the relationship between the components (see the table on page 110).
- Finally, two syntactically linked verb phrases can be temporally unrelated, in which case they can be paraphrased by two separate sentences.

The two, or more, temporally separate phrases in the last case above are said to be in different segments. Given the temporal relations (Rs) and eventualities (Vbs) from one sentence, the procedure segments/5 computes these segments (Segs), which means finding transitive closures over the 'real' temporal relations. Not all of the Rs's belong to this group, however, since some of them might be adjuncts. So the processing is divided into two stages: maprels/8 separates the relations from the adjuncts, and find_segs/5 computes the segments.

```
SENGMENTS(Rs,Vbs,NonPs) (segments/5)
begin
  Rs, Adj, Es ← MAP_RELS(Rs,Vbs,NonPs) (maprels/9)
  Vbs, Pairs ← MAP_VBS(Vbs,Adj) (map_vbs/4)
  Segs ← FIND_SEGS(Vbs,Pairs,Rels) (find_segs/5)
  return Segs, Es
end
```

The input parameter 'NonPs' contains the common nouns in the sentence, in order to 'coerce' them into eventualities, if needed. Those that escape this fate are returned in 'Es.'

Mapping relations

Those temporal relations that have been implemented\(^1\) are listed below. Note that conjunctions and prepositions are formalised in the same way.

<table>
<thead>
<tr>
<th>Type</th>
<th>Linguistic form</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>r_eq</td>
<td>when, while, on, at</td>
<td>Same time</td>
</tr>
<tr>
<td>r_relative</td>
<td>Sub-ordinate phrase</td>
<td>Relative to governing phrase</td>
</tr>
<tr>
<td>r_before</td>
<td>before</td>
<td>First phrase, then second</td>
</tr>
<tr>
<td>r_after</td>
<td>after</td>
<td>Second phrase, then first</td>
</tr>
</tbody>
</table>

Not all possible combinations of eventualities, coerced entities and relation types are valid. The following table lists the possibilities (compare the table on page 110).

\(^1\)See check_temp_re1/3, section A.4.
Eventuality  | Eventuality  | Relation | Any type
---|---|---|---
Eventuality  | Coercible entity  | Relation | Not r-relative
Eventuality  | Non-coercible entity  | Adjunct | Not r-relative
Coercible entity  | Eventuality  | Relation | Any type
Coercible entity  | Coercible entity  | Relation | Not r-relative
Coercible entity  | Non-coercible entity  | Adjunct | Not r-relative
Non-coercible entity  | Eventuality  | — | —
Non-coercible entity  | Coercible entity  | — | —
Non-coercible entity  | Non-coercible entity  | — | —

And the algorithm below implements the classification of the temporal relations.

```plaintext
MAP_RELS(Rs,Vbs,Es) (map_rels/9)
begin
  Rel, Adjs ← empty lists
  for each relation R in Rs do
    X1 ← left-hand side of R
    X2 ← right-hand side of R
    if Vbs contains an eventuality X1 then
      if Vbs contains an eventuality X2 then (check_rhs/11)
        Rel ← Rel + relation(X1,X2)
      else
        E2 ← entity X2 from Es
        if E2 can be coerced into an eventuality then (coerce_entity/4)
          Vbs ← Vbs + E2
          Rel ← Rel + relation(X1,E2)
        else
          Adjs ← Adjs + adjunct(X1,E2)
      end if
    end if
  else
    E1 ← entity X1 from Es
    coerce E1 into an eventuality (coerce_entity/4)
    Vbs ← Vbs + E1
    if Vbs contains an eventuality X2 then (check_rhs/11)
      Rel ← Rel + relation(E1,X2)
    else
      E2 ← entity X2 from Es
      if E2 can be coerced into an eventuality then (coerce_entity/4)
        Vbs ← Vbs + E2
        Rel ← Rel + relation(E1,E2)
      else
        Adjs ← Adjs + adjunct(E1,E2)
      end if
    end if
  end if
end for
return Rel, Adjs, Es
end
```

Entities that are coerced into eventualities are also removed from the list 'Es.'
Mapping propositions

The information expressed by the adjuncts (Adj s) is not used further in the current implementation, but it seems courteous to include it in the output anyway. The procedure map_vbs/4 adds the adjuncts, as attributes, to the eventualities and it also converts these (Vbs) into the format find_segs/3 needs (Pairs).

```
MAP_VBS(Vbs,Adj s) (map_vbs/4)
begin
  Vars, Pairs ← empty lists
  for each eventuality in Vbs do
    V, Asp, Evs ← data from eventuality
    xi ← eventuality in an 'adjunct' relation\(^2\) (ev_adjuncts/4)
    if xi = V then
      R, x2 ← type and entity\(^3\) from the same relation
      for each Ev in Evs do
        add 'R=x2' to attributes of Ev\(^4\)
      end for
    end if
  end for
  Vars ← Vars + V
  Pairs ← Pairs + \{ V, Asp, Evs \}
end for
return Vars, Pairs
end
```

The variable ‘Vars’ contains the variables only, in the same order as ‘Pairs.’

Meaning of temporal relations

The meaning of the relations, which are listed in table on page 167, has been formalised in terms of the time points used to describe the external meaning (section 7.1). The procedure relation/10 implements this table.

<table>
<thead>
<tr>
<th>Type</th>
<th>Times 'Left' ev.</th>
<th>Times 'Right' ev.</th>
<th>Clauses(^5)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Base</td>
<td>Facts</td>
<td>Event</td>
</tr>
<tr>
<td>r_relative</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>r_eq</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>r_before</td>
<td>B</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>r_before(X)</td>
<td>B</td>
<td>E1</td>
<td>B</td>
</tr>
<tr>
<td>r_after</td>
<td>B</td>
<td>E1</td>
<td>B</td>
</tr>
<tr>
<td>r_after(X)</td>
<td>B</td>
<td>E1</td>
<td>B</td>
</tr>
</tbody>
</table>

Note that these meanings only apply when the relations hold between two eventualities. As adjuncts they have no meaning. The ‘X’ in the table is an entity expressing a duration, as in ‘after a moment,’ and it is not interpreted further.

\(^2\)Only one adjunct per eventuality is supported. Extending this would be trivial but, even so, probably not worth the trouble.

\(^3\)This is typically a time related noun phrase such as ‘the next day.’

\(^4\)Since this information is not used further there is no point in doing anything more sensible.

\(^5\)See section 6.6.
Restrictions on the eventualities

As discussed in section 7.3, not all possible combinations of tenses are acceptable in a temporal relation. With eleven possible external aspects, there are more than a hundred combinations, but only a fraction seem well-formed. The following table is a rough approximation of the valid ones without perfect aspect.

Philip ceases after he settles.  
Philip settles before he ceases.  
? Philip ceases after he had settled.  
? Philip had settled before he ceases.  
Philip ceases after he settled.  
Philip settled before he ceases.  
Philip ceased after he settled.  
Philip settled before he ceased.  
Philip ceased after he had settled.  
Philip had settled before he ceased.  
? Philip will cease after he will settle.  
? Philip will settle before he will cease.  
Philip will cease after he will have settled.  
Philip will have settled before he will cease.  
Philip would cease after he settled.  
? Philip settled before he would cease.  
? Philip will be going to cease after he will settle.  
?? Philip will settle before he will be going to cease.

For reasons explained on page 110, perfect aspect is not good on the right-hand side of conjunctions. On the left-hand side, these combinations are possible.

Philip have settled since he ceased.  
? Philip have settled since he ceased.  
Philip had settled since he ceased.  
? Philip had settled since he ceased.

In principle, it is perhaps not necessary to have any restrictions at all on the relations. The unreasonable combinations would presumably result in output with some degree of inconsistency. To take a simple example, the sentence ‘Philip is going to cease before he had settled’ says that a point in the future is prior to some point in the past. Such a statement in the output could then be detected later. For practical reasons however it is desirable to limit the ambiguity, and as a first approximation the following patterns are accepted.

\[
\begin{array}{c|c|c|c|c}
X & X & X & X \text{perfect} & X \\
X \text{past} & X & X & X \text{perfect} & X \text{past} \\
X & X \text{past} & & & \\
\end{array}
\]

‘X’ means any combination of external aspects. The same category on both sides is fine, and they can differ by an extra ‘past’ on one side, as so on.

---

6See the table on page 177.  
7121  
8In the meaning sense, see section 6.5.  
9In the most obvious literal interpretation.  
10Implemented by rel3a/2.
Representation of segments

At this stage of the analysis, a proposition consists of a set of situation type instances, and the not yet interpreted tense- and aspect-related information. When the segments are computed, two more pieces of data are needed: the time points that are used to express the meaning of the temporal relations (page 169), and the set of possible internal and external meanings (see the table on page 106). The tables below show the representations used for propositions, as input to the segment identifier (Pairs), and as output (Segment).

<table>
<thead>
<tr>
<th>Pairs</th>
<th>List of ‘Syn-P’</th>
<th>P</th>
<th>‘p(B,F,E,Asp,Is,Evs)’</th>
<th>Syn</th>
<th>Eventuality symbol</th>
<th>Eventuality symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segment</td>
<td>‘seg(B,F,PPs,Cs)’</td>
<td>PP</td>
<td>‘pp(B,F,E,Asp,Is,Evs)’</td>
<td>Cs</td>
<td>Time clauses 12</td>
<td>Time clauses 12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Asp</th>
<th>AV13</th>
<th>I1/I2</th>
<th>Ops15</th>
<th>EA16</th>
<th>IA17</th>
</tr>
</thead>
<tbody>
<tr>
<td>vb(1,Ops)</td>
<td>start</td>
<td>finite(...)</td>
<td>start</td>
<td>past</td>
<td>imperfective</td>
</tr>
<tr>
<td>(ops,Av,I1,I2)</td>
<td>step</td>
<td>infinitive(...)</td>
<td>end</td>
<td>present</td>
<td>imperfective</td>
</tr>
<tr>
<td>cence</td>
<td>participle(...)</td>
<td>imperfective</td>
<td>future</td>
<td>perfect</td>
<td></td>
</tr>
<tr>
<td>finish</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The single segment, with two propositions, expressed by ‘He arrived a moment after he turned off’ has the following form, with the eventualities omitted.

\[
\begin{align*}
\text{[seg}(B,F, & \text{, } \text{pp}(B,F,E_1,\text{vb(finite(past)},[\text{end}])),[i([\text{past},[],[]],\text{[end]})),[\ldots]), \\
\text{[pp}(B_2, & \text{, } \text{vb(finite(past}),[\text{start}])),[i([\text{past},[],[]],[\text{start}]),[\ldots]])
\end{align*}
\]

The lexical semantics of ‘arrive’ is ‘end(movement(...))’ and ‘turn off’ has the formalisation ‘start(movement(...))’, so both propositions have a lexical operator. Since simple past is unambiguously,18 they have only one element each in the variable ‘Is.’

Segment identification

The procedure find_segs/5 takes the eventualities (Pairs, Vars) plus the relations, and produces a list of segments. It replaces the relations with time clauses (see page 169), and it looks up the internal and external aspectual meaning of the propositions. For simplicity, a relation must have one unique interpretation. There are no ambiguities in the example,19 but in general that would seem to be

11Synthetic tense/aspect, aspectual verbs, lexical aspectual operators. Held in ‘Asp’ in the table on page 155.
12See page 178.
13Aspectual verbs.
14See the table on page 92.
15Ops is a list of these lexical aspectual operators.
16External aspect. This is also a list. The possible combinations can be found in the table on page 177.
17Internal aspect. Like the other two, this is a list.
18There are two meanings listed on page 106, but ‘habitual’ is ignored here. See the procedure infl1/3.
19See sections 1.1, 4.1, and 7.4.
a possibility. The algorithm is described in pseudo-code below, with references to the Prolog procedures on the right.

```prolog
FIND_SEGS(Vars,Pairs,ReIs) (find_segs/5)
begin
Segs, PPs, Cs ← empty lists
while Vars not empty do
  V ← head of Vars
  Vars ← rest of Vars
  Open ← list of V only
  while Open not empty do (closure/9)
    X ← head of Open
    Open ← rest of Open
    if there is a relation in ReIs involving X then
      ReIs ← ReIs minus that relation
      Y ← the other proposition in the relation
      Vars ← Vars minus Y
      find the data for X and Y in Pairs
      use this to find the meaning of the relation
      update the data for X and Y in Pairs
    else
      find the data for X in Pairs
      if there is no internal/external data (Is) then add it
      add the data to PPs
    end if
  end while
Segs ← Segs + { PP, Cs }
PPs, Cs ← empty lists
end while
return Segs
end
```

And here follows the Prolog code for finding segments.

```
segments.pl
% Copyright 2001 Tomas By
% http://web.bazum.net/homepages/tomas/thesis/
% Copyrighted freeware. All rights reserved.
:- use_module(library(library(lists), [append/3, reverse/2, member/2, select/3, delete/3]).
segments(ReIs, Vbs0, Es0, Segs, Es) :-
  map_rels(ReIs, Vbs0, Es0, Es, [], ReIs, [], AdjEs),
  map_rels(AdjEs, Vbs, Es, Pairs, ReIs, [], Segs).
map_rels([r(ReIs, X1, X2)|ReIs], Vbs0, Es0, Es, Rs0, Rs, As0, As) :-
  ( relation(X1, X2, ..., Xn) ->
    format("segments: reflexive relation:[r(ReIs, X1, X2)],
          Vbs0 = Vbs0, Es0 = Es0, Rs0 = Rs0, As0 = As0
            ; member(rb(X1, X2), Vbs0), X == X1 ->
          check_rhs(X1, X2, Vbs0, Es0, Es2, Rs0, Rs1, As0, As1)
        ; select(de(X, Ev, Attrs), Es0, Es1), X == X1 ->
          ( coerce_entity(Ev, Attrs, Sp, Es) ->
            check_rhs(X1, X2, [rb(X1, Sp, Evs)|Vbs0], Vbs1, Es1, Es2, Rs0, Rs1, As0, As1)
```


P = pp(S,F,E,Sp,Is,Events),
Ys1 = Ys0, Opex = Opexo, Pairs = Pairs0,
&i = Rs0, Ps = [P[Ps0]], Cs = Cso ),
closure(Open,Ys,Js,Pairs,ksi,ks,Ps,Cs,Seg).
closure([], Ys, Ys,, RevYs, ksi, RevPs, Cs, Seg) :-
reverse(RevYs, ksi),
reverse(RevPs, Ps).
( Ps = [pp(S,F,---,---,---,---,---)], Ys ) ->
Seg = seg(S,F,Ps,Cs) ;
format('closure:illegal_segment :\'v', [Seg]).
abort .
c_rel(S, X1, X2, Pairs0, RelCs) :-
lookup(X1,p(B1,F1,E1,Spi,Isi,Evsi),Pairs0),
lookup(X2,p(B2,F2,E2,Spi,Is2,Ev2),Pairs0),
infl_form(Spi, I1, AVsi, Ops1),
infl_form(Sp2, I2, AVs2, Ops2),
findall(Y, ( infl(I1,E1,---),
infl(I2,EA2,---),
relation(S,B1,F1,E1,B2,F2,E2,EA2,Cs),
Y = y(E1,EA2,Cs) ), Ys),
length(Ys, N),
( N = 1 ->
infl(I1,EA1,Is1),
infl(I2,EA2,Is2),
relation(S,B1,F1,E1,B2,F2,E2,EA2,RelCs),
update(X1, p(B1,F1,E1,Spi, [i(EA1, Is1, AVs1, Ops1) | [Is1, Evsi]], Pa0, Pa0),
update(X2, p(B2,F2,E2,Spi, [i(EA2, Is2, AVs2, Ops2) | [Is2, Ev2]], Pa0, Pa0)) ;
N = 0 ->
format('closure:illegal_relation: \'v', S, Sp1, Sp2),
fail ;
format('closure:ambiguous_relation: \'v', [Ys]),
fail ).
remove(X0, Ys0, Ys) :-
( select(X,Ys0, Ys1), X0 == X ->
Ys = Ys1
; Ys = Ys0 ).
lookup(X,P,Pairs) :-
( member(X0,0, Pairs ), X0 == X ->
P = P0
; format('lookup_failed: \'v', [X, Pairs]),
abort ).
infl_form(vb(I0,Ops), I, [], Ops) :-
remove_passive(I0, I).
infl_form(as(AV, I1, I2, Ops), I, AVs, Ops) :-
remove_passive(I1, I),
remove_passive(I2, I2),
( ( I2 = infinitive([]) ; I2 = participle([]) ) ->
I = I1,
AVs = [AV]
; format('infl_form:illegal_ tense/asp_of_asp_v_verb_complement: \'v', [I2]),
fail )).
remove_passive(I0, I) :-
X = X0,
delete(X0, passive, Xs),
I = . , Xs.
update(X,P, Ps0, Ps) :-
( select(X0, Ps0, P, P, P, Ps0), X0 == X ->
P = [X-P[Ps1],
format('update_failed: \'v', [X, Ps0]),
abort ).
infl(finite([present]), [], []).
infl(finite([present]), [present], []).
infl(finite([present]), [past], []).
infl(finite([present]), [present, future], []).
infl(finite([past]), [past], []).
infl(finite([past, perfect], [past, perfect], []).
infl(finite([past, perfect]), [past, perfect], []).
infl(finite([past, perfect]), [past, perfect], []).
inf1(\text{finite}([\text{present}, \text{progressive}]), \text{[present, future]}, \text{[]}).
inf1(\text{finite}([\text{present}, \text{progressive}]), \text{[present]}, \text{[\text{imperfective}]}).
inf1(\text{finite}([\text{past}, \text{progressive}]), \text{[past]}, \text{[\text{imperfective}]}).
inf1(\text{finite}([\text{past, perfect}, \text{progressive}]), \text{[present, perfect]}, \text{[\text{imperfective}]}).
inf1(\text{finite}([\text{past}, \text{perfect}, \text{progressive}]), \text{[past, perfect]}, \text{[\text{imperfective}]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{present}]}, \text{[\text{imperfective}]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{present, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{present, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{present, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future}]}, \text{[]}).
inf1(\text{finite}([\text{modal}, \text{will}]), \text{[\text{past, future, perfect}]}, \text{[]}).

(inf1/3 implements the table on page 106, and relation/10 the one on page 169.)
A.6 Discourse interpretation

The remaining operations apply to each segment separately. They transform the eventualities, interpret the external time specifications to find appropriate temporal locations for the propositions, and integrate the entities with existing ones in the database.\(^1\) Of the semantic units in the picture below, which is similar to those on pages 3 and 84, the internal and external times are the most relevant here. Entities, roughly corresponding to nouns phrases\(^2\) are not shown.

\[^1\]The data-flow is illustrated by the graph on page 132.
\[^2\]See page 157.
\[^3\]See the table on page 106.
movement).\textsuperscript{4} is not supported. The selection is implemented by \texttt{apply}\_\texttt{asp}/6 and shown in the table below. The variable names are the same as on page 171.

<table>
<thead>
<tr>
<th>Verb group</th>
<th>Asp. verb</th>
<th>Lexical operator</th>
<th>Procedure (applied to \texttt{Ev}s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( (A_x) )</td>
<td>( (A_v) )</td>
<td>( (O_p) )</td>
<td></td>
</tr>
<tr>
<td>( [\text{imperfective}] )</td>
<td>( [\text{start}] )</td>
<td>( \text{-} )</td>
<td>imperfective/3</td>
</tr>
<tr>
<td>( [\text{start}] )</td>
<td>( \text{-} )</td>
<td>( \text{-} )</td>
<td>start/3</td>
</tr>
<tr>
<td>( [\text{stop}] )</td>
<td>( \text{-} )</td>
<td>( \text{-} )</td>
<td>end/3</td>
</tr>
<tr>
<td>( [\text{cease}] )</td>
<td>( \text{-} )</td>
<td>( \text{-} )</td>
<td>end/3</td>
</tr>
<tr>
<td>( [\text{finish}] )</td>
<td>( \text{-} )</td>
<td>( \text{-} )</td>
<td>end/3</td>
</tr>
<tr>
<td>( [\text{start}] )</td>
<td>( \text{-} )</td>
<td>( \text{-} )</td>
<td>start/3</td>
</tr>
<tr>
<td>( [\text{end}] )</td>
<td>( \text{-} )</td>
<td>( \text{-} )</td>
<td>end/3</td>
</tr>
<tr>
<td>( [\text{imperfective}] )</td>
<td>( \text{-} )</td>
<td>( \text{-} )</td>
<td>imperfective/3</td>
</tr>
</tbody>
</table>

That combinations of these meanings are not handled is an obvious defect in this analysis of 'internal' aspect. Possible improvements are discussed on page 126.

External time (section 7.1), the more assertive part of the meaning of the verb group syntax,\textsuperscript{5} has been classified into the following eleven forms, with the linguistics realisations listed to the right.

\begin{align*}
\text{External aspect} & \quad \text{Syntactic realisation} \\
\text{[present]} & \quad \text{Present} \\
\text{[past]} & \quad \text{Past} \\
\text{[present,future]} & \quad \text{Present/progressive/will/is going to} \\
\text{[past,future]} & \quad \text{would/was going to} \\
\text{[past,past]} & \quad \text{Past perfect} \\
\text{[present,future,past]} & \quad \text{will + Perfect} \\
\text{[present,future,future]} & \quad \text{will be going to (Not supported)} \\
\text{[present,perfect]} & \quad \text{Present perfect} \\
\text{[past,perfect]} & \quad \text{Past perfect} \\
\text{[present,future,perfect]} & \quad \text{will + Perfect} \\
\text{[past,future,perfect]} & \quad \text{would + Perfect} \\
\end{align*}

As discussed on page 108, the interpretation of the external aspect is governed by two rules that operate on the lists of symbols. These are implemented by \texttt{ext}\_\texttt{time}/7, and a full presentation of the meanings can be found on page 107.

\begin{align*}
\text{Initial} & \quad \text{present} \quad T = \text{Base} \\
\text{past} & \quad T < \text{Base} \\
\text{Non-initial} & \quad \text{past} \quad T_y < T_x \\
\text{future} & \quad T_y > T_x \\
\text{perfect} & \quad T_y < T_x; T_x : \text{Events relevant}; \text{Add } T_y \text{ to focus list}
\end{align*}

The effect of these rules is to, among existing times in the database, and new ones, find referents for the time points associated with the proposition \((B, F, E)\) on page 171, which determine the temporal location of the eventuality, and, in the case of perfect aspect, the 'relevance' expression (see section 6.5).

\textsuperscript{4}For other similar examples, see page 26.

\textsuperscript{5}See the table on page 106.
### The database

The Prolog database is used to store the discourse state, which contains eventualities, entities, temporal information, and book-keeping data. The sentence/4 clauses are simply asserted in the order in which the sentences occur, and contains the symbols created in the interpretation of the sentence. Those clauses are not subsequently used by the code.

<table>
<thead>
<tr>
<th>Eventuality</th>
<th>$ev(E, \text{Situation}, Attributes)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>$\text{time(Clause)}$</td>
</tr>
<tr>
<td>$Clause$</td>
<td>$\text{happen}(T, E)$                 $\text{Eventuality } E \text{ happens at time } T$</td>
</tr>
<tr>
<td></td>
<td>$\text{relevant}(T, E)$              $\text{Eventuality } E \text{ is relevant at } T$</td>
</tr>
<tr>
<td></td>
<td>$\text{seq}(T_1, T_2)$               $T_1 \text{ is before } T_2$</td>
</tr>
<tr>
<td></td>
<td>$\text{dur}(T_1, T_2, X)$            $\text{Time from } T_1 \text{ to } T_2 \text{ is } X$</td>
</tr>
</tbody>
</table>

| Focus times | $\text{focus}(T's)$                   $\text{(Time symbols)}$ |
| Counters    | $\text{centres}(X's)$                 $\text{(Entity symbols)}$ |
| History     | $\text{sentence(}\text{Words, }E_s, X_s, T_s)$ |

There are several more or less obvious requirements for a discourse database to be consistent: all the entity symbols used in $ev/3$ clauses must also occur in exactly one entity/2 clause each; eventuality symbols in time/1 must likewise be defined by $ev/3$ clauses; each eventuality symbol must occur in one $\text{happen}/2$ clause; the seq/2 relation must not form cycles etc. These are not checked.

The sentence ‘Philip had written a letter’ has the following temporal structure.

```prolog
[past, perfect]
times: t1 t2 t0 now
  e0 e1 (e0)
  (e1)
```

And analysing it, as the first sentence in the discourse, produces these clauses.

```prolog
ev(e0, activity, [concept=creation, agent=e0, object=x1]).
ev(e1, transition('x1', x1 & ('x1' location=\(x0' location))). [concept=creation, agent=e0, object=x1]).
extity(x0, [name='Philip', is_a=male]).
extity(x1, [is_a=document]).
time(happen(t1, e0)).
time(happen(t2, e1)).
time(seq(t0, now)).
time(seq(t1, t0)).
time(seq(t1, t2)).
time(seq(t2, t0)).
time(seq(t2, t0)).
time(relevant(t0, e0)).
time(relevant(t0, e1)).
focus([t0]).
centres([e0]).
counters([2, 3]).
sentence(['Philip', 'had', 'written', 'a', 'letter', [x0, x1], [x0, e1], [now, t0, t1, t2]).
```

The numbers in counters/3 are the ‘next’ indices for the symbols for eventualities, entities, and times.

---

$^6$See also section 6.6.

$^7$The entity ‘$X$’ is not interpreted further.
Times

Here follows the algorithmic description of the temporal part of the discourse interpretation. This procedure reads the database but does not write to it.

TIMES(Seg,B,F)  
begin  
Evs, Ts, Cs, Fs ← empty lists  
PPs, RelCs ← data from Seg  
while PPs not empty do  
PP ← head of PPs  
PPs ← rest of PPs  
Evs, Ts, Cs, Fs ← Evs, Ts, Cs, Fs + INTERPRETATIONS(PP)\(^8\)  
end while  
Cs ← Cs + RelCs  
return Evs, Ts, Cs, Fs  
end

INTERPRETATIONS(PP)  
begin  
B, F, E, Is, Events ← data from PP  
for each I in Is do  
AssEv, ImpEv ← asserted/implied events  
ExtAsp ← 'external' aspect, from I  
NewTs, NewCs, E, R ← interpretation of ExtAsp\(^9\)  
ImpVs, AssVs ← time clauses  
if the ImpVs can be found in the database\(^10\) then  
NewTs ← NewTs + ImpVs  
NewEv ← AssEv  
NewCs ← NewCs + AssVs  
else  
NewTs ← NewTs + ImpVs + AssVs  
NewEv ← AssEv + ImpVs  
NewCs ← NewCs + ImpVs + AssVs  
end if  
if R is defined then  
NewCs ← NewCs + 'relevance' clauses  
NewFs ← reference point  
else  
NewFs ← event point  
end if  
return NewEv, NewTs, NewCs, NewFs\(^8\)  
end for  
end

Outputs are the new eventualities, new times (uninstantiated variables), time clauses, and the new focus times. The code follows on the next three pages.

\(^8\)The procedure times/8 returns several solutions, one per element in Is, via back-tracking.
\(^9\)See the table on page 107.
\(^10\)See section 7.2.
times.pl
% Copyright 2001 Tomas By
% http://www.baswn.net/homepages/tomas/thesis/
% Copyrighted freeware. All rights reserved.
:- use_module(library(lists), [append/3, reverse/2, member/2]).
:- use_module(library(ords)).
:- op(900, fy, []).
:- multifile ev/3, entity/2, time/1, sentence/4, focus/1, centres/1, counters/3.
:- dynamic ev/3, entity/2, time/1, sentence/4, focus/1, centres/1, counters/3.

times(Seg(R,F,Ps,RelCs), R,F,Form,EvS,Keys, Ts, Cs, Fs) :-
times(Ps,Form, [], [], [], Ts, [], PropCs, [], Fs),
append(PropCs,RelCs, Cs).

times([P|Fs], [Extasp|Form], EvS0, EvS, Ts0, Ts, Cs0, Cs, Fs0, Fs) :-
P = pp(R,F,E,...,Is,Events),
nmember(Is, Extasp, Intasp, Asp, Dps),
apply_ap(Extasp, Intasp, Asp, Dps, Events, ImpEvS, AssEvS),
ext_time(B,F,Extasp, NewTs0, NewCs0, X,R),
time_classes(Is, ImpEvS, Cs, ImpYars),
time_classes(AsEvS, E, Te, AssCs, AssYars),
( instantiated(ImpCs) ->
  var_union(NewTs0, AssYars, NewTs),
  NewEvS = AssEvS,
  append(NewCs0, AssCs, NewCs2)
; var_union(NewTs0, ImpYars, NewTs1),
  var_union(NewTs1, AssYars, NewTs),
  append(ImpEvS, AssEvS, NewEvS),
  append(NewCs0, ImpCs, NewCs1),
  append(NewCs1, AssCs, NewCs2) ).
( % = yes(Ttr),
  rel_evrs(AssEvS, Te, Tr, MoreCs),
  append(NewCs2, MoreCs, NewCs),
  NewFs = [Tr] # [Tt, Tr] ;
  % = no,
  NewCs = NewCs2,
  NewFs = [Te] ),
append(EvS0, NewEvS, EvS1),
append(Ts0, NewTs, Ts1),
append(Cs0, NewCs, Cs1),
append(Fs0, NewFs, Fs1),
times(Ps, Form, EvS1, EvS, Ts1, Cs1, Cs, Fs1, Fs).
times([], [], [], [], [], Ts, Cs, Cs, Fs, Fs).

time_classes([[Ei, E2]|Exs], T0, T, [happen(T0,X)|Cs], [T0|Ys]) :-
  E1 = ev(X,E1,...),
duration(Ev1, D1),
time_classes(Exs, T0, D1, E2, Cs, T, Ys),
time_classes(E, T, T, [happen(T,X)|T], Y) :-
  E = ev(X,...),
time_classes([], T, T, [], []).

time_classes([E|Exs], T1, O, E2, [happen(T1,X)|Cs], T, Ys) :-
  E1 = ev(X2,E1,...),
duration(Ev2, O),
time_classes(Exs, T1, O, Cs, T, Ys),
time_classes([E|Exs], T1, I, E2, [happen(T2,X)], seq(T1,T2)|Cs], T, [T2|Ys]) :-
  E2 = ev(X2,E2,...),
duration(Ev2, O),
time_classes(Exs, T2, O, Cs, T, Ys),
time_classes([], T1, I, E2, [happen(T1,X)], T1, []) :-
  E2 = ev(X2,...),
time_classes([], T1, I, E2, [happen(T2,X)], seq(T1,T2)|T2, [T2|Ys]) :-
  E2 = ev(X2,...).
rel_evrs([Ev|EvS], Te, Tr, [seq(Te,Tr)], relevant(Tr,X)|Cs)] :-
  Ev = ev(X,...),
rel_ev([Ev|EvS], Te, Tr, Cs),
rel_ev([], [], []).
var_union(As, Bs, Cs)
; var_union(As, [A | Bs, Cs])
var_union([], Bs, Bs).

instantiated([C | Cs]) :-
  time(C),
  instantiated(Cs).
instantiated([]).

ext_time(X, F, Is, Ts, Cs, E, R) :-
  format('ext_time : "w"-w"-w",[B,F,T],
  new_info Infos),
times_init(Is, F, Infos),
  get_info(Infos, Ts, Cs, E, R).

times_init([], Is, F, Infos, Info) :-
  ( I = present,
    time_sin(Is, F, Infos, Infos)
  ; I = past,
    time_seq(Is, F, Infos, Infos)
  ).

new_info(Info) :-
  Info = info([], [], []).

get_info(Infos, Ts, Cs, E, R) :-
  Infos = info(Ts, Cs, Rs, Rs),
  ( Rs = [E] ->
    true
  ; R = yes(R)
  ; Rs = [E] ->
    R = yes(R)
  ; R = no,
    format('ext_time :event:time:well-defined:"w"">",[E]) ).

time_sin(T, Infos, Info).

time_seq(T1, T2, Infos, Info) :-
  Infos = info(Ys0, Cs0, Es, Rs),
  ord_add_element(Ys0, T1, Ys1),
  ord_add_element(Ys1, T2, Ys2),
  Cs = [seq(T1, T2)]Cs0,
  Infos = info(Ys, Cs, Es, Rs).

time_relevant(R, Infos, Info) :-
  Infos = info(Ys, Cs, Es, Rs),
  Infos = info(Ys, Cs, Rs, Rs).

time_events(E, Infos, Info) :-
  Infos = info(Ys, Cs, Es, Rs),
  Infos = info(Ys, Cs, Rs, Rs).

apply_assoc([], []).
start(Evs, ImEv, AsEvs).
apply_asp([[], stop, [], Evs, ImEv, AsEvs] ->
end(Evs, ImEv, AsEvs).
apply_asp([[], [[], Evs, ImEv, AsEvs] ->
end(Evs, ImEv, AsEvs).
apply_asp([[], [[], Evs, ImEv, AsEvs] ->
end(Evs, ImEv, AsEvs).
apply_asp([[], [[], Evs, ImEv, AsEvs] ->
end(Evs, ImEv, AsEvs).
apply_asp([[], [[], Evs, ImEv, AsEvs] ->
end(Evs, ImEv, AsEvs).
apply_asp([[], [[], Evs, ImEv, AsEvs] ->
end(Evs, ImEv, AsEvs).
apply_asp([[], [[], imperfective], Evs, ImEv, AsEvs] ->
imperfective(Evs, ImEv, AsEvs).

start(Evs, ImEv, AsEvs) :-
  ( initial_zero(Evs, X, _) ->
    ImEv = [X], AsEvs = X
  ; Evs = [Ev], _ ->
    Ev = ev(X, E),
    ImEv = [Ev],
    AsEvs = [ev(_, transition(X, X), [])]
  ; format('start : illegal events : \n\n', [Evs]),
    fail).

imperfective(Evs, ImEv, AsEvs) :-
  ( initial_zero(Evs, X, Ys) ->
    Ys = [X] _ ->
    ImEv = [X], AsEvs = [X]
  ; write('imperfective : eventuality has no duration'), nl,
    fail)
  ; Evs = [Ev], _ ->
    ImEv = [Ev], AsEvs = [Ev]
  ; write('imperfective : eventuality '), nl,
    fail).

end(Evs, ImEv, AsEvs) :-
  ( Evs = [Ev] _ ->
    Ev = ev(X, E),
    ImEv = [Ev],
    AsEvs = [ev(_, transition(X, X), [])]
  ; reverse(Evs, RevEvs),
    initial_zero(RevEvs, RevEvs, RevEvs),
    reverse(RevEvs, Ys),
    reverse(RevEvs, Ys) ->
    ImEv = Ys, AsEvs = Ys
  ; format('end : illegal events : \n\n', [Evs]),
    fail).

initial_zero([X | Xs], Ys, Zs) :-
  ev_duration(X, 0),
  initial_zero(Xs, [X], Ys, Zs).

initial_zero([X | Xs], Ys, Zs) :-
  ev_duration(X, 0) ->
  initial_zero(Xs, [X | Ys], Ys, Zs)
  ; reverse(Ys, Ys),
    Zs = [X | Xs]
  .

initial_zero([], Ys, Zs, []) :-
  reverse(Ys, Ys).

ev_duration(X, 0) :-
  X = ev(_, E),
  duration(E, 0).

duration(X, 0) :-
  ( dur(E) ->
    D = X
  ; D = -1,
    format('duration : not an eventuality : \n\n', [X]))
  .

dur(state, 1).
dur(point, 0).
dur(transition(_, 0).
dur(activity, 1).
Entities

The handling of entities in discourse is based on Allen (1995, chap. 14 & sect. 16.3) but greatly simplified. A list is kept of 'centres,' which are the entities most likely to co-refer with pronouns. Allen (1995, pp. 436-437) has rules for updating this list, but here it is only assigned the proper nouns in the first sentence, and does not change after that. Pronoun interpretation, here, consists of checking the centres, and the proper nouns in the same sentence, to see if the 'is_a' attributes are the same.

terms.pl
% Copyright 2001 Tomas By
% http://www.basun.net/homepage/tomas/thesis/
% Copyrighted freeware. All rights reserved.
:- use_module(library(lists), [member/2, append/3]).
entities(Ps,NonPs,ImpEs,NewEs,NewCs) :-
  centres(Cs),
  de_ents(NonPs,NonPs),
  de_ents(ImpEs,ImpEs),
  named_ents(NonPs,NewEs),
  pros_ents(Ps,NewEs,Cs,ProEs),
  ( Cs = [] ->
    ent_sums([],NewEs,NewCs),
    append(ProEs,NonPs,Temp),
    append(Temp,ImpEs,NewEs).
  de_ents([Head|Tail], [Entity|Tail]) :-
    de_ents(Tail, Entity).
  ent_sums([], []).
  ent_sums([Entity|Tail], [Entity|Tail]).
pros_ents([], Ps, Cs, Es) :-
  P = de([a..]),
  ( member(C,Cs),
    entity(C,Cs),
    compatible_attributes(A,B) ->
      X = C,
      Es = Es0 ;
    member(Entity, Ps),
    compatible_attributes(A,B) ->
      X = Entity,
      Es = [Entity|As0] ;
    member(Entity, Ps),
    compatible_attributes(A,B) ->
      X = Entity,
      Es = [Entity|Es0]
  pros_ents(Ps,Es,Cs,Es0).
pros_ents([], [], [], []).
compatible_attributes(A,B) :-
  format('compatible attributes: "u""n",[A,B]),
  member(is_same, A),
  member(is_same, B),
  A == B.
named_ents([], []).
  E = entity(Head),
  ( member(Entity, Head) ->
    E = [Entity|Es0],
    E = Es0 ),
  named_ents(Es, Es0).
  named_ents([], []).

While the method used here for entity interpretation is extremely simple, it does produce correct results for the example (sections 1.1, 4.1, and 7.4).
Discourse

The complete discourse interpretation system, whose data-flow is shown on page 132, is used by making one call to init/0, to initialise the database, and then repeated calls to sentence/1, once for each sentence in the discourse. The code listed below is the only place where the information in the database is changed.

discourse.pl

% Copyright 2001 Tomas By
% http://www.basum.net/homepages/tomas/thesis/
% Copyrighted freeware. All rights reserved.
:- use_module(library(lists), [member/2, append/3]).
:- multfile er/3, entity/2, time/1, sentence/4, focus/1, centres/1, counters/3.
:- dynamic er/3, entity/2, time/1, sentence/4, focus/1, centres/1, counters/3.

init :-
    retractall (er(_,_,_)),
    retractall (entity(_,_)),
    retractall (time(_,_)),
    retractall (sentence(_,_,_,_)),
    retractall (focus(_)),
    retractall (centres(_)),
    retractall (counters(_,_,_)),
    assert (focus([])),
    assert (centres([])),
    assert (counters(0, 0, 0)).

sentence (Vs) :-
    s(_, LF, Vs, [I]), !.
    report ("n n' n", [Vs]),
    collect (LF, Ys, Es, Ns, PSs, IPs, SSs, Rs),
    segments (Rs, Ys, Ns, PSs, IPs, SSs, Rs),
    entities (Ps, Es, IPs, SSs, Centres),
    retract (counters(Ne0, Ne0, St0, St0)),
    new Entities (Ne0, Ne0, St0, EntSyns),
    assert terms (Ne0),
    length (Segs, S),
    assert (sentence has-"n n' n", [S]),
    sentence_segments (Segs, I, Ne0, Ne0, St0, St0, [I], EntSyns, [I], Ts),
    report ("n n' n", [I]),
    report entities (Ne0),
    report ("centres n n' n", [Centres]),
    assert (counters(Ne0, Ne0, St0, St0)),
    retract (centres([])),
    assert (centres (Centres)),
    assert (sentence (Vs, EntSyns, EvSyns, Ts)).

sentence_segments ([Seg Segs], [N1, Ne0, Ne0, St0, St0, EvSyns0, EvSyns, Ts0, Ts]) :-
    B = now,
    findall (X, ( times ([Seg, B, B, B, B, B, B, B, B, B],
                     X = x, Ns),
              length (Ns, N2),
              report ("segment n n' n", [EvSyns, Ts0, Ne0, Ne0, St0, St0, EvSyns0]),
              retract (focus (Fs)),
              ( Fs = [] ->
                 r = _ ; member (F, Fs) )
            times (Seg, B, B, Form, Evs, Ne0s, Evs, Ne0s) ->
            report_form (Form),
            new_events (Evs, e, Ne0, Ne0, Evs, Ne0s),
            new_times (Ne0s, Ts0, St0),
            append (EvSyns0, Ne0s, EvSyns, EvSyns1),
            append (Ts0, Evs, Ts1),
            assert terms (Evs),
            assert clauses (Ev),
            report clauses (Ev),
            report_events (Evs).
TEARS IN THE RAIN

; New = New0, St = St0, EvSyns = EvSyns0, Ts = Ts0,
  NewFs = Fs).
  assert(focus(NewFs)).
  report('focus_time(s): w~s~n', [NewFs]),
  N3 is St + 1.
  sentence_segments(Segs, N3, New, New, St, EvSyns, EvSyns, Ts, Ts).
  sentence_segments([], New, New, St, EvSyns, EvSyns, Ts, Ts).

assert_terms((X|Xs)) :-
  assert(X),
  assert_terms(Xs).
assert_terms([]).

assert_classes((C|Cs)) :-
  assert(time(C)),
  assert_classes(Cs).
assert_classes([]).

As a final example, 'Philip arrived in Paris' produces the following output.

| ?- init.

yes
| ?- sentence(["Philip", arrived, in, "Paris"]).

Philip.arrived, in, Paris
sentence has 1 segment(s)
segment 1: 1 interpretation(s)
  [past]
times: t0 now
e0
e1
e2
e0: transition(x0^location=x2, "x0^location=x2"
  [concept=movement, agent=x0, object=x0, target=x1, source=x2]
e1: activity [concept=movement, agent=x0, object=x0, target=x1, source=x2]
e2: transition("x0^location=x1, x0^location=x1"
  [concept=movement, agent=x0, object=x0, target=x1, source=x2]
  focus time(s): [t0]
x0: entity [name=Philip, is_a=male]
x1: entity [name=Paris, is_a=city]
x2: entity [is_a=location]
centres: [x0.x1]

yes
| ?- 

The procedures report/1, report_entities/1, report_form/1, report_clauses/1,
and report_events/1, which produce most of the output above, are not listed
in this Appendix, but they are provided in the electronic distribution (see
page 131). The full analysis of the example is shown in section 7.4.
References


Index

abort, 26
accomplishment, 34-38, 41, 45, 82n, 94, 95
achievement, 34, 35, 37, 38, 41, 45, 82n, 94, 95, 123
action type, see situation type
activity, 22, 34, 35, 37, 38, 40, 41, 45, 82n, 94, 95
adjunct, time, see time adjunct
adverbial, 14, 17
containment, 24
frame, 23n
sentence, 18
after, 17, 23, 47, 104
ago, 23
algorithm, discourse interpretation, 51, 76
Allen, 29, 44-46, 48, 49, 73, 73n, 74-81, 82n, 83, 84, 84n, 93, 94, 103, 105, 107, 107n, 108, 120, 121, 123-125, 133, 183
almost, 36
along, 99
always, 26, 125
and, 100
anterior, 28, 29, 50
Aristotelian aspect, 29, see also situation type
Aristotle, 27, 29, 32, 42, 65
arrival, 100
arise, 39, 123, 171
as, 125
aspect, 3, 7, 16, 25-26
Aristotelian, 29, see also situation type
progressive’, 97n
inceptive’, 69
progressive’, 26, 97n, 99n
perfect, 2, 5, 7, 8, 10-12, 14, 17, 24, 26, 29, 38, 42, 46, 49, 51, 58, 69, 82, 83n, 86, 92, 95, 102-103, 107, 108, 110, 113, 115, 121, 124, 176, 177
progressive, 2, 7, 8, 10-12, 15, 17, 18, 25, 26, 30, 34, 38, 42, 49, 58, 69, 82, 83, 88, 90, 92, 94, 95, 100, 121, 124
propositional, 29, see also situation type
‘progressive’, 69
semantic, 29, see also situation type
‘terminative’, 69
verb, 29, see also situation type
aspectual class, 29, see also situation type
aspectual type, 29, see also situation type
aspectual verb, 2, 3, 8, 26, 47, 52, 59, 69, 70, 75, 82-84, 95, 99-100, 121, 124, 126, 176
at, 23, 104
atelic, 33
auxiliary verb, 7-11, 26, 86, 88
away, 26
backshift, 13, 15, 19, 124
take, 41
ballim, 63, 65, 93
bare infinitive, see infinitive
tense form, 7
be, 7-9, 24, 88
be about to, 9, 15
be going to, 9, 15, 47, 142
be to, 9, 15
become, 58
been, 88, 89
before, 17, 23, 104
begin, 2, 8, 30, 69, 123
being, 88
believe, 66
between, 23, 104
been, 63
Hammond, 26, 29, 31-33, 35, 126n
hold, 40
by, 23, 24
cake, 41
call, 7
can, 7, 9, 65
cardinal numbers, 20
carefully, 31, 36
cause, 47, 52, 97
class
aspectual, 29, see also situation type
correction, 101, 102, 121, 155, 166-168
composite sense, of proposition, 65
Comrie, 2, 14, 29, 33, 44n, 36, 48, 53, 102
conceptual graphs, 63, 68
conjunction, 2, 3, 10, 23, 47, 104, 107, 107n-111, 113, 139, 146, 155, 167, 170
‘containers’, 24
containment adverbial, 24
continue, 8
conversion, 11
Cooper, 42
gopia, 29, 9, 24
copular verb, 8, 9, 24
could, 9
count noun, 30
counterfactual, 10, 13, 19, 23n, 34, 63n
Cuttar, 29, 46, 63, 68, 68n, 69-72, 81, 82, 82n, 83, 105, 107, 107n, 120, 123-125
daily, 25
date, 9
Davidson, 93
DCG, see definite clause grammar
de dicto, 65
de re, 65
definite clause grammar, 4, 121-128
definitely, 31
determiners, 21
Dinsmore, 26, 29, 46, 48, 57, 63-65, 67-72, 81, 82n, 83, 93, 103, 105, 107, 107n, 108, 120, 121, 123-125
Diodorus Cronus, 43
discourse interpretation algorithm, 51, 70
discourse representation theory, 46, 49-57, 82, 83, 120
discover, 41
divided sense, of proposition, 65
do, 9, 88
Donnellan, 66, 67
Dowty, 25, 29, 31, 33, 35, 36, 42
drawn, 36
duration adjunct, 19, 21-23, 35, 37
during, 23, 123
dynamic, 29, 31, 34, 37
-ed participle, see past participle
‘en-gressive’ aspect, 97n
emphatic positive, 9
environment, 63, see also partitioned representations
event, 47-49, 52, 75, 82n
event calculus, 93
event nucleus, 38, 39, 49
event point, 27, 68
eventive noun, 4, 23, 101-102, 110, 121, 147, 166
every, 25
exactly, 22
experience, 30
extended tenses, see progressive aspect
external time, 2-4, 84, 85, 95, 96, 102, 105-108, 113, 120, 121, 124, 125, 130, 132n, 147, 160-172, 176, 177
Fauconnier, 61-65, 67, 68, 68n, 125
finish, 8, 26, 36, 69
finite verb, 7, 10, 11, 16
finite verb group, 10, 11, 33-16, 18
following, 23
for, 23, 24, 37, 38
force, 31
formal logic, 42
frame adverbial, 23n
frequency adjunct, 19, 25-26, 31, 124
from, 23, 104
full verb, 7
future, 15
‘future perfect’, 50, 69
‘future tense’, 7, 15, 27, 50, 58, 69, 82, 95
Caeldon, 9, 30, 37, 42, 44, 45, 123
Caudal, 86
generative lexicon, 40
gerund, see nominal clause
give, 8, 26
Gross, 73
Grosvenor, 29, 46, 48, 57-62, 82, 82n, 83, 93, 94, 103, 107n, 120, 121
habitual, 13, 15, 16, 25, 26, 31, 38, 96, 121, 124, 125, 171
have, 8, 26, 89
have got to, 9
have to, 9
Hendrix, 63
Hitzeman, 29, 46, 48, 57-62, 82, 82n, 83, 93, 94, 103, 107n, 120, 121
Hwang, 29, 73-75, 120
imperative mood, 7, 10, 12, 19, 92
imperfective, 14, 25, 42, 45n, 94, 99, 100, 127
imperfection, 3, 24, 37-40
‘inceptive’ aspect, 69
indicative mood, 10, 12, 92
infinitive, 7, 10-12, 18, 19, 89, 92
-ing participle, see present participle
‘Ingressive’ aspect, 26, 97n, 99n
inside of, 24
internal time, 2-4, 84, 85, 95-97, 99, 100, 106, 109, 115, 121, 124, 126, 130, 171, 172, 176, 177
interrupt, 26
intervals versus points, 44, 123
inversion, 9, 88
Joshi, 73
Kameyama, 57, 62, 107n
Kamp, 29, 38, 40, 46, 48-54, 56, 57, 63, 82n, 83, 84, 93, 94, 103, 106, 107, 120, 121
keep, 8
keep on, 8
kick, 34
Kowaish, 93
Lasnik, 42
last, 21
lexical semantics, 95, 97-102, 147-149
le, 31
lee, 31, 32
location adjunct, 19, 21-23, 37
location time, 29, 50, 52
logic, 42, 43, 93
logicistic form, 4, 88, 131, 132, 146-147, 155
main verb, 7, 9, 10
make, 31
Martins, 63
mass noun, 30
may, 9
mental spaces, see partitioned representations
metonymy, 67
metric tense logic, 43
might, 9
miss, 34
modal verb, 7, 9, 11, 15-16, 24, 28, 82, 89, 95
Moens, 29, 30, 38-40, 42, 46, 48, 49, 57-62, 82, 82n, 83, 93, 94, 103, 107n, 120, 121, 123, 126, 127
mood, 10, 12
imperative, 7, 10, 12, 19, 92
indicative, 10, 12, 92
past subjunctive, 10, 12, 92
present subjunctive, 10, 12, 92
subjunctive, 7, 10, 12, 19, 92
must, 9
necessarily, 65
need, 9
negation, 9
next, 21
node, 2
nominal clause, 11
nominalization, 3, 4, 11, 16, 18-19, 30, 82, 84, 100-101, 110n, 121, 176
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>past tense, 7, 11-14, 27, 50, 58, 69, 82, 92, 100</em></td>
<td>Past tense, see past participle</td>
</tr>
<tr>
<td><em>proposition, divided sense, 65</em></td>
<td>Proposition, divided sense</td>
</tr>
<tr>
<td><em>propositional aspect, 29, see also situation type</em></td>
<td>Propositional aspect</td>
</tr>
<tr>
<td><em>propositional attitudes, 64</em></td>
<td>Propositional attitudes</td>
</tr>
<tr>
<td><em>Pseudo-direct constructions, 31</em></td>
<td>Pseudo-direct constructions</td>
</tr>
<tr>
<td><em>Pullum, 86</em></td>
<td>Pullum, 86</td>
</tr>
<tr>
<td><em>Pustejovsky, 40-42, 49n, 82n, 94, 101, 121</em></td>
<td>Pustejovsky, 40-42, 49n, 82n, 94, 101, 121</td>
</tr>
<tr>
<td><em>put, 47, 52</em></td>
<td>Put, 47, 52</td>
</tr>
<tr>
<td><em>queries, 9</em></td>
<td>Queries, 9</td>
</tr>
<tr>
<td><em>Quine, 65</em></td>
<td>Quine, 65</td>
</tr>
<tr>
<td><em>Rappaport, 63</em></td>
<td>Rappaport, 63</td>
</tr>
<tr>
<td><em>rarely, 25</em></td>
<td>Rarely, 25</td>
</tr>
<tr>
<td><em>realize, 41</em></td>
<td>Realize, 41</td>
</tr>
<tr>
<td><em>reference point, 27, 29, 45, 50, 52, 68, 110</em></td>
<td>Reference point, 27, 29, 45, 50, 52, 68, 110</td>
</tr>
<tr>
<td><em>referring functions, 67</em></td>
<td>Referring functions, 67</td>
</tr>
<tr>
<td><em>Reichenbach, 4, 9, 27-29, 44n, 45-53, 57, 63, 68, 69, 74, 75, 93, 96, 105, 107, 110, 120</em></td>
<td>Reichenbach, 4, 9, 27-29, 44n, 45-53, 57, 63, 68, 69, 74, 75, 93, 96, 105, 107, 110, 120</td>
</tr>
<tr>
<td><em>repetition, 14, 25-26, 94, 96, 97, 109, 121, 124, 127</em></td>
<td>Repetition, 14, 25-26, 94, 96, 97, 109, 121, 124, 127</td>
</tr>
<tr>
<td><em>sense, 8</em></td>
<td>Sense, 8</td>
</tr>
<tr>
<td><em>Reyle, 29, 38, 40, 46, 48-54, 56, 57, 63, 82n, 83, 84, 93, 94, 103, 105, 107, 120, 121</em></td>
<td>Reyle, 29, 38, 40, 46, 48-54, 56, 57, 63, 82n, 83, 84, 93, 94, 103, 105, 107, 120, 121</td>
</tr>
<tr>
<td><em>roughly, 22</em></td>
<td>Roughly, 22</td>
</tr>
<tr>
<td><em>Ryle, 33</em></td>
<td>Ryle, 33</td>
</tr>
<tr>
<td><em>Sag, 86</em></td>
<td>Sag, 86</td>
</tr>
<tr>
<td><em>Scholastics, 65</em></td>
<td>Scholastics, 65</td>
</tr>
<tr>
<td><em>Schubert, 29, 73-75, 120</em></td>
<td>Schubert, 29, 73-75, 120</td>
</tr>
<tr>
<td><em>same, 34</em></td>
<td>Same, 34</td>
</tr>
<tr>
<td><em>segment</em></td>
<td>Segment</td>
</tr>
<tr>
<td><em>semantic aspect, 29, see also situation type</em></td>
<td>Semantic aspect, 29, see also situation type</td>
</tr>
<tr>
<td><em>sentence adverbial, 18</em></td>
<td>Sentence adverbial, 18</td>
</tr>
<tr>
<td><em>Serlet, 93</em></td>
<td>Serlet, 93</td>
</tr>
<tr>
<td><em>shall, 9, 15</em></td>
<td>Shall, 9, 15</td>
</tr>
<tr>
<td><em>Shapiro, 63</em></td>
<td>Shapiro, 63</td>
</tr>
<tr>
<td><em>Shieber, 86, 88, 133</em></td>
<td>Shieber, 86, 88, 133</td>
</tr>
<tr>
<td><em>should, 9, 15</em></td>
<td>Should, 9, 15</td>
</tr>
<tr>
<td><em>simple past, see past tense</em></td>
<td>Simple past, see past tense</td>
</tr>
<tr>
<td><em>simple present, see present tense</em></td>
<td>Simple present, see present tense</td>
</tr>
</tbody>
</table>
TEARS IN THE RAIN

since, 23, 24, 43, 104
sit, 2, 31
situation type, 2, 3, 27, 29–36, 45, 75, 82–85, 94, 121, 123, 130, 176
accomplishment, 34–38, 41, 45, 82n, 94, 95
achievement, 34, 35, 37, 38, 41, 45, 82n, 94, 95, 123
activity, 22, 34, 35, 37, 38, 40, 45, 82n, 94, 95
dynamic, 29, 31, 34, 37
even, 47–49, 52, 75, 82n
point, 34, 37, 38, 45, 82n, 94, 95
process, see activity, 94
stance, 31, 82n
state, 22, 36, 38, 40, 45, 47–49, 52, 75, 82n, 94, 95
stative, 29, 31
transition, 40, 82n, 94, 95
slowly, 31, 47
some, 21
Sowa, 63, 65, 68, 93
space builder, 64–65, 125
space case, see space builder
spaces, see partitioned representations
span of time, 22
speak, 7
speech point, 27, 68
stance, 31, 82n
stand, 31
start, 8, 26, 99
state, 22, 36, 38, 40, 45, 47–49, 52, 75, 82n, 94, 95
stative, 29, 31
stay, 8
Steedman, 29, 36–38, 40, 42, 123, 126, 127
Stoics, 43
stop, 8, 26, 36
subjunctive mood, 7, 10, 12, 19, 92
tea, 101, 121, 166
telic, 33
temporal logic, 42, 47
tense, 2, 3, 7, 16, 95, 124
‘future’, 7, 15, 17, 27, 50, 58, 69, 82, 95
past, 7, 11–14, 27, 50, 58, 69, 82, 92, 100
present, 7, 11–13, 15, 27, 50, 58, 69, 82, 92, 95, 124, 125
tense logic, 42, 43
tense logic, metric, 43
tense tree’, 73, 120
terminative aspect, 69
that, 21
thematic role, 6, 151
then, 21
Thomas Aquinas, 65
throughout, 23
full, 23
time
external, see external time
internal, see internal time
time adjunct, 2, 5, 14, 17, 19–21, 25, 29, 38, 40, 41, 47, 48, 50, 70, 82, 102–105, 109, 115, 123, 124
duration, 19, 21–23, 35, 37
frequency, 19, 25–26, 31, 124
location, 19, 21–23, 37
time point, 27–29, 44n, 46–48, 51, 106, 125, 169, 171, 177
time span, 22
time units, 20
timeless, 13, 15–17, 23n, 36, 43, 46n, 96, 102, 115, 124, 125
sled, 30
to, 23, 104
transition, 40, 82n, 94, 95
trundle, 98, 99, 150
trundle off, 99, 150, 171
try, 18
twice, 25, 36
type
action, see situation type
aspectual, 29, see also situation type
type-shifting, 2, 3, 34n, 36–42, 82–85, 124, 126, 127, 150, 176
units of time, 20
until, 23, 43, 104
ug, 26
use to, 25, 26, 124
used to, 9
utterance time, 29, 50, 52
van Benthem, 43, 44
van Eijck, 42, 48, 49
Vendler, 31, 34, 35, 37, 49, 75, 94
verb
aspectual, 2, 3, 8, 26, 47, 52, 59, 69, 70, 75, 82–84, 95, 99–100, 121, 124, 126, 176
auxiliary, 7–11, 26, 86, 88
finite, 7, 10, 11, 16
full, 7
main, 7, 9, 10
modal, 7, 9, 11, 15–16, 24, 28, 82, 89, 95
non-finite, 7, 10, 70, 82
verb aspect, 29, see also situation type
verb group
finite, 10, 11, 13–16, 18
non-finite, 10, 16–18, 124
verb particle, 2, 3, 26, 47, 83, 84, 98–99, 121, 176
verb phrase
non-finite, 59, 75, 82, 111, 121
Verkuyl, 35, 42
voice, 7, 8
passive, 8, 10, 11, 17, 18, 88, 89
Webber, 107
weekly, 25
Weinstein, 73
were, 10
were subjunctive, see past subjunctive mood
when, 23, 47, 104, 111, 124
whenever, 25
while, 23, 104
White, 42
Wiese, 63
Williams, 63–65, 93
will, 9, 15, 124, 125, 142
William of Ockham, 43
within, 23, 24
weak, 9, 15, 96
white, 95