CONTROLLED NATURAL LANGUAGE WITH TEMPORAL FEATURES

A thesis submitted to the University of Manchester for the degree of Doctor of Philosophy in the Faculty of Science and Engineering

2018

By
Ayoade O. Adeniyi
School of Computer Science
# Contents

- **Declaration**
  - 2
- **Copyright**
  - 3
- **Acknowledgements**
  - 4

## 1 Introduction

- **1.1 Motivation**
  - 6
- **1.2 Research Hypothesis and Questions**
  - 7
- **1.3 Thesis Outline**
  - 8

## 2 Technical Preliminaries

- **2.1 Language**
  - 10
- **2.2 Grammar**
  - 11
    - **2.2.1 Context-Sensitive and Context-Free Grammars**
      - 13
    - **2.2.2 Definite Clause Grammar**
      - 14
    - **2.2.3 Categorial Grammar**
      - 15
    - **2.2.4 Tree- adjoining Grammar(TAG)**
      - 17
    - **2.2.5 Dependency Grammar(DG)**
      - 18
    - **2.2.6 Discussion**
      - 19
- **2.3 Natural Language Semantics**
  - 20
    - **2.3.1 Discourse Representation Theory**
      - 21
- **2.4 Formal Interpretation of Natural Language Sentences**
  - 22
    - **2.4.1 Lambda Calculus**
      - 23
    - **2.4.2 Montague Semantics**
      - 25
- **2.5 Conclusion**
  - 25
3 Temporality in English

3.1 Temporal Ontology

3.1.1 Instant-Based Model of Time
3.1.2 Interval-Based Model of Time

3.2 Tense Logic

3.2.1 Branching Time Logic
3.2.2 Interval Based Temporal Logic

3.3 TimeML

3.4 Tense

3.4.1 Reichenbach’s Theory of Tense

3.5 Grammatical Aspect

3.5.1 Syntax of English Tense
3.5.2 Representation of Tenses with Temporal Intervals

3.6 Aspectual Classification

3.6.1 States and Events

3.7 Temporal Modifiers

3.8 Discussion

4 Controlled Natural Language

4.1 What is a Controlled Natural Language?

4.2 Applications of CNLs

4.3 Survey of Controlled Natural Languages

4.3.1 Attempto Controlled English (ACE)
4.3.2 PENG
4.3.3 Rabbit
4.3.4 Computer Processable Language (CPL)
4.3.5 E2V

4.4 Discussion

4.5 Summary

5 Temporal Modifiers

5.1 Corpus Analysis

5.2 Tensed Sentences

5.3 Subordinate Clauses and Phrases

5.4 Temporal Adverbials

5.5 Temporal Prepositions
5.5.1 During ..................................................... 69
5.5.2 The Locative Temporal Prepositions *In, On* and *At* ........ 74
5.5.3 Durative Prepositions *For* and *In* ................................. 78
5.5.4 Before and After ............................................. 83
5.5.5 By ............................................................ 87
5.5.6 Until .......................................................... 89
5.5.7 Since ......................................................... 92
5.6 Temporal Conjunctions ............................................... 94
5.6.1 Before and After ............................................. 95
5.6.2 Until .......................................................... 97
5.6.3 While ......................................................... 99
5.7 Discussion ...................................................... 100

6 Design of a CNL with Temporal Features .......................... 103
6.1 Lexicon .................................................................. 104
6.2 Grammar Rules ..................................................... 105
6.2.1 Determiners ..................................................... 106
6.2.2 Nouns and Noun phrases ....................................... 107
6.2.3 Tense ............................................................ 109
6.2.4 Verbs and Verb Phrases ......................................... 110
6.2.5 Temporal Modifiers ............................................. 115
6.2.6 Sentences ........................................................ 118
6.3 Semantics ........................................................... 119
6.3.1 Determiners ..................................................... 120
6.3.2 Noun and Noun phrases ....................................... 121
6.3.3 Tense ............................................................ 122
6.3.4 Verb and Verb Phrases ......................................... 123
6.3.5 Temporal Modifiers ............................................. 125
6.3.6 Sentences ........................................................ 126
6.4 Conclusion .......................................................... 127

7 Conclusions and Future Work .......................................... 129
7.1 Conclusions ........................................................ 129
7.2 Future Work ........................................................ 131

A Syntactic Analysis of Temporal Modifiers .......................... 133
List of Tables

3.1 Allen’s Thirteen Temporal Interval Relationships .................. 29
3.2 Time coordinates of Reichenbach’s tense .......................... 36
3.3 Dowty’s aspectual class test ............................................. 44

5.1 CFG categories and their symbols ................................. 67
5.2 Data of the Syntactic Structure of During Complements ........... 70
5.3 Data of the Tense, Aspect, Aspectual Class Configuration During Main Clauses ...................................................... 71
5.4 Data of the Syntactic Structures of In, On and At Complements . 76
5.5 Data of the Syntactic Structure of Before Non-sentential Complements .......................................................... 84
5.6 Data of the Syntactic Structure of After Non-sentential Complements .................................................. 85
5.7 Data summary of the syntactic structure of By complements ...... 88
5.8 Data of the Syntactic Structure of Until Non-sentential Complements .................................................. 90
5.9 Data of the Syntactic Structure of Before Sentential complements .................................................. 95
5.10 Data of the Syntactic Structure of After Sentential Complements .................................................. 95
5.11 Data of the Syntactic Structure of Until Sentential Complements .................................................. 98

6.1 POS and their Specific Features ........................................... 105
6.2 Determiners Features and Range of Values ....................... 107
6.3 Noun Features and range of Values ..................................... 108
6.4 Verb Features and Range of Values ..................................... 111
6.5 Auxiliary verb Annotations ................................................. 112

A.1 Syntactic Structure of Main clause of in Sentences ............... 133
A.2 Syntactic Structure of Main clause of on Sentences ............. 134
A.3 Syntactic Structure of Main clause of at Sentences .............. 134
A.4 Syntactic Structure of Main clause of by Sentences .............. 135
A.5 Syntactic Structure of Main clause of before Sentences ........ 135
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Simple Parse Tree</td>
<td>14</td>
</tr>
<tr>
<td>2.2</td>
<td>Derivation of simple English sentence using Lambek’s Elimination rules</td>
<td>16</td>
</tr>
<tr>
<td>2.3</td>
<td>α-trees</td>
<td>17</td>
</tr>
<tr>
<td>2.4</td>
<td>β-trees</td>
<td>17</td>
</tr>
<tr>
<td>2.5</td>
<td>Derived Tree</td>
<td>18</td>
</tr>
<tr>
<td>2.6</td>
<td>DG Graph</td>
<td>18</td>
</tr>
<tr>
<td>2.7</td>
<td>Semantic annotated parse tree</td>
<td>26</td>
</tr>
<tr>
<td>3.1</td>
<td>Deep Structure</td>
<td>40</td>
</tr>
<tr>
<td>3.2</td>
<td>Diagrammatic representation of aspectual class (adopted from Smith 1983)</td>
<td>42</td>
</tr>
<tr>
<td>5.1</td>
<td>Parse Tree of a simple past tensed sentence</td>
<td>63</td>
</tr>
</tbody>
</table>
Abstract

A controlled natural language (CNL) is a subset of a given natural language with explicitly regimented syntax and vocabulary such that member sentences are unambiguously translated to some formal logic. Initially created to serve as a tool for teaching non-native speakers of a given natural language, CNLs have more recently served as an interface between natural languages and formal languages. Quite a number of CNLs have been defined over the past two decades, these CNLs have varied domains of application such as software and hardware specification, ontology authoring and editing, air traffic control etc. Controlled natural languages have gradually gained some popularity in the real world.

From a survey of existing CNLs, we observe the lack of a controlled natural language such that member sentences contain temporal expressions. In order to be able to provide correct syntactic structures and corresponding semantic interpretations for these temporal expressions, we consider thousands of generated sentences analysing the various tense, aspect, aspectual class and temporal modifier configuration. We also analyse sentence extracts from the Brown corpus from which we are able to define semantic interpretations for the various temporal modifiers of interest, taking into consideration how these interpretations are affected by the sentence tense and aspect and aspectual class configuration.

From this analysis, we are able to provide syntactic rules defined with definite clause grammar. Natural languages are intuitive, evolving naturally without definitive grammars. Representing their syntactic structure with a definite clause grammar would require consideration of factors other than syntactic categories. In order to be able to cater for these other syntactic restrictions, for example, tense and number agreement, we define features and a range of values for each terminal and non-terminal symbol in our grammar. We can, therefore, define rules to satisfy these syntactic restrictions. We recursively generate the semantic interpretations of structures in our language applying Montague semantics.
Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.
Copyright

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the “Copyright”) and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works in the thesis, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=487), in any relevant Thesis restriction declarations deposited in the University Library, The University Library’s regulations (see http://www.manchester.ac.uk/library/aboutus/regulations) and in The University’s policy on presentation of Theses.
Acknowledgements

I most particularly grateful to God for affording me this opportunity and strenght to run my Ph.D program at a university as prestigious as the University of Manchester.

I am also very appreciative of the guidance given me by my supervisor – Dr. Ian Pratt Hartmann. His wisdom, patience and high level of attention to my requirements as his students cannot be over emphasized.

My parents Mr. Joshua Adeniyi and Mrs. Christianah Adeniyi are nothing but the best. Their words of encouragement helped me immensely in difficult times.

My siblings – Modupe, Fadeke and Toyin, I like to thank you for you numerous help and for always being there. I would also like to thank Debola for her helping hand and her care.
Chapter 1

Introduction

For the purposes of this thesis, controlled natural languages (CNLs) are computer processable subsets of natural languages with explicitly regimented syntax and vocabulary such that member sentences are unambiguously translated to some formal logic. Temporal expressions are natural language expressions that provide information such as “when” and “how long” an event or state of affairs holds. This thesis proposes a controlled natural language with temporal features.

Controlled natural languages can be considered a form of language that combines the understandability of natural languages and the strict structure and unambiguity of formal languages. They hence serve as interfaces for formal languages and have been applied to various domains, for example, hardware and software specification, ontology authoring and querying, etc.

This thesis explores the development of a controlled natural language with temporal features such that member sentences contain temporal expressions – tense, aspects and temporal modifiers.

The rest of this chapter is organised as follows. Section 1.1 discusses the current domain of controlled natural languages and identifies the motivation for a controlled natural with member sentences containing temporal expressions – Controlled natural language with temporal features. In section 1.2 we establish a research hypothesis and proffer corresponding research questions. Section 1.3 provides the outline for the rest of the thesis.
CHAPTER 1. INTRODUCTION

1.1 Motivation

Aristotle’s syllogisms which contain only a few sentence forms are perhaps the earliest known controlled natural language. Another early controlled natural language is Basic English ([Ogden (1933)]) was developed as an auxiliary language to aid teaching English to non-native speakers. More recently, controlled natural languages have been designed to be computer processable subsets of a given natural language such that member sentences are unambiguously translated into some formal logic, these later CNLs unlike Aristotle’s syllogisms and Basic English have more expressive sentence forms and corresponding semantic interpretations.

Given that CNLs have become considerably more expressive in recent times, they have been applied to a number of real-life problems. A popular domain of application is software/hardware specification authoring, for example, ACE ([Fuchs et al. (1999)]) and PENG ([Schwitter (2002)]). Another area of application is ontology authoring and query, for example, Rabbit ([Hart et al. (2008)]). Other areas of application include tax fraud detection ([Calafato et al. (2016)]), proof checking mathematical text ([Cramer et al. (2010)]), speech recognition interfaces ([Kaljurand, Alumäe (2012)]).

Temporal expressions — that is, tense, aspects and temporal modifiers are used in our day to day communication to provide temporal information such as “when”, “for how long” and “how often” an event or state of affairs holds, these temporal expressions in interaction with verbs of different aspectual classes describe varied temporal relationships. There are also formal languages that have been adapted to provide temporal information and relationships, for example, basic tense logic ([Prior (1957)]), interval temporal logic ([Halpern, Shoham (1991)]), first-order temporal interval logic, etc. However, despite the regular use of temporal expressions in our daily communications, the advancement in the expressiveness of controlled natural languages, and the existence of formal languages adapted to express temporal information and relationships, there has been little efforts in developing a controlled natural language with temporal features.

Generally, controlled natural languages are developed in response to some real-world problems. We, therefore, need to consider possible applications for a controlled natural language with temporal features. Take for example a controlled natural language such as Attempto Controlled English ([Fuchs et al. (1999)]) which is applied to software specification authoring. It is easy to think of a specification that describes events in sequence or events occurring concurrently. At the
moment, ACE provides event order based on the order of occurrence in the input sentence. Temporal modifiers, however, provide more expressive and intuitive ways of providing temporal relationships such as temporal order, temporal containment, etc. Furthermore, ontology authoring has been a popular domain of application for CNLs. A controlled natural language with temporal features can be applied as interface for OWL-Time (Cox, Little (2017)) – that is, time ontology which is a knowledge representation language for temporal information.

In order to be able to develop a controlled natural language with temporal features, we require syntactic representation and corresponding semantic interpretation for the temporal expressions, the effect they have on verbs of different aspectual classes and how they behave when they co-occur in a natural language sentence.

This thesis, therefore, proposes the development of a controlled natural language with temporal features where member sentences contain temporal expressions – tense, aspect and temporal modifiers. The corresponding semantic interpretation of member sentences express the effect of the co-occurrence of said temporal expressions.

### 1.2 Research Hypothesis and Questions

The purpose of this thesis is to propose a syntactic representation and corresponding semantic interpretation for English sentences with temporal expressions for a controlled natural language with temporal features. This approach is based on the following hypothesis.

- A controlled natural language grammar with temporal features can sufficiently represent temporal information and relationships expressed by temporal expressions – tense, aspect, aspectual class and temporal modifiers.

This hypothesis is tested based on the following questions.

1. *How do we determine what combinations of temporal expressions are grammatical?*

2. *What effect does the co-occurrence of temporal expressions have on the semantic interpretation of a given sentence?*
3. How can the syntactic behaviour and semantic interpretations of the combination of the various temporal expressions be defined in a controlled natural language?

1.3 Thesis Outline

Chapter 2 provides an overview of tools required for the definition of a controlled natural language. We begin by defining languages – formal and natural. After which we discuss formal grammars, presenting existing options in order to decide on which is most suitable for the syntactic representation of sentences of a controlled natural language. Some of the formal grammars discussed include – context-free grammar, context-sensitive grammar, definite clause grammar, tree-adjoining grammar, dependency grammar, etc. In the same chapter, we compare three formal languages which could be used for the semantic interpretation of sentences of CNL; these formal languages include, first-order logic, propositional logic, and discourse representation structure. Finally, we consider the automatic translation of natural language sentences to first-order logic with the aid of lambda-calculus and Montague semantics.

Chapter 3 considers temporal expressions in natural language. We define and provide the properties of the two models of time – instant-based and interval-based models. We further discuss a few formal logics that are based on these models, for example, basic tense logic, branching time logic, interval-based temporal logic. We present a brief description of TimeML – a temporal information annotation language. After which we proceed to describe various theories of the semantics of tense and grammatical aspects. Chapter 3 also discusses the aspecual classes of verbs, their structure and how they interact with tense and aspect. Finally, we briefly consider temporal modifiers outlining the effect tense, aspect, aspecual class, and temporal modifier can have on the grammatical correctness and semantic interpretation of a given natural language sentence.

In Chapter 4 we provide a more detailed definition of controlled natural languages, this include a brief history of CNLs and how they have evolved. We also discuss applications of CNLs; examples of some these fields include software/hardware specification authoring, ontology authoring, and query, business
contract development, proof checking mathematical texts, etc. Chapter 4 provides a survey of a few CNLs; we briefly discuss their syntax and semantic interpretations and applications. Finally, we present a brief description of what sentence forms of a controlled natural language with temporal features and what it entails to design such a language.

Chapter 5 provides syntactic analyses and semantic interpretations for temporal modifiers. We present the syntactic structure of simple tensed sentences and their corresponding semantic interpretation. We also show the semantic interpretation of subordinate clauses, highlighting how they are interpreted differently from simple tensed sentences. We distinguish three types of temporal modifiers based on their syntactic structures, namely – temporal adverbials, temporal prepositions, and temporal conjunctions. Having observed the syntactic structures of these types of temporal modifiers, we provide syntactic rules for each temporal modifier, noting the tense, aspect, aspectual clause configuration of the main and subordinate clauses or the type of nouns they are complemented with as applicable.

Chapter 6 presents the grammar rules as well as the semantics of a controlled natural language with temporal features. First, we describe the lexicon of our language, describing the features and range of values required for defining lexical items of the various parts of speech featured in our language. We proceed to describe how various syntactic categories are generated from lexical items and other syntactic categories. We highlight the features required by each syntactic category and the conditions required for their generation. We provide the semantics our language, providing rules governing the generation of the semantic interpretation of various syntactic structures from the semantic interpretation of their constituents.

Chapter 7 draws concluding remarks on the contributions of this thesis providing answers to the research questions raised in this chapter.
Chapter 2

Technical Preliminaries

Controlled Natural Languages are simply natural languages with regimented syntax, lexicon and semantic interpretation designed in order to reduce or exclude ambiguity. We are however not only interested in the syntactic representation of our proposed language, but we also aim to provide appropriate unambiguous semantic interpretations for member sentences of the language. In order to achieve this, we need to consider a number of required techniques and tools.

We often describe a natural language utterance as grammatical or not, but many native speakers of a given natural language are not necessarily aware of the role grammar plays in the language definition. In this chapter, we consider the meaning and definition of a language in section 2.1. Section 2.2 provides a brief description grammar. We consider a few examples of formal grammar in this section. In section 2.3 we consider the various formal languages that can be used in expressing meaning. Section 2.4 shows how we can generate the semantic interpretation of sentences from the interpretation of their syntactic constituents, hence providing a systematic technique for providing semantic interpretations for sentences of a controlled natural language.

2.1 Language

The term language according to the Oxford Dictionary is the method of human communication, either spoken or written, consisting of the use of words in a structured and conventional way. The dictionary description of language can be considered rather inadequate particularly for formal languages. Formally, a language is any set of sentences over an alphabet or vocabulary. A vocabulary
is a finite set of symbols, for example, the Greek alphabet \( \{ \alpha \ldots \omega \} \), the Latin alphabet \( \{ a, b, \ldots, z \} \), the binary alphabet \( \{0,1\} \), etc. We can, therefore, define a natural language as a set of sentences over a vocabulary containing words of the said natural language.

As speakers of the English language, we have an inherent idea of how grammatical sentences are constructed. Although this ability is intuitive, formal languages have mathematically defined rules – grammars that determine correct sentences of the said language. Due to the vagueness of natural languages, they do not have definitive grammars like formal languages do.

Controlled natural languages combine the precision and strictness of formal languages with the intuitiveness and immediate comprehensibility of natural languages. We, therefore, consider some of the formal grammars that can be employed in the definition of the syntax of controlled natural languages.

## 2.2 Grammar

Having discussed languages in the previous section, we describe in this section how their syntax is defined. Grammar is a set of rules responsible for the producing sentences of a given language. Initially a linguistic concept primarily used for the representation of natural languages, grammars have been more recently applied to the development of formal and computer languages. Grammars provide rules that define what is or is not a valid sentence of a given language as well as providing structural descriptions of the said language’s sentences. One of the aims of having formal grammar representation for natural languages is to enable machine processing. Unfortunately, we do not have a definitive grammar for natural languages. Computer/formal languages, on the other hand, have well-defined grammars.

Formally as defined by [Chomsky (1956)](https://www.chomsky.info/), we denote a grammar as a tuple as given in \( (1) \)

\[
G = (V_N, V_T, P, S),
\]

where \( V_N \) is a set of non-terminal symbols which are also known as syntactic categories; \( V_T \) is the set of terminal symbols which are morphosyntactic categories and/or words in the language’s vocabulary; \( P \) is a set of production rules which show the relationship between various strings of non-terminal and terminal symbols; and \( S \) is the start symbol such that \( S \in V_N \).
CHAPTER 2. TECHNICAL PRELIMINARIES

It is assumed that $V_N$ and $V_T$ have no common elements, that is, $V_N \cap V_T = \emptyset$. The production rules $P$ are expressions of the form $\alpha \rightarrow \beta$, where $\alpha$ (the head of the production rule) is a string in $V^+$ (where $V^+$ is any string from the set $V_T \cup V_N$ symbols except for the empty string $\epsilon$) and $\beta$ (the body of the production rule) is a string in $V^*$ (where $V^*$ is any string from the set of $V_T \cup V_N$ symbols).

Given a grammar $G$, we need to provide a definition for the language it generates. In order to do that we need to provide a few definitions first: If $\alpha \rightarrow \beta$ is a production rule, $\gamma$ and $\delta$ are strings in $V^*$, then $\gamma\alpha\delta \xrightarrow{G} \gamma\beta\delta$ means the production $\alpha \rightarrow \beta$ is applied to string $\gamma\alpha\delta$ to obtain $\gamma\beta\delta$. Thus $\xrightarrow{G}$ relates two strings when the second is obtained from the first by the application of a single production. Assuming $\alpha_1$, $\alpha_2$, $\alpha_3$, ..., $\alpha_m$ are strings in $V^*$, and $\alpha_1 \xrightarrow{G} \alpha_2$, $\alpha_2 \xrightarrow{G} \alpha_3$, ..., $\alpha_{m-1} \xrightarrow{G} \alpha_m$. Then we say $\alpha_1 \xrightarrow{\ast} \alpha_m$, meaning we obtain $\alpha_m$ from $\alpha_1$ by applying a number of production rules. We therefore define the language $L(G)$ in (2) generated by the grammar $G = \{w \mid w \in V^* \text{ and } S \xrightarrow{G} w\}$, where $w$ is a string or sentence in $L(G)$ if the string consists solely of terminals ($V_T$) and the string can be derived from $S$. The language of $G$ is a set of terminal strings that have the head of their production rules as the start symbol;

(2) $L(G) = \{v \subset V_T^* \mid S \xrightarrow{G} v\}$

We say grammars $G_1$ and $G_2$ are equivalent if $L(G_1) = L(G_2)$.

Chomsky [1959] described a theory of language as containing a specification $F$ of functions from which grammars may be derived. He proposed restrictions with different strengths which can be placed on the specification $F$; this leads us to the concept of types of grammar (Chomsky hierarchy).

From the type 0 grammar otherwise known as the unrestricted grammar, Chomsky defined three levels of restrictions which produce three types of grammar namely the type 1, type 2 and type 3 grammars, these types of grammar are otherwise known as the context-sensitive grammars, context-free grammars, and regular expression grammars respectively. Chomsky, however, suggested the context-free grammars and context-sensitive grammars are most suitable for the syntactic representation of languages.
2.2. Grammar

2.2.1 Context-Sensitive and Context-Free Grammars

Context-free and context-sensitive grammar are defined as tuples as seen in (1), where \( V_N \) is a finite set of non-terminal symbols or syntactic categories, \( V_T \) is a finite set of terminal symbols, \( P \) is the finite set of production rules. Production rules of context-free grammar are specifically required to have exactly one non-terminal symbol in their head and zero to many terminal and/or non-terminal symbols in their body (i.e. \((V_N \cup V_T)^*\)), for example, (3).

\[
3 \quad A \rightarrow \alpha_1, \beta
\]

The production rule of context-free grammars do not require left or right-contexts. Context sensitive grammar, in contrast, has production rules of the form

\[
4 \quad \alpha_1 A \alpha_2 \rightarrow \alpha_1 \beta \alpha_2,
\]

where \( \beta \neq \epsilon \). Variable \( A \) is only permitted to be replaced by string \( \beta \) given the right-context \( \alpha_1 \) and left-context \( \alpha_2 \).

Parse Trees

Production rules \( P \) can be represented graphically with the aid of parse trees. Given a grammar \( G \) defined as in (1), where the tree leaves are nodes without any children, the parse trees of \( G \) must have each node labelled by a member of \( V_N \), except for the leafs which are labelled with either the empty string \( \epsilon \) or a terminal symbol \((V_T)\). Given a node labelled \( Y \), and its children are labelled \( X_1, X_2, \ldots, X_n \) from left to right respectively, the production rule for the said node is given as \( Y \rightarrow X_1, X_2, \ldots, X_n \). Consider the following sentence.

\[
5 \quad \text{John loves Mary}
\]

Given a sentence such as (5), from basic English grammar we know the sentence consists of the categories noun phrase followed by a verb phrase. We define a context-free grammar in (6) to generate sentence (5) with its corresponding parse tree given in Figure (2.1).

\[
6 \quad S \rightarrow NP, VP \\
NP \rightarrow PN \\
VP \rightarrow V, NP
\]
CHAPTER 2. TECHNICAL PRELIMINARIES

$PN \rightarrow $ John
$PN \rightarrow $ Mary
$V \rightarrow $ loves

There are however other grammars that are capable of representing the syntactic structure of language sentences. Some of these grammars include definite clause grammar, categorial grammars, tree-adjoining grammar, dependency grammar, etc. There has been sufficient research on these formalisms providing information about relationships between them such as equivalence. Before we consider possible relationships between some of these grammars, let us briefly introduce how they work.

2.2.2 Definite Clause Grammar

Definite clause grammar is a formalism in logic programming languages such as Prolog and Mercury, such that production rules are expressed in first-order definite clauses. The context-free grammar in (6) can be expressed in Prolog clauses as seen below.

(7) $s(A,Z) :- np(A,X), \text{vp}(X,Z).$
    $np(A,Z) :- pn(A,Z).$
    $\text{vp}(A,Z) :- v(A,X), np(X,Z).$
    $pn([\text{john}|X],X).$
    $pn([\text{mary}|X],X).$
    $v([\text{loves}|X],X).$

where the functors of the terms are the non-terminal symbols in the grammar, the arguments are variables representing the string of terminal symbols that make
2.2. GRAMMAR

up the non-terminal symbol of interest and the definite clauses govern symbols that are required to combine to form the head symbol. Prolog however provides a syntactic sugar for normal Prolog definite clauses. The definite clauses in (7) are therefore rewritten as follows.

(8) \[ s \rightarrow np, vp. \]
    \[ np \rightarrow pn. \]
    \[ vp \rightarrow v, np. \]
    \[ pn \rightarrow [john]. \]
    \[ pn \rightarrow [mary]. \]
    \[ v \rightarrow [loves]. \]

The simple definite clause grammar in (8) can be considered the Prolog representation of a context free grammar. We are, however, able to add generative power by providing arguments to rules (8) as seen below.

(9) \[ s(C) \rightarrow np(A), vp(B). \]
    \[ np(A) \rightarrow pn(A). \]
    \[ vp(Z) \rightarrow v(X,Z), np(Y), \{X=transitive, (Y=singular; Y=plural)\}. \]
    \[ pn(Y) \rightarrow [john]. \]
    \[ pn(singular) \rightarrow [mary]. \]
    \[ v(transitive, singular) \rightarrow [loves]. \]

Rule (9) allows sub-categorization of the non-terminal symbols. We can hence enforce linguistic restrictions such as number agreement, etc. Extra conditions are places in curly brackets which are required to be satisfied before the production rule can be applied, for example, rule (9) generates a verb phrase from the combination of a verb and a noun phrase provided the constituent symbols have the same number (i.e. both are either singular or plural) and the verb is transitive.

2.2.3 Categorial Grammar

There are various forms of categorial grammar, for example, bi-directional categorial grammar, combinatory categorial grammar, etc. In this section introduces bi-directional categorial grammar proposed by Bar-Hillel et al. (1960) (we sometimes refer to this grammar as AB-grammar). The basic idea behind AB-grammar is that the syntactic category of a given lexical item is determined by
what it concatenates with on its right and on its left to form another syntactic
category. AB-grammar, therefore, has the following rules.

1. \( \alpha \Rightarrow \alpha/\beta, \beta \)

2. \( \alpha \Rightarrow \beta, \beta \setminus \alpha \)

The first rule means, given an alphabet or string with category \( \alpha/\beta \), it requires
concatenation to its right with an alphabet or string of category \( \beta \) to generate
a string with category \( \alpha \). The second rule means an alphabet or string with
category \( \beta \setminus \alpha \) requires concatenation to its left with another alphabet or string
of category \( \beta \) to produce a string of category \( \alpha \). The above rules are called
reduction rules.

[1958] presents AB-grammar rules using the Gentzen style sequent
presentation. Consider Lambek sequent rules.

\[
\begin{align*}
(10) & \quad X \Rightarrow X & \text{(axiom)} \\
(11) & \quad Z \Rightarrow \Delta X \Delta' \quad X \Rightarrow \Gamma \\
& \quad Z \Rightarrow \Delta \Gamma \Delta' & \text{(Cut)} \\
(12) & \quad Y \Rightarrow X \Gamma \\
& \quad X \setminus Y \Rightarrow \Gamma & \text{(/⇒)} \\
\end{align*}
\]

where \( X, Y \) and \( Z \) are types, \( \Gamma \) and \( \Delta \) are sequences of types and \( \Rightarrow \) relates a
type with a string of types it is generated from.

Figure 2.2 provides the syntactic representation of sentence (5) in the sequent
format. Figure 2.2 illustrates the right and left elimination rules of Lambek

\[
\begin{align*}
\text{John} & \Rightarrow \text{loves : NP \setminus S/ NP} \\
\text{Mary} & \Rightarrow \text{loves : NP \setminus S} \\
& \Rightarrow \text{Mary : NP} \\
\text{John loves Mary} & \Rightarrow \text{loves : NP \setminus S} \\
& \Rightarrow \text{John : NP} \\
& \Rightarrow \text{S} \\
\end{align*}
\]

Figure 2.2: Derivation of simple English sentence using Lambek’s Elimination
rules.

calculus. It produces a derivation tree that shows how a lexical items concatenates
to form a sentence based on their individually assigned syntactic category.

Lambek calculus is otherwise known as type-logical grammar. [1993] proved that context-free grammar and type-logical grammar generate the same
set of languages.
2.2.4 Tree-adjoining Grammar (TAG)

Tree-adjoining Grammar is a tree generating system consisting of a number of elementary trees as opposed to context free grammar which is a string generating system consisting of production rules. The set of elementary trees in tree-adjoining grammars is divided into two, namely, initial (α) trees and auxiliary (β) trees. Suppose we are given a sentence such as (5), we represent it in tree-adjoining Grammar as given in Figure 2.3, where ↓ represents substitution. We

\[
\text{S} \quad \text{NP} \quad \text{NP} \\
\text{NP} \downarrow \text{VP} \quad \text{john} \quad \text{mary} \\
\text{loves} \quad \text{NP} \downarrow
\]

Figure 2.3: α-trees

can intuitively observe the similarity between CFG production rules in (6) and the subtrees in Figure 2.3. However, Figure 2.3 does not quite totally describe the tree-adjoining Grammar, it can rather be referred to as tree-substitution Grammar (TSG). It is worth mentioning however, that TSGs and CFGs generate the same language.

Suppose we attempt representing sentence (14) with the TSG.

(14) John really loves Mary,

we are unable to provide the tree structure for the given sentence with the grammar given in Figure 2.3. Joshi et al. (1975) therefore introduced the β-trees to deal with such instances. Consider Figure 2.4.

\[
\text{VP} \quad \text{really} \quad \text{VP}* \\
\]

Figure 2.4: β-trees

Auxiliary β-trees as in Figure 2.4 are required to have a leaf with the same non-terminal symbol as the root annotated with an asterisk symbol. These β-trees enable us to suspend the nodes in the α-tree with the same symbol as the β-tree root (in this case the VP node), replacing it with the β-tree and then reintroducing the previously suspended subtree as seen in Figure 2.5.
Tree-adjoining grammar is considered a stronger grammar than the context-free grammar, that is, it is able to represent the syntax of more complex languages than CFGs. It is sometimes classed as a mildly context-sensitive grammar. Although this formalism appears relatively easy to understand, computation is not as easy.

### 2.2.5 Dependency Grammar (DG)

The basic idea behind the dependency grammar as proposed by [Tesnière (1959)] is as follows. Given a sentence, every lexical item except one in the said sentence is dependent on another lexical item; the lexical item which does not depend on any other item is called the root of the sentence. In most cases, this root is the main verb of the sentence. Consider the following sentence.

(15) Every student spoke angrily.

The sentence verb *spoke* acts as the root and therefore depends on nothing. The determiner *Every* depends on the noun *student*, the noun *student* and the adverb *angrily* depend on the root. Figure 2.6 provides a graphical representation of the dependency relationship among the lexical items in sentence (15), where each edge represents dependency of the child(ren) on the parent.

![DG Graph](image)
Robinson (1970) proposed four axioms to govern well-formed dependency structures:

1. Strictly one element is independent
2. All other elements depend directly on some element
3. No element is dependent directly on more than one other element
4. If A depends on B and some element C intervenes between them, then C depends directly on A or B or some other intervening element.

The first three axioms are self-explanatory, the fourth axiom, otherwise called the requirement of projectivity, prevents crossing edges in dependency trees.

Hays (1964) and Gaifman (1965) provided a formal definition for the dependency grammar. A dependency grammar is defined as

\[
G = \langle R, L, C, F \rangle,
\]

where \( R \) is a set of dependency rules over the auxiliary symbols \( C \), \( L \) is a set of terminal symbols (vocabulary), \( C \) is a set of auxiliary symbols, \( F \) is an assignment function which assigns terminal symbols to categories. Hays and Gaifman’s definition of dependency grammar complies with Robinson’s axioms. From the above definition of dependency grammar, is its possible to prove that CFGs and DGs generate the same set of languages.

2.2.6 Discussion

In this chapter, we introduced various grammars and showed how they can be applied in providing the syntactic representation for natural language sentences. Grammars in the Chomsky hierarchy can be said to provide standards by which other grammars are measured. Bi-directional categorial grammar, tree-substitution grammar, and Dependency grammar are known to generate the same language as the context-free grammar. Definite clause grammar is able to express both context-free and context-sensitive grammar, and tree-adjoining grammar is considered mildly context-sensitive.

There are linguistic factors to consider in deciding which of the grammar formalisms discussed thus far is most suitable for the syntactic representation of sentences of a controlled natural language with temporal features. One of these
factors is number agreement. English sentences generated from the combination of a noun phrase and a verb phrase require number agreement between both constituent structures before a sentence can be generated. Other linguistic factors include tense agreement, sub-categorization of verbs to transitive and intransitive verbs, etc.

Context-free grammar by definition is not concerned with the syntactic structure of other categories nearby. As a result, we are unable to enforce number and tense agreement before applying a particular context-free production rule. Tree substitution grammar and dependency grammar are known to generate the same language as the context-free grammar and hence are thought by some linguist to be unable to represent the syntactic structures of natural language sentences sufficiently.

Context-sensitive grammar, on the other hand, determines whether or not a symbol is appropriate depending on the structure of surrounding symbols. We can, therefore, enforce number and tense agreement, provide rules for generating verb phrases from the combination of verbs and noun phrases provided the verb is transitive, etc.

Tree-adjoining grammar is considered mildly context-sensitive. It, however, has the drawback of being difficult to implement. Definite clause grammar, on the other hand, was developed as an expression for grammars in logic programming. Given that DCG has higher generative power than CFG’s by providing arguments, hence allowing the enforcement of agreements and other linguistic factors and it can easily be implemented in Prolog, DCG appears to be the most suitable grammar for the syntactic representation of a controlled natural language with temporal features.

2.3 Natural Language Semantics

We often tend to represent the meaning of natural language sentences with the aid of some formal language. One such formal language is propositional logic. Propositional logic is interested in the truth or falsity of a given proposition without consideration of the internal structure of the said proposition. From our discussion of the various grammar formalisms, it is quite obvious that the internal structure of sentences is of considerable interest. We, therefore, require a different formal language that considers the internal structure of sentences. An example
of such language is first-order logic. Given therefore a sentence such as

(17) Every boy loves some girl,

we interpret (17) as follows in first-order logic,

(18) ∀x(boy(x) → ∃y(girl(y) ∧ love(x, y))).

Unlike propositional logic, first-order formulas like (18) consider the internal structure of the sentence such that we can quantify existentially or universally over variables which represents object. Predicates and logical symbols provide information about the relationships between variables.

There are however linguistic/philosophical issues to observe before we can consider first-order logic as being adequate for the interpretation of natural language sentences. One of these issues include, the interpretation of time and temporal expressions. Although there are many theories on how temporal expressions should be represented, there is still not an agreed-upon approach. One of the central questions on the interpretation of temporal expression is whether temporal variables should represent time instants or time intervals.

Another issue of interest is predicate adverbials, for example, given a sentence such as

(19) John is walking quickly,

we are unable to represent verbal modifiers like quickly in first-order logic.

Finally, while first-order logic can provide interpretations for sentences with anaphoric references, given an already quantified variable, it is difficult to include an anaphoric reference from other sentences contained in a discourse. Discourse representation theory (DRT), a variant of first-order logic is able to manage anaphoric references contained in discourses sufficiently.

2.3.1 Discourse Representation Theory

Discourse Representation Theory was proposed by Kamp (1981); it is excellent in handling linguistic and logical issues such as anaphora, conditionals, and quantification. DRT is among a family of semantic frameworks known as dynamic semantics. DRT provides interpretations for discourses and not just simple sentences in what is referred to as a Discourse Representation Structure (DRS).
Informally a DRS is made up of two parts – a universe of discourse referents which represents the object of the said sentences or set of sentences, and a set of DRS conditions that encode information accumulated from the member sentences of the discourse of interest. Therefore given a simple sentence such as

(20) A farmer chased a donkey,

we represent the DRS thus:

(21) \[x, y : \text{farmer}(x), \text{donkey}(y), \text{chased}(x, y)]\]

where \(x\) and \(y\) are the discourse referents and \{\text{farmer}(x), \text{donkey}(y), \text{chased}(x, y)\}\ is the set of DRS conditions from the given sentence. Supposed we have sentence (20) followed by verb phrase (22).

(22) caught it.

Without the context of the previous sentence, we represent (22) as (23)

(23) \[v, w : \text{caught}(v, w)]\]

Placing sentence (22) within the context of sentence (20), we have the DRS (24).

(24) \[x, y : \text{farmer}(x), \text{donkey}(y), \text{chased}(x, y), \text{caught}(x, y)]\]

where the variables \(v\) and \(w\) have been substituted with the \(x\) and \(y\) as they are anaphors of \text{farmer} and \text{donkey} respectively.

### 2.4 Formal Interpretation of Natural Language Sentences

We have observed from section 2.2 that grammars are primarily responsible for providing syntactic rules to govern the generation of correct sentences of a given language. However, when given a sentence of a particular language, we are not only concerned about the grammatical correctness and structure of the given expression, but its meaning as well.

We concern ourselves in this section with tools providing formal semantic interpretations for sentences. From the discussion on natural language semantics above it appears first-order logic has the expressive power to handle this requirement. First order logic is a language in itself and possesses all the properties of
2.4. FORMAL INTERPRETATION OF NATURAL LANGUAGE SENTENCES

a language as enumerated in section 2.1. We, therefore in this section, provide
an overview of a system that enables us to assign semantic interpretations to
sentences of a given natural language from the interpretation of their constituent
categories. We begin by discussing the mathematical system Lambda calculus.

2.4.1 Lambda Calculus

Lambda calculus is a system developed by Church (1940). Lambda calculus
enables us to apply functions to arguments. For example, supposed we are given
a polynomial as in (25), and we want to compute the value of the expression if
\( x = 4 \).

\[ (25) \quad x^2 + 4x - 5 \]

We can apply lambda calculus by turning the polynomial into a lambda term as
seen below:

\[ (26) \quad \lambda x[x^2 + 4x - 5] \]

We then apply it to the argument thus:

\[ (27) \quad \lambda x[x^2 + 4x - 5](4) = 4^2 + 4.4 - 5 = 27 \]

A lambda term is therefore defined as follows (adapted from Carpenter (1997),
DEFINITION

**DEFINITION 2.4.1** TYPES: from a nonempty set \( \text{BasTyp} \) of basic types, the
set \( \text{Typ} \) of types is the smallest set such that

1. \( \text{BasTyp} \subseteq \text{Typ} \),

2. \( (\sigma \rightarrow \tau) \in \text{Typ} \) if \( \sigma, \tau \in \text{Typ} \)

Types in the form \( \sigma \rightarrow \tau \) are called the functional types, which are elements
which map objects of type \( \sigma \) to objects of \( \tau \). Functional types can be represented
in other forms for example, the above functional type can be represented as a
tuple:\( \langle \sigma, \tau \rangle \).

**DEFINITION 2.4.2** \( \lambda \)-term: For every type \( \tau \), we have the following sets

1. \( \text{Var}_\tau \): infinite set of variables of type \( \tau \)

2. \( \text{Con}_\tau \): a collection of constants of type \( \tau \)
DEFINITION 2.4.3 The collection of Terms$\tau$ of $\lambda$-terms of type $\tau$ are defined as the smallest set such that

1. $\text{Var}_\tau \subseteq \text{Term}_\tau$,

2. $\text{Con}_\tau \subseteq \text{Term}_\tau$,

3. $(\alpha(\beta)) \in \text{Term}_\tau$ if $\alpha \in \text{Term}_{\sigma\rightarrow\tau}$ and $\beta \in \text{Term}_\sigma$,

4. $\lambda x.(\alpha) \in \text{Term}_\tau$ if $\tau = \sigma \rightarrow \rho$ and $x \in \text{Var}_\sigma$ and $\alpha \in \text{Term}_\rho$

A term of the form $\alpha(\beta)$ is called a functional application of $\alpha$ to $\beta$. For example, if an expression $\text{walk}(\text{esther})$ is of the type $\text{Bool}$, $\text{esther}$ is of the type $\text{Ind}$, then $\text{walk}$ will be of the type $\text{Ind} \rightarrow \text{Bool}$, that is, it is a function from individuals to Boolean.

A term of the form $\lambda x.(\alpha)$ is a functional abstraction. As explained in the previous paragraph, application always involves a functional type, while abstraction always produces a functional type. Lambda abstraction is governed by the following axiom schemes.

DEFINITION 2.4.4 Axioms for $\lambda$-calculus

1. $\vdash \lambda x.\alpha \Rightarrow \lambda y.(\alpha[x \mapsto y])(\alpha\text{-reduction})$

2. $\vdash (\lambda x.\alpha)(\beta) \Rightarrow \alpha[x \mapsto \beta](\beta\text{-reduction})$

3. $\vdash \lambda x.(\alpha(x)) \Rightarrow \alpha(\eta\text{-reduction})$

Given therefore a lambda-abstracted formula as seen below,

(28) $\lambda Q[\forall x(\text{girl}(x) \rightarrow Q(x))]$

we assign the lambda-abstracted variable in [28] the semantic type $((\text{Ind} \rightarrow \text{Bool}) \rightarrow \text{Bool})$—that is, it is a function from a function from individuals to Boolean to Boolean. In order to apply $\beta$-reduction we need to apply the lambda-term in (28) to an lambda-term of type $\text{Ind} \rightarrow \text{Bool}$, that is, a function from individuals to Boolean, for example,

(29) $\lambda y[\text{human}(y)]$. 

We show in (30) the application of the lambda-term in (28) to the lambda term in (29).
2.5. **CONCLUSION**

\[
\lambda Q[\forall x (\text{girl}(x) \to Q(x))](\lambda y [\text{human}(y)]) = \\
\forall x (\text{girl}(x) \to \lambda y [\text{human}(y)](x)) = \\
\forall x (\text{girl}(x) \to \text{human}(x))
\]

Semantic types of the form \text{Ind} \to \text{Bool} can be simply written as \((\text{Ind}, \text{Bool})\).

### 2.4.2 Montague Semantics

Montague semantics allows us to provide a semantic interpretation for a given syntactic category, from the semantic interpretation of its constituent categories. We can, therefore, define semantic rules that enable the generation of the semantic interpretation of a sentence from the semantic interpretation of its constituents.

Suppose we modify the grammar in (30) so that non-terminal symbols are annotated with their semantic interpretations, and we define semantic rules that show how the semantic interpretation of non-terminal symbols are generated from the semantic interpretation of their constituent categories as in (31). We can hence recursively generate the semantic interpretation of a given sentence from the semantic interpretation of its syntactic constituents.

\[
(31) \quad \text{S} / \varphi(\psi) \to \text{NP} / \varphi, \text{VP} / \psi \\
\text{NP} / \varphi \to \text{PN} / \varphi \\
\text{VP} / \varphi(\psi) \to \text{TV} / \varphi, \text{NP} / \psi \\
\text{PN} / \lambda P[P(\text{john})] \to \text{John} / \lambda P[P(\text{john})] \\
\text{PN} / \lambda Q[Q(\text{mary})] \to \text{Mary} / \lambda Q[Q(\text{mary})] \\
\text{TV} / \lambda x_3 \lambda x_4[x_3(\lambda x_5[\text{love}(x_4, x_5)])] \to \text{loves} / \lambda x_3 \lambda x_4[x_3(\lambda x_5[\text{love}(x_4, x_5)])],
\]

Figure [2.7](#) shows the parse tree from the grammar (31).

### 2.5 Conclusion

Languages in general, whether formal or natural, are governed by rules, our aim of defining a fragment of English language that is, computable requires syntactic rules. In this chapter, we discussed a number of formal grammars highlighting similarities between them and in some cases equivalence. We are therefore left with the choice of which grammar to apply in the definition of our language. Given the four Chomsky grammars (regular grammar, context-free grammar, context-sensitive grammar and unrestricted grammar), categorial grammar,
We discussed three formal representations, namely, propositional logic, first-order logic, and Discourse representation theory. While propositional logic is quite simple to understand, its application to natural language interpretation is impractical as it does not consider the internal structure of the sentences of interest. As a result, we are unable to generate the semantic interpretation of a sentence from the semantic interpretation of its constituents. We see in section 2.4.2 how we can recursively generate interpretation for sentences in our language in first-order logic. Although DRT is an expressive formal language, first-order logic appears to provide an intuitive approach for the representation of temporal information. We, therefore, consider temporal expressions and their representation in the following chapter.
Chapter 3

Temporality in English

3.1 Temporal Ontology

There has been a perpetual discussion on the most appropriate model of time. Given that time and its representation is of immense importance in many fields of study, there has been various choices of the model of time applied. For example, time could be modelled as discrete or dense, instant based or interval based, bounded or unbounded, linear, branching or circular.

In this section we consider some of these choices in order to make decisions on the most appropriate model of time for a controlled natural language with temporal features.

3.1.1 Instant-Based Model of Time

The primitive temporal entity of instant-based model of time is time instants such that the only relation between any two time points is precedence. We can therefore define an instant-based model of time as follows.

\[ \Gamma = \langle T, < \rangle, \]

where \( T \) is a set of time points and \(<\) is the binary relation precedence. There are a number of properties that are naturally imposed on the instant-based model of time, for example, the set \( T \) of time points is a strictly partially ordered set, that it is irreflexive and transitive. We present some properties of instant-based model of time in first-order logic sentences (33)-(38).

(33) Irreflexive: \( \forall i \neg (i < i) \)
(34) Transitive: $\forall i_1 \forall i_2 \forall i_3 ((i_1 < i_2) \land (i_2 < i_3) \rightarrow (i_1 < i_3))$

(35) Anti-symmetry: $\forall i_1 \forall i_2 ((i_1 \leq i_2) \land (i_2 \leq i_1) \rightarrow (i_1 = i_2))$

(36) Connectedness: $\forall i_1 \forall i_2 ((i_1 < i_2) \lor (i_2 < i_1) \land (i_1 = i_2))$

(37) Density: $\forall i_1 \forall i_2 ((i_1 < i_2) \rightarrow \exists i_3 ((i_1 < i_3) \land (i_3 < i_2)))$

(38) No Beginning: $\forall i_1 \exists i_2 (i_2 < i_1)$

No End: $\forall i_1 \exists i_2 (i_1 < i_2)$

### 3.1.2 Interval-Based Model of Time

Instant-based model of time is mostly unsuitable for the reasoning about durative real-life events such as those described in the following sentences.

(39) John lived in Morocco for five years before moving to Canada

(40) Mary slept until Mark arrived.

Sentences (39) and (40) require an interval model of time in order to be sufficiently represented. The interval-based model of time therefore has its primitive temporal entity as intervals. Unlike instants, we are able to define more temporal relations on temporal intervals, these include, precedence $<$, inclusion $\subseteq$ and overlap $O$.

Formally an interval-based model of time is defined as follows.

(41) $\Gamma = \langle T, <, \subseteq, O \rangle$

$T$ is a set of time intervals and the operators $<$, $\subseteq$ and $O$ are binary operators precedence, inclusion and overlap respectively. There are a number of properties that are imposed on these operators.

(42) Reflexivity of $\subseteq$: $\forall i (i \subseteq i)$,

(43) Anti-symmetry of $\subseteq$: $\forall i_1 \forall i_2 ((i_1 \subseteq i_2) \land (i_2 \subseteq i_1) \rightarrow (i_1 = i_2))$

(44) Atomicity of $\subseteq$: $\forall i_1 \exists i_2 ((i_2 \subseteq i_1) \land \forall i_3 ((i_3 \subseteq i_2) \rightarrow (i_3 = i_2)))$

(45) Downward monotonicity of $<$ with respect to $\subseteq$:

$\forall i_1 \forall i_2 \forall i_3 ((i_1 < i_2) \land (i_3 \subseteq i_1) \rightarrow (i_3 < i_2))$
3.2. TENSE LOGIC

(46) Symmetry of $O$: $\forall i_1 \forall i_2 ((i_1 O i_2) \rightarrow (i_2 O i_1))$.

(47) Overlapping intervals intersect in sub-interval:

$$\forall i_1 \forall i_2 (i_1 O i_2 \rightarrow \exists i_3 (i_3 \subseteq i_1 \land z \subseteq i_2 \land \forall i_4 ((i_4 \subseteq i_1 \land i_4 \subseteq i_2) \rightarrow i_4 \rightarrow i_3)))$$

Allen (1983) defines thirteen binary operators on temporal interval, namely, equal, before, after, meets, met-by, overlaps, overlapped by, finishes, finished-by, begins, began-by, contains and during. Table 3.1 describes Allen’s temporal relationships.

<table>
<thead>
<tr>
<th>Relation</th>
<th>Symbol</th>
<th>Inverse</th>
<th>Inverse Symbol</th>
<th>Pictorial Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>X before Y</td>
<td>&lt;</td>
<td>Y after X</td>
<td>&gt;</td>
<td>XXX YYY</td>
</tr>
<tr>
<td>X equal Y</td>
<td>=</td>
<td>Y equal X</td>
<td>=</td>
<td>XXX YYY</td>
</tr>
<tr>
<td>X meets Y</td>
<td>m</td>
<td>Y met-by X</td>
<td>mi</td>
<td>XXXYYYY</td>
</tr>
<tr>
<td>X overlaps Y</td>
<td>o</td>
<td>Y overlapped by X</td>
<td>oi</td>
<td>XXX YYY</td>
</tr>
<tr>
<td>X during Y</td>
<td>d</td>
<td>Y contains X</td>
<td>di</td>
<td>XXX YYYYYYYY</td>
</tr>
<tr>
<td>X starts Y</td>
<td>s</td>
<td>Y started by X</td>
<td>si</td>
<td>XXX YYYYYYYY</td>
</tr>
<tr>
<td>X finishes Y</td>
<td>f</td>
<td>Y finished by X</td>
<td>fi</td>
<td>XXX YYYYYYYY</td>
</tr>
</tbody>
</table>

Table 3.1: Allen’s Thirteen Temporal Interval Relationships

3.2 Tense Logic

Tense Logic can be considered an extension of classical propositional logic. Formulas of classical propositional logic are interpreted as either true or false. Suppose a valuation is a function mapping propositional logic formulas to truth values, the assigned truth value of a classical propositional formula is fixed, while the valuation of formulas in tense logic is time-dependent. Given therefore a proposition $q$, $\delta$ is a valuation that assigns a truth value to the proposition at a given time instant $t$, this is written as follows.

(48) $\delta(t)(q)$

Prior (1957) defined four temporal modal operators that assign truth values to propositions dependent on time thus:
• $P$: “It has at some time been the case that...”

• $F$: “It will at some time be the case that...”

• $H$: “It has always been the case that...”

• $G$: “It will always be the case that...”

These operators can be combined to define more complex temporal functions. For example, $GP(John\ arrives)$ means “it will always be the case that it has at some time been the case John arrives.”

Formally, the set of Prior tense logic is defined as the smallest set containing the propositional variables that is, closed under constructing new formulas using the Boolean connectives $\neg$ and $\land$ and the temporal operators $G$ and $H$. We can therefore define the notion of truth of a propositional formula $q$ at a time point $t$ in a model $M = (T, <, \delta)$:

- $M, t \vDash q$ if $\delta(t)(q) = 1$
- $M, t \vDash \neg q$ if not $M, t \vDash q$
- $M, t \vDash q \land p$ if $M, t \vDash q$ and $M, t \vDash p$
- $M, t \vDash Gq$ if $M, s \vDash q$ for all $s$ given that $t < s$
- $M, t \vDash Hq$ if $M, s \vDash q$ for all $s$ given that $s < t$

Prior tense logic is however considered low in expressiveness and there exist various extensions. The most popular amongst these extensions is Kamp’s tense logic \cite{Kamp1968}. This extension includes two new temporal operations – $S$ and $U$ which represent \textit{since} and \textit{until} respectively. These operators are formally defined thus:

- $M, t \vDash Uqp$ if $M, s \vDash q$ for some $s$ such that $t < s$ and $M, u \vDash p$ for all $u$ such that $t < u < s$,
- $M, t \vDash Sqp$ if $M, s \vDash q$ for some $s$ such that $s < t$ and $M, u \vDash p$ for all $u$ where $s < u < t$. 
3.2. TENSE LOGIC

3.2.1 Branching Time Logic

In this section we consider a model of time where time is linear in the past but branched in the future (Burgess (1979)). We therefore require a special treatment for $Fq$. One way to interpret the branched time is to assume flows of time are trees. That is, they are connected strictly partial orders that is linear in the past. Each branch of the tree is defined as $\Gamma = \langle T, < \rangle$ with a common time point $t$. If each branch represents a possible course of events, $b$, we say $b$ passes through $t$ or $t$ lies on $b$. We therefore consider possible future of $t$ as a set of time points on a fixed branch $b$ which passed through $t$. The semantics of a proposition within a given branch are as follows.

- $M, t, b \models q$ if $\delta(t)(q)$
- $M, t, b \models \neg q$ if not $M, t, b \models q$
- $M, t, b \models q \land p$ if $M, t, b \models q$ and $M, t, b \models p$
- $M, t, b \models Gq$ if $M, t, b \models q$ for all $s$ on $b$ such that $t < s$
- $M, t, b \models Hq$ if $M, t, b \models q$ for all $s$ on $b$ such that $s < t$
- $M, t, b \models \Box q$ if $M, t, c \models q$ for all branches $c$ through $t$

Just as with the linear tense logic, branching time systems can be extended by including the $\text{Since}$ and $\text{Until}$ operator. This extension creates the language computation tree logic (CTL).

3.2.2 Interval Based Temporal Logic

The basic tense logic and branching time logic assume time instants are the primitive entity of time. An alternative way of representing time as discussed in section 3.1.2 is having time intervals as the primitive entity of time.

Halpern, Shoham (1991) defined a set of equivalent modal operators of the form $\langle X \rangle$. Halpern and Shoham’s temporal logic (HS logic) consists of three basic operators – $\langle A \rangle$, $\langle B \rangle$ and $\langle E \rangle$, these operators are equivalent to Allen’s $\text{meets}$, $\text{begins}$ and $\text{ends}$ temporal interval relations respectively. The inverse of these operators denoted $\langle \bar{A} \rangle$, $\langle \bar{B} \rangle$, $\langle \bar{E} \rangle$ are equivalent to Allen’s temporal relations $\text{met-by}$, $\text{began-by}$, and $\text{ended-by}$ respectively. In (49)-54 we show how the
equivlent operators of Allen’s relations before, after, during, contains, overlaps and overlapped-by are expressed in HS logic respectively.

\[(49) \langle L \rangle \phi \equiv \langle A \rangle \langle A \rangle \phi\]

\[(50) \langle L \rangle \phi \equiv \langle A \rangle \langle A \rangle \phi\]

\[(51) \langle D \rangle \phi \equiv \langle B \rangle \langle E \rangle \phi\]

\[(52) \langle D \rangle \phi \equiv \langle B \rangle \langle E \rangle \phi\]

\[(53) \langle O \rangle \phi \equiv \langle E \rangle \langle B \rangle \phi\]

\[(54) \langle O \rangle \phi \equiv \langle B \rangle \langle E \rangle \phi\]

Although it is a matter of debate as to which is the more appropriate model of temporal representation between instants and intervals. Both ontologies are closely related and reducible to each other. Temporal intervals can be bounded by pairs of instants – beginning and end. Time instants can be viewed as intervals whose endpoints coincide or have no duration.

### 3.3 TimeML

Setzer (2001) developed an annotation scheme for the identification of features in texts that enable a reader to determine the temporal order and time of events reported. TimeML (Pustejovsky et al. (2005)) represents temporal expressions contained in natural language text, this is done by annotating a given natural language text with XML tags. It is the ISO standard for time and event markup and annotation (ISO 24617-1:2012). TimeML (Markup Language for Temporal and Event Expressions) was designed to address specifically, four basic problems in event-temporal identification:

1. Time stamping of events (identifying an event and anchoring it in time);

2. Ordering events with respect to one another (lexical versus discourse properties of ordering);

3. Reasoning with contextually underspecified temporal expressions (temporal expressions such as last week and two weeks before);
4. Reasoning about the persistence of events (how long does an event or the outcome of an event last).

TimeML performs information extraction using XML-based tags these include – EVENT, TIMEX3, SIGNAL, LINK

The Event tag \langle EVENT \rangle is used for the extraction and annotation of situations; these situations could either be instantaneous or within intervals (punctual or occur over a period of time). Events are also used to describe states or circumstances in which something holds. Events are expressed by tensed and untensed verbs, adjectives, predicate clauses or prepositional phrases.

The TIMEX3 is used to mark up explicit temporal expressions like times of the day, dates, duration, etc. There are three major types of TIMEX3 expressions

- Fully specified temporal expressions June 11, 1989, Summer 2002;
- Unspecified temporal expressions, Monday, Next month, Last year, Two day ago;
- Durations, Three months, Two years.

The SIGNAL tag is used to annotate sections of the text, typically function words that indicate how temporal objects are to be related to each other. SIGNAL contains several types of linguistic elements which serve as indicators of temporal relations, for example, temporal propositions, temporal connectives, and subordinators. The SIGNAL tag also annotates polarity (not, no, none, etc.) as well as indicators of temporal quantifications such as twice, three times, etc.

TimeML introduces three types of LINKs namely – TLINK, SLINK, ALINK. The TLINK represents the temporal relationship between events or between events and times. It establishes a link between the involved entities making explicit if they are simultaneous, one before the other, one after the other, one immediately after the other one including the other, one being included in the other, one immediately before the other, one being the beginning of the other, one being begun by the other, one being the ending of the other, one being ended by the other. The SLINK is used for contexts that involve modality and factives. The SLINK is applied when an event instance subordinates another. The SLINK or the Subordination Link is used for context introduction relations between two events or an event and a signal.
3.4 Tense

Speakers of the English language and indeed any other natural language describe situations holding within a temporal context which can either be in the past, present or future. Consider the following simple English sentences.

(55) John played football.
(56) I see Mary.
(57) John will play football.

Sentences (55)-(57) relate the time of event with times before, simultaneous to and after the time of speech respectively. Relationships between the event time and speech time, however, do not quite account for all the possible tenses constructions in English. We, therefore, discuss the existing tense theories as well as other related temporal phenomena required for appropriate semantic representation of natural language in this section.

3.4.1 Reichenbach’s Theory of Tense

We have established that tenses relate the time of speech to the time of event. Interestingly, tenses have been observed to involve quite a bit more complex temporal relationships. Along with the obvious relationship between the event time and speech time, Reichenbach (1947) observed a third coordinate required for the representation of tenses in sentences called the reference time. Reichenbach’s temporal coordinates largely serve as a basis for most theories of the semantic representation of temporal expressions in natural language.

Reichenbach, therefore, provides for each tense the order of the temporal coordinates on a timeline relative to each other. Therefore sentences (55)-(57) given above show different order between the event time and speech time. However, there seems to be no explicit indication that there is a third coordinate. We can, however, observe the third coordinate in tense constructions such as those in the following sentences.
(58) John had gone.

(59) John had gone before Mary arrived.

Sentences such as (58) according to Reichenbach refer to two events rather than one. First, the moment of *John leaving* which is the event time and second, a time between the event time and speech time – *reference time*. It is difficult to notice the existence of the reference point from sentence (58), but when modified by a temporal prepositional phrase, it becomes a lot more obvious as we can see in sentence (59). We observe there exists between the time of event, i.e., *John leaving* and the speech time – a time when *Mary arrived*. This time is what Reichenbach refers to as the *reference time*.

Having established the existence of the three time coordinates as defined by Reichenbach, we can now show how each tense in English is represented on a timeline based on the relative positions of the three temporal coordinates. The simple past tense is interpreted as the event time (denoted as $E$) and the reference time (denoted as $R$) coinciding before the speech time (denoted as $S$) on the timeline. The present tense is interpreted as the event, speech and reference time coinciding; the future tense is interpreted as the event time and the reference time occurring simultaneously after the speech time.

Other than the three simple tenses in English – past, present and future, there are more complex tense constructions as in sentence (58) and (59). The progressive is viewed as an extended event. We, therefore, represent the event time of extended events as $E^-$. Table 3.2 provides the positions of the three coordinates on a timeline as specified by the twelve tense and aspect constructions we have in English.

It might be worth mentioning that other theories of tense such as Comrie (1976) and Binnick, Stowell (2012) consider the inclusion of reference time as a simple tense coordinate as redundant given that they always coincide with the event time.

3.5 Grammatical Aspect

It is rather difficult to separate the notion of tense from that of aspect. Comrie (1985) defines tense as the grammaticalization of location of time, Binnick and deSwart (2012) considered tense as deictic because it only requires reference to
CHAPTER 3. TEMPORALITY IN ENGLISH

<table>
<thead>
<tr>
<th>Tense</th>
<th>Timeline Coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Past tense</td>
<td>E,R,S</td>
</tr>
<tr>
<td>Present Tense</td>
<td>E,R,S</td>
</tr>
<tr>
<td>Future Tense</td>
<td>S,E,R</td>
</tr>
<tr>
<td>Past Perfect</td>
<td>E,R,S</td>
</tr>
<tr>
<td>Present Perfect</td>
<td>E,S,R</td>
</tr>
<tr>
<td>Future Perfect</td>
<td>S,E,R</td>
</tr>
<tr>
<td>Past progressive</td>
<td>E→,R,S</td>
</tr>
<tr>
<td>Present Progressive</td>
<td>E→,S,R</td>
</tr>
<tr>
<td>Future Progressive</td>
<td>S,R,E→</td>
</tr>
<tr>
<td>Past Perfect Progressive</td>
<td>E→,R,S</td>
</tr>
<tr>
<td>Present Perfect Progressive</td>
<td>E→,R,S</td>
</tr>
<tr>
<td>Future Perfect Progressive</td>
<td>S,E→,R</td>
</tr>
</tbody>
</table>

Table 3.2: Time coordinates of Reichenbach’s tense

the speech time. Aspects on the other hand according to Comrie (1976) are the different ways of viewing the internal temporal structure of a situation. We have two different grammatical aspects namely the perfective aspect and the imperfective aspect. As proposed by Binnick (1991), there are three forms of the imperfective aspect – concrete-processural, indefinite-iterative and general-factual. Concrete-processural aspect refers to the durative continuation of a particular episode as in sentence (60).

(60) I am walking to the mall.

indefinite-iterative aspects describe habitual, repetitive episodes distantly spaced. Consider the following sentence.

(61) Women are wearing fur coats this season.

Similarly, we have three forms of the perfective aspect – concrete-factual, aggregate meaning and graphic exemplary. The concrete-factuals are definite, specific episodes viewed strictly as an occurrence as seen in the following sentence.
3.5. GRAMMATICAL ASPECT

(62) John had walked the dog this morning.

Aggregate meaning perfective aspect describes iterative, repetitive episodes closely spaced in time and viewed as a unit, for example,

(63) He repeated the question to me several times.

Referring to Reichenbach’s theory of tense, we observed that the idea of aspects seems to be part of tense and it is quite difficult to consider the two as separate concepts. However, there is much more to aspect than just the position of time coordinates on a timeline or in a predicate.

Syntactically when a verb is in the perfective, it is required to be preceded by the auxiliary verb *have*, for example, sentence (64). When in the perfective, the main verb takes the participle inflection as in sentence (64). Note however that there are verbs that do not have participle inflection and just retain the past inflection for its perfective. The information of the tense of a perfective verb phrase is therefore provided by the tense of the auxiliary verb *have*. Sentence (64) is therefore a present perfective sentence. As opposed to (65) which is a past perfective sentence since it has the past-tensed auxiliary verb *had*.

(64) I have eaten.

(65) I had eaten

Semantically, the perfect in the present tense always locates the event time in the past, such that there is a result state of the said event that is still of relevance until the time of speech. For instance, sentence (64) can be interpreted as the event *eating*, occurring at some past time. As a result, there has been a state of completion of the event which is true until the time of speech, in this case, is also the time of reference. Prior’s tense logic defines the present perfect as being similar to the past tense. In the past and future perfect, the time of reference is not simultaneous with the speech time but rather coincides with some temporal modifier as in sentence (66).

(66) I had eaten before Mary arrived

(67) I will have eaten before Mary arrives.
Due to the deictic nature of the present tense, we often do not have the present perfect with a temporal modifier providing information of the time of reference, as it just coincides with the time of speech. The past and future perfect, however, require a temporal modifier.

The progressive aspect syntactically requires the main verbs to have the suffix -ing. The main verb is also required to be preceded by the auxiliary verb be. Just like the perfect, given that the main verb does not carry information of the tense of the sentence, the be auxiliary verb provides information on the tense of the sentence.

The semantics of progressive is one that had been extensively studied. While we intuitively use the progressive to express continuous events, we observe that its relationship with tense is an interesting one. An event in the progressive is considered to be continuous from a given time of reference. Consider the following sentences.

(68) Jane was running.

(69) Jane was drawing a circle.

We can interpret sentence (68) as there exists a time in the past such that the event running began at a time prior to a time of reference and continued afterward (Heny (1982)). Suppose we attempt to interpret sentence (69) similarly, we would say there exists a time in the past such the event drawing a circle began at a prior time and was completed at a time afterward. Making such a claim requires us to assume that the event of drawing a circle was complete. Suppose Jane was unfortunately struck by lightning before the completion of the event, sentence (69) will still be an appropriate statement; our interpretation will however not be, this sort of situation as observed by Dowty (1979) is called the imperfective paradox. To understand why our interpretation for sentence (68) is correct and (69) is not, we require an information of aspectual classification of verbs as given in section 3.6.

3.5.1 Syntax of English Tense

Having discussed the tense-aspect constructions in English, we provide in this section, their syntactic structure following the framework of generative grammar as proposed by Chomsky (1957), Chomsky (1969). Chomsky’s syntactic theory of
English tense and auxiliary verbs was designed to generate the set of grammatical English sentences and their respective phrase structure representations. Syntactic structure of tense according to Chomsky is based on certain production rules consisting of categories– $S$ Sentence, $Aux$ Auxiliary phrase, $VP$ Verb Phrase etc. An example of Chomsky’s grammar for English tense is given below

$S \rightarrow NP, Aux, VP$
$Aux \rightarrow Tense, (Modal), (have+en), (be+ing)$
$Tense \rightarrow past$
$Tense \rightarrow present$
$VP \rightarrow V, NP$

The first production rule in (70) defines a three-branch structure for sentences. The second rule tells us that depending on what the tense is, we may choose zero or more of the parenthesized elements in the given order. The first symbol Tense is the mandatory morphosyntactic category with two possible values past or present defined by the third and fourth rules. When we have multiple auxiliary verbs in a sentence, their linear order relative to each other is determined by the second rule as well.

(71) John could have been reading a book.

From the production rules in (70) we have the parse tree in Figure 3.1 representing the deep structure of tensed sentence (71).

3.5.2 Representation of Tenses with Temporal Intervals

We consider in this section the interpretation of the temporal information provided by tenses with temporal intervals. To do this, we need to identify the temporal coordinates in Reichenbach’s theory and represent them as temporal intervals. The ordering placed on these intervals will be dependent on the tense of the sentence of interest.

(72) Mary kissed John.

(73) $\lambda I[\exists i_0 (\text{kiss(mary, john)}(i_0) \land (i_0 < \text{now}) \land (i_0 \subseteq I))]$
Given a past-tensed sentence such (72), we interpret it as (73) where the interval $i_0$ represents the temporal interval within which the event occurred, and the constant now represents the speech time. From Reichenbach’s theory, the past tense places the event time prior to the time of speech. We, therefore, represent temporal ordering with the precedence operator $<$. Hence the sentence above is interpreted as there exists an interval where the event of Mary kissing John occurred, the said interval is before the time of speech, and it is within a yet to be defined temporal context. The future tense works similarly to the past except for the temporal ordering of the event and speech time is reversed.

The interval $I$ can be considered as a temporal context within which the event or state described by the sentence occurred. We can, therefore, provide an interpretation for sentences such as (74).

(74) Mary kissed John yesterday.

### 3.6 Aspectual Classification

Aspectual class is the classification of verbs according to the situation that the verb phrase describes. The aspectual class of a given verb phrase is determined by its internal and external structure. Vendler (1967) defined these classes first,
after which more elaborate descriptions were provided by Dowty (1979), Smith (1983). In this section, we will discuss these aspectual classes and enumerate tests for distinguishing them.

### 3.6.1 States and Events

The most basic aspectual classes are the *states* and *events*. States describe static and homogeneous situations with no internal structure. They hold over a period of time (can be judged at a moment in time), and they have no inherent endpoint or culmination (i.e., they can continue indefinitely except otherwise specified mostly by a temporal modifier), for example,

\[(75)\text{ Steve loves Kate.}\]

*Events* on the other hand are heterogeneous and dynamic, meaning they contain phases. They can occur over a period of time (interval) or a particular point in time (moment), they may have inherent endpoint (not mandatory), for example,

\[(76)\text{ Ben wrote the poem.}\]

There are a few factors that distinguish states from events. For example, states are not compatible with the progressive aspect – *be ... – ing*

\[(77)\text{ *John is owning the house.}\]

\[(78)\text{ ?Kate is loving the summer.}\]

\[(79)\text{ John is walking to the store.}\]

Sentence \[(77)\] is ungrammatical but there are certain readings where \[(78)\] is permissible where the verb *loving* is treated as an event. Sentence \[(79)\] illustrates the compatibility of the progressive with events.

States cannot be used in imperatives. Consider the sentences below.

\[(80)\text{ Hey you! come here.}\]

\[(81)\text{ *Hey you! know English.}\]
Sentence (81) is ungrammatical as opposed to sentence (80) which passes a command with an event verb.

Event verb phrases in simple present tense have a habitual/repetitive interpretation. States in simple present tense are interpreted as holding at the present moment simultaneous with the time of speech.

(82) John washes the dishes.

(83) John loves Mary.

**Bounded and Unbounded Events**

Events are further subdivided into *bounded* and *unbounded events*. Bounded events have an inherent endpoint after which the event of interest ceases to occur (i.e., the event culminates), this inherent termination or culmination is usually signified by a change in state of affairs, for example,

(84) The dog died last night.

We know the sentence verb *died* is bounded because its completion causes a change in the state of affairs from the dog being alive to being dead. Bounded events are also known as telic.

*Unbounded event (Activities)* have no inherent endpoint or culmination. The same event can continue over an indefinite period of time. They are also referred to as the atelic.
As proposed by Dowty, one way of distinguishing between these events is through their interaction with the temporal preposition *in*. Activities (unbounded events) cannot occur with *in*-phrase time adverbials, while bounded events can. Consider the sentences below.

(85) Walden finished the paper in 2 weeks.

(86) *Walden played the cards in 2 minutes.

**Achievements and Accomplishments**

The bounded event is subdivided to *accomplishments* and *achievements*. Accomplishments have two structures - a process which happens over a period of time leading to an *endpoint* or *culmination*, for example,

(87) Jane drew a circle.

We can observe from the above sentence that the eventual state of existence of a circle is preceded by an activity of *Jane drawing*. When in the progressive, the culmination of the accomplishments is stripped. Consider the following sentence.

(88) Jane was drawing a circle.

Sentences such as (88) describe the event but omit the culmination. Dowty (1979) refers sentences (i.e., accomplishments in the progressive) as the imperfective paradox.

Achievements have no process preceding the endpoint. Instead, they correspond to the transition point between states or can be said to introduce a new state of affairs. They are expected to occur at a moment in time, for example,

(89) Linda’s grandma passed away.

A detailed aspectual class test as proposed by Dowty is given in table 3.3.

### 3.7 Temporal Modifiers

There is one more temporal expression left for us to discuss – temporal modifiers. These modifiers can also be considered as adverbials because of their semantic roles on the state or event described in the main sentence. Consider the following sentences.
CHAPTER 3. TEMPORALITY IN ENGLISH

<table>
<thead>
<tr>
<th>Criterion</th>
<th>States</th>
<th>Activities</th>
<th>Accomplishments</th>
<th>Achievements</th>
</tr>
</thead>
<tbody>
<tr>
<td>meets nonstative tests</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>?</td>
</tr>
<tr>
<td>habitual interpretation in simple present tense</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>φ for an hour</td>
<td>permitted</td>
<td>permitted</td>
<td>permitted</td>
<td>not permitted</td>
</tr>
<tr>
<td>spend an hour φing:</td>
<td>permitted</td>
<td>permitted</td>
<td>permitted</td>
<td>not permitted</td>
</tr>
<tr>
<td>φ in an hour, take an hour to φ:</td>
<td>nor permitted</td>
<td>not permitted</td>
<td>permitted</td>
<td>permitted</td>
</tr>
<tr>
<td>φ for an hour entails φ at all times in the hour</td>
<td>yes</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>x is φing entails x has φed:</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>complement of stop</td>
<td>permitted</td>
<td>permitted</td>
<td>permitted</td>
<td>not permitted</td>
</tr>
<tr>
<td>complement of finish</td>
<td>not permitted</td>
<td>not permitted</td>
<td>permitted</td>
<td>not permitted</td>
</tr>
<tr>
<td>ambiguity with almost</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>x φed in an hour entails x was φing during that hour:</td>
<td>no</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>occurs with studiously, attentively, carefully, etc.</td>
<td>not permitted</td>
<td>permitted</td>
<td>permitted</td>
<td>not permitted</td>
</tr>
</tbody>
</table>

Table 3.3: Dowty’s aspectual class test

(90) John kissed Mary

(91) John kissed Mary yesterday.

(92) John Kissed Mary before the meeting yesterday.

The above sentences describe the same event of *John kissing Mary*. There is, however, a difference due to the type of temporal context provided by the complements in sentences (91) and (92).

Some of the examples of temporal prepositions in English include *in, on, at, for, by, since, until, before, after, during, from, to, between, when, while etc.* One might have noticed that some of these prepositions have other linguistic uses other than temporal, for example, the locative prepositions like *in, on and at* can either be temporal or spatial. Consider the sentences below.

(93) Mark married Jane in Las Vegas.

(94) Mark married Jane on the rooftop.

(95) Mark married Jane at the galleria

(96) John arrived in January.

(97) John arrived on Monday.

(98) John arrived at noon.
Sentences (93)-(95) from our intuitive reading as English speakers illustrate the spatial use of the prepositions *in*, *on* and *at* as opposed to sentences (96)-(98) which illustrate the temporal use. Amongst the topics of interest in the interpretation of prepositions include syntactic markers for distinguishing between temporal and non-temporal uses of the prepositions of interest. Brée (1985), Mittwoch (1988) amongst others provided syntactic markers for various prepositions. Brée in particular employed a very systematic procedure by analysing random sentences from an English corpus from which he was able to propose syntactic and semantic rules.

Temporal prepositions can be said to provide a temporal context for the main event or state of a given sentence. We notice from sentences (96)-(98) that there are restrictions to the type of complements individual prepositions permit. The preposition *in* for example will most likely permit years and months, while *on* is most compatible with days and dates while *at* permits clock times. The syntactic structure of the main clause is also of huge importance in the correct use of temporal prepositions. Consider the following sentences.

(99) John did not return until Mary called.

(100) *John did not return since Mary called.

(101) John has not returned since Mary called.

From the above sentences we observe that sentence (100) is unacceptable, whereas, sentence (101) in the perfect is. Other than the tense/aspect construction of the main clause, an important factor in determining the grammatical use of temporal preposition is the aspectual class of the main verb. From the Dowty’s study of the relation of aspectual classes and temporal prepositions we observe certain restrictions. For example, the preposition *for* only permits states or activities as main the verb.

Restrictions are also placed on the subordinate clause as we have already observed above. Other than that, temporal prepositions can either have temporal nouns, events nouns or sentences as complements. The syntactic category selected by a given prepositions is often peculiar. Consider the following sentences.

(102) Marcos drank a cup of tea during the lecture.

(103) *Marcos drank a cup of tea during noon.
It appears from sentences (102)-(104) that the preposition *during* will mostly permit event nouns and reject temporal proper nouns and sentential complements. Other temporal prepositions have their peculiar restrictions as well.

The semantic purpose of temporal prepositions is primarily that of context provision. In our discussion of tense we observed tense provides a temporal ordering between the speech time and the event time. While aspect provides a location for the time of reference on the timeline, temporal prepositions provide a real context for the time of reference, and in many cases, grammatical aspects require temporal prepositions for us to have complete grammatical sentences. As a result, there is a requirement of agreement between the tense aspects and temporal prepositions as observed by Partee (1973) and Hornstein (1977).

Syntactically, temporal adverbials can occur in two different positions: first after the main sentences, as in sentences like (102), or they can precede the main sentences, as in (105).

(105) During the lecture, Marcos drank a cup of tea.

Most of the syntactic rules that govern the grammatical use of temporal prepositions, in general, are mostly governed by their semantic interpretation. Pratt-Hartmann, Francez (2001), Von Stechow (2002), Pratt-Hartmann (2005) provide semantic interpretations for the commonly used temporal prepositions in first-order logic or some of its variants like Interval Temporal Logic (ITL).

3.8 Discussion

Expressing temporality in natural language requires the involvement of several linguistic and semantic phenomena, some of which we introduce in this chapter. This includes whether time should be represented as time instants or intervals. We observe that instant-based models are not expressive enough for providing semantic interpretation for durative events. Allen’s temporal relations, however, appear to provide an extensive set of operators that can express relationships between temporal intervals. We, therefore, adopt first-order interval logic (a variety of interval temporal logic) for the representation of temporal expressions in our language.
3.8. DISCUSSION

From a linguistic perspective, temporal expressions do not function in isolation, there are therefore several rules that govern the mutual compatibility of various temporal expressions within a sentence. For instance, we observed that stative verbs are not permitted to be in the progressive; moreover, temporal prepositions are particular about the syntactic structure of their complements as well as their main clauses. We are also required to enforce tense agreement between main and subordinate clauses as observed by (Partee, 1973) and Hornstein (1977).

There are also semantic consequences as a result of the interaction of the various temporal expressions considered in this chapter. For instance, Dowty (1979) observed that the semantic interpretation of accomplishments in the progressive creates the imperfective paradox. We also observe that certain temporal modifiers are selective about the aspectual classes of their main and subordinate clauses. For example, the temporal conjunction while requires an activity or stative verb in its subordinate clause as opposed to until which is complemented by punctual events – achievements.

Defining a controlled natural language with temporal features, therefore, requires us to consider how we can provide the syntactic representation for sentences such that the required restrictions on temporal expressions are observed. Given that the interaction of these temporal expressions also have semantic implications, we require semantic rules that generate the appropriate semantic interpretation for sentences with regard to the effects of the interactions of tense, aspects, aspectual class, and temporal modifiers. We discuss in more detail the syntax and semantic interpretation of the temporal modifiers adopted in our proposed language, comparing it to existing theories in chapter 5.
Chapter 4

Controlled Natural Language

Natural Language is the primary mode of communication amongst humans. It requires every party in the communication channel to have an intuitive understanding of the language grammar and meaning. Despite this inherent property of natural languages, they appear to be difficult to parse by non-native speakers and machines. As a result, there have been several attempts to develop languages that are just as intuitive as regular natural languages but also easy to parse by non-native speakers and more importantly, machines; this has led to the definition of a class of languages called Controlled Natural Languages. In this chapter, we define controlled natural languages and provide a brief survey of existing ones. Finally, we provide a summary of what the definition of a controlled natural language with temporal features entails.

4.1 What is a Controlled Natural Language?

Defining CNLs is not very straightforward given that existing ones have been developed in response to varied problems such as linguistic, interfaces for formal languages, etc. Existing controlled languages, however, share the common property of having one base language, where a CNL base language is the natural language from which the said CNL is derived. We, therefore, do not have a CNL that is a composition of more than one base language. The level of restriction however placed on a given base natural language is dependent on various factors such as its users and application.

Perhaps the earliest known form of restriction placed on natural language syntax is that provided by Aristotle’s Syllogisms. Although the syllogisms were based
4.1. WHAT IS A CONTROLLED NATURAL LANGUAGE?

on Greek, we can define a similar language in English with only four sentence
structures namely,

(106) Every $X$ is a $Y$

(107) Some $X$ is a $Y$

(108) No $X$ is a $Y$

(109) Some $X$ is not a $Y$.

These syllogisms can easily be translated to the following first order logic formu-
las;

(110) $\forall \alpha (X(\alpha) \rightarrow Y(\alpha))$

(111) $\exists \alpha (X(\alpha) \land Y(\alpha))$

(112) $\neg \exists \alpha (X(\alpha) \land Y(\alpha))$

(113) $\neg \forall \alpha (X(\alpha) \rightarrow Y(\alpha))$

The interpretation of controlled natural language sentences into some formal lan-
guage has recently led controlled natural languages to be defined as the fragment
of a given natural language that is computer processable (Schwitter (2005), Cur-
ran et al. (2007)), thus enabling application to software and hardware specifica-
tion, ontology authoring, query systems etc.

Kuhn (2014) in his detailed survey of CNLs observed that every controlled
natural language must indeed possess four properties; we, therefore, take these
properties to be definitive.

DEFINITION 4.1.1 A language is called a controlled natural language if and
only if it has all of the following four properties:

1. It is based on exactly one natural language (its “base language”).

2. The most important difference between it and its base language (but not
necessarily the only one) is that it is more restrictive concerning lexicon,
syntax, and/or semantics.
3. It preserves most of the natural properties of its base language, so that
speakers of the base language can intuitively and correctly understand texts
in the controlled natural language, at least to a substantial degree.

4. It is a constructed language, which means that it is explicitly and consciously
defined, and is not the product of an implicit and natural process (even
though it is based on a natural language that is the product of an implicit
and natural process).

Recognizing the huge disparity amongst the existing CNLs, Kuhn defined
four criteria for classifying CNLs, namely – precision, expressiveness, naturalness,
and simplicity. Precision measures a language’s ambiguity, predictability, and
formality of definition. Expressiveness is dependent on the level of restriction
on the base language, where languages with more restrictions (e.g., Aristotle’s
syllogisms) are less expressive, while less restriction produces more expressive
CNLs. Naturalness is based on the understandability and the natural look and
feel of the language. Simplicity measures how easy it is to learn, use and apply as
required. These properties make up Kuhn’s classification scheme – PENS. We do
not explain here how Kuhn assigns scores to CNLs based on these four criteria;
the user can refer to Kuhn (2014) for a detailed report on assigning scores to
CNLs.

4.2 Applications of CNLs

Although the regimentation of a natural language’s grammar in order to permit
machine processing appears to be an interesting endeavour, a noteworthy question
is why anyone would care about the development of such a language. As we have
previously stated, in the early days of controlled natural languages, they were
mostly developed to ease the learning of a given language by non-speakers. That
is, they were made to aid the familiarisation of non-native speakers with a lan-
guage’s grammar and vocabulary. This use of controlled natural languages might
be considered less significant given the level of technological progress, particu-
larly in software and the internet. We now tend to consider a controlled natural
language as a subset of a given natural language that is computer processable.

The more popular controlled natural languages have English as their base
language; however, there are controlled natural languages with other base languages. Examples include Mandarin (Zhang, Peng (2012)), Greek (Vassiliou et al. (2003)), Portuguese (Marrafa et al. (2012)), isiZulu (Keet, Khumalo (2014)) and Runyankore (Byamugisha et al. (2016)). We, however, consider examples of CNLs with English as base language in the rest of this chapter given that our aim is the development of a controlled English with temporal features.

Many of the currently existing controlled natural languages are designed to serve as interfaces for formal languages. Attempto Controlled English (Fuchs et al. (1999)), for example, provides DRS interpretations for its member sentences. Representing sentential semantics in some formal language has found some real-world applications. A popular domain of application is the writing of software/hardware specification (Fuchs et al. (1990)). Another popular area of application as shown by Smart et al. (2010), Denaux et al. (2010), Ferré (2012), Power (2012) is ontology authoring, querying and editing.

Other than these areas of application, controlled natural languages also have applications in business contract development (Pace, Rosner (2010)), tax fraud detection (Calafato et al. (2016)), proof checking mathematical text (Cramer et al. (2010)) and speech recognition interfaces (Kaljurand, Alumäe (2012)). Controlled natural languages over the last decade have been an interesting area of study primarily because of their ability to enhance access to computer software and formal languages.

4.3 Survey of Controlled Natural Languages

Most controlled natural languages vastly exceed Aristotle’s syllogisms in their syntactic ambition. In this section, we consider in greater detail some of these control natural languages, their syntax, semantic interpretation, and applications.

4.3.1 Attempto Controlled English (ACE)

Attempto Controlled English (Fuchs et al. (1990)) is a controlled, precisely defined subset of English that can automatically and unambiguously be translated into DRS or, optionally, Prolog source code.

Initially, ACE was used for the disambiguation of software requirement specification documents. The system is now used in many other domains. ACE
converts text inputs into DRS, a variant of first-order logic which is a knowledge representation of the inputted text.

Attempto’s lexicon has the following categories – nouns, verbs, adjectives, numbers and technical symbols. Nouns in Attempto are either common nouns, proper nouns or personal pronouns. Common nouns can either be countable or mass, where countable nouns are preceded by determiners or numbers. Mass nouns, in contrast, are not preceded by determiners.

Verbs in Attempto denote states and events; they can only be used in the active voice third person singular or plural present tense. Phrasal and modal verbs are not permitted. Attempto distinguishes two functions for adjectives – sentential complements and nominal modifiers. Modal adjectives such as possible, probable, etc. are not accepted.

Attempto has two sentential forms – declarative and interrogative sentences. Declarative sentences are used to describe the world, while interrogative sentences are used for verification of the specification. Simple declarative sentences combine to form composite sentences with the aid of coordinates.

ACE uses the Attempto Parsing Engine (APE) in generating DRS interpretations, syntax tree as well as paraphrases for the input text. A paraphrase represents the understanding of the ACE text by the parser and can hence be rephrased by the user in case of wrong interpretations. DRSs are used in reasoning and can optionally be represented in OWL/SWRL and Prolog.

Reasoner for Attempto Controlled English (RACE) is a first-order reasoner that can show the consistency or inconsistency of a set of ACE sentences, prove theorems and answer queries from translated ACE axioms.

4.3.2 PENG

PENG (Processable ENGlisht) [Schwitter (2002)] is a controlled natural language with a well-defined grammar. It provides interpretations for its sentences in first-order logic via discourse representation structures (DRS). PENG shares a lot of similarities with ACE. PENG, however, has a look-ahead text editor which restricts the possible sentence constructions inputted by a user.

Just like ACE, PENG was developed to aid authors in writing unambiguous, complete, consistent specifications and facilitating the acquisition of contained knowledge by machines in order to prove theorems and build models. PENG’s lexicon contains predefined function words, these include words in the category
determiners, conjunctions and prepositions. Content words are updated based on the domain of application by the user, these include words of the category nouns, verbs, adjectives and adverbs.

Simple sentences in PENG are of the following structure
\[
S \rightarrow \text{Subject} + \text{Predicate} \\
\text{Subject} \rightarrow \text{Determiner} + \{\text{Pre-nominal modifier}\} + \text{Nominal Head} \{+\text{Post-nominal modifier}\} \\
\text{Subject} \rightarrow \text{Nominal Head} \\
\text{Predicate} \rightarrow \{\text{Negation}\} + \text{Verbal Head} + \text{Complement}
\]

Schwitter (2002) reproduced Schubert’s Steamroller specification problem Pelletier (1986) in PENG syntax in order to show that reasoning on generated semantic interpretations will produce the same inference.

4.3.3 Rabbit

Rabbit (Hart et al. (2008)) is a controlled natural language designed primarily for writing ontologies. It serves as a natural language interface for OWL. Rabbit was designed to allow domain experts to express their knowledge easily with as much necessary detail as possible.

Rabbit has three language elements – those for expressing axioms, those for expressing declarations and those used to refer to other ontologies. Declarations in Rabbit allow authors to declare concepts, relationships and instances. Rabbit declaration syntax is given in (114). Sentences (115) and (116) are examples of Rabbit declarations.

(114) <concept name> is a concept [, plural<plural name>]

(115) River is a concept.

(116) Sheep is a concept, plural Sheep.

Instances are introduced with the key phrase “is a”, for example,

(117) Australia is a Continent.

Relationships are defined with the following syntax

(118) <relationship name> is a relationship
For example,

(119) is the parent of is a relationship.

Instances and concepts are by convention capitalized, relationships are not.

Axioms describe the declared concepts. Axioms have the following syntax.

(120) [Every] <concept> <relationship clause> <object clause>

For example,

(121) Every Table is a kind of Furniture.

(122) Every River contains Water.

Finally, we have import statements which enable the use of concepts from other ontologies. Rabbit applies the use of OWL reasoners in detecting inconsistencies. Inconsistencies found are fed back into Rabbit for correction.

4.3.4 Computer Processable Language (CPL)

Computer Processable Language (CPL) (Clark et al. (2005)) sentences have the basic structure

subject + verb + complements + adjuncts,

such that complements are obligatory and adjuncts are optional modifiers. CPL grammar is made up of a number of syntactic categories, namely, prepositional phrases, compound nouns, ordinal modifiers, proper nouns, adjectives, relative clauses, and sentences. Simple sentences are conjoined with the keyword AND to form compound sentences.

CPL sentences are translated into a knowledge representation language which can be used for reasoning. The target knowledge representation language is Knowledge Machine (KM). Knowledge machine is a frame-based language with first-order semantics (Clark, Porter (1997)). CPL requires three steps for translating sentences to KM, namely, parsing, generation of an intermediate logical form and conversion of the intermediate logical form to statements in knowledge machine.

Syntactic parsing of CPL sentences is done with the aid of SAPIR, a GPSG wide coverage parser. During parsing, an intermediate logical form is generated.
Given a sentence such as (123), (124) is generated as the intermediate logical form

(123) The cat sat on the kitchen mat

(124) ((var ?x1 "the" "cat") (var ?x2 "the" "mat"
    (NN "kitchen" "mat") (s ?x1 "sit" (PP "on" ?x2))),

The intermediate logical form given in (124) is a simplified tree structure with logic-type elements which are generated by rules parallel to grammar rules such that variables are assigned to noun phrases in the sentence.

Finally, the intermediate logical forms such as (124) are used in generating knowledge machine assertions by recursively applying a set of simple rewrite rules. The generated KM assertions are binary clauses of the form $r(x, y)$, where each syntactic relation translates to a binary predicate. Sentence (123) for example, is translated to the following set of KM assertions.

(125) subject(_Sit1, _Cat1)
    'on'(_sit1, _Mat1)
    mod(_Mat1, _Kitchen1),

where _Sit1 represents an instance of sitting, _Cat1 represents an object cat, _Kitchen represents a kitchen and _Mat1 represents an object mat.

Spatiotemporal Extension to Computer Processable Language

The spatiotemporal extensions to CPL as defined by Murray, Singliar (2012) were developed to act as an interface to a surveillance data exploitation system with a machine learning algorithm (Inverse Reinforcement Learning) at its core. This surveillance system tracks vehicles, alerting a human analyst of abnormal tracks. Providing vehicle track information requires temporal and spatial data specific to the country/city of application given that normal vehicle behaviour is defined relative to day times such as work times, meal times as well as spatial features such as markets, town centres, etc.

The Inverse Reinforcement algorithm (IRL) as applied by this system does not consider spatial and temporal features as sharply demarcated. IRL provides a scaling system similar to fuzzy logic by assigning values between 0.0 and 1.0 to a given time in comparison to a temporal feature. For example, suppose the system
attempts to track vehicle behaviours during Jummah (Islamic prayer time) which is about noon on Fridays, 12:01 is assigned a value close to 1.0 as opposed to 13:10 which will be assigned a value closer to 0.0. Spatial features are treated similarly. As a result, the CNL interface to the surveillance system has two types of queries – proximity query and relational query.

Proximity query tells whether a track point satisfies particular temporal and spatial concepts, or comes close to satisfying them and if close, how close is it?

Relational query supports general Q&A, such that intervals are associated with events and places represent spatial settings rather than actual times and locations. The relational query attempts to provide inference based on Allen’s temporal algebra and Region Connection Calculus (Li, Ying (2004)).

4.3.5 E2V

Pratt-Hartmann (2003) proposes a slightly different approach in the definition E2V. While other languages such as ACE provide semantic interpretations from a defined set of natural language sentence forms, E2V defines a controlled language that translates to the two variable fragments of first-order logic. That is, the semantic interpretations of sentences of this language are members of the two variable fragment of first-order logic. Pratt-Hartmann is able to prove that the satisfiability of E2V is NEXPTIME complete.

The syntax of E2V defined with definite clause grammars permits the following syntactic categories – determiners, quantified nouns, transitive and intransitive verbs. Unlike other CNLs, E2V also permits the use of relative clauses, reflexives, and pronouns. Pratt-Hartmann, however, notes that the satisfiability problem of the language can easily become undecidable if the level restrictions on pronouns are slightly relaxed.

The syntax of E2V consists of four components – grammar, a lexicon, a movement rule and a set of indexing rules. The following is an example of E2V definite clause grammar rules.

(126) VP(A,I) → V(B,I,J), NP(B/A,J),

where labels VP, V, and NP are syntactic categories, the arguments A, B, B/A are semantic interpretations, I and J unify with indices which regulate variable bindings in the semantic interpretation. That is, the indexing rule ensures that variables assigned are specific and identifiable.
The lexicon is divided into two parts – closed class lexicon and open class lexicon. The closed class lexicon consists of words concerned with logical forms these include determiners, pronouns, reflexives, relative pronouns, and negation. The open class lexicon, in contrast, is an indefinitely large set of DCG rules for words of the noun and verb parts of speech.

From the syntax of E2V the semantic interpretation of a given sentence is a complex term formed from primitives occurring in the sentence. For example, sentence (127) is assigned the semantic interpretation in (128).

\[(127)\] Every artist despises every beekeeper.

\[(128)\] every(\(\text{artist}(x_1)\),every(\(\text{beekeeper}(x_2)\), \(\text{despise}(x_1, x_2)\))

Pratt-Hartmann defines a function \(T\) that translates E2V semantic interpretations like (128) to first-order logic sentences as seen below

\[(129)\] \(\forall x_1(\text{artist}(x_1) \rightarrow \forall x_2(\text{beekeeper}(x_2) \rightarrow \text{despise}(x_1, x_2)))\)

### 4.4 Discussion

Aristotle’s syllogisms are composed of a few relatively simple sentence forms which are easily translatable to some formal logic. We can, therefore, see how Aristotle’s syllogisms possess the four properties of controlled natural languages as enumerated by [Kuhn (2014)](Kuhn2014). Most CNLs are considered computer processable languages given that their member sentences have formal language interpretations. More recently, CNLs have become more expressive in the sense that they contain more natural and expressive sentence forms and as a result, corresponding semantic interpretations are more complex. With more expressive sentence forms, we are able to find interesting applications for CNLs, for example, software/hardware specification authoring, ontology authoring and query, question and answering systems etc.

In section 4.3 we considered a number of CNLs, their syntax and the semantic interpretation of their member sentences. Most of the considered CNLs are applied in domains such as specification authoring, ontology authoring, and query etc. as the semantic interpretations of these languages can be used by a reasoner to maintain consistency of a discourse. CNLs can also be considered natural language interfaces to formal languages, this perspective is made more obvious by
Pratt-Hartmann (2003) E2V which is a language that translates to two variable fragments of first-order logic. Unlike the other CNLs considered in section 4.3, E2V is not defined in response to a domain problem rather its sentence forms are restricted by the expressive power of their semantic representation. We, therefore, observe that a controlled natural language can either be defined as required by its domain of application as with ACE, PENG, Rabbit etc. or based on the semantic interpretations assigned to its member sentences as with Pratt-Hartmann’s E2V.

Most of the languages considered in the above survey are unable to provide explicitly the syntactic representation and corresponding semantic interpretation for temporal expressions. ACE, for example, provides some form of event order based on the order of occurrence of said events in a given ACE sentence. The spatiotemporal extension to CPL, of course, considers temporal relations, precisely those of the temporal algebra introduced by Allen (1983). This language, however, applies fuzzy logic in providing semantic interpretations to temporal nouns.

While Pratt-Hartmann’s E2V translates to two variable fragments of first-order logic, one can think of a controlled natural language such that member sentences have their semantic interpretations in first-order interval logic. That is, member sentences of this language express relationships in Allen’s temporal algebra where temporal intervals are treated as objects rather than assigning fuzzy logic scores as with the spatiotemporal extension to CPL. Defining such a language will require understanding natural language expressions that provide temporal information as well as relationships. We refer to such a language as a controlled natural language with temporal features.

4.5 Summary

Controlled natural languages are essentially natural languages with syntax that have corresponding formal representations. From the linguistic perspective, we are aware of the fact that the meaning of a given natural language text is subject to context or domain. As a result, many controlled natural languages are domain specific in order to provide unambiguous interpretations.

This research attempts however to include the meanings of temporal expressions to controlled natural languages. We find that there can be a number of
information expressed by controlled natural languages that are of temporal significance, for example, event ordering or weather reports. As with many other linguistic problems, the interpretation of temporal expressions is not without its challenges, we, however, apply a systematic research which provides us with a rather clear idea of how these temporal expressions behave in real life and can hence be applied in the development of our language. We discuss this and our results in the following chapter.
Chapter 5

Temporal Modifiers

We identify three types of temporal modifiers based on their syntactic structure, namely – temporal prepositions, temporal conjunctions, and temporal adverbials. Temporal prepositions are complemented by temporal noun phrases, for example, *in, during, at,* etc. Temporal conjunctions are complemented by sentential structures, for example, *while.* There are also a few modifiers that permit both nominal and sentential complements, they, therefore, possess properties of temporal prepositions as well as temporal conjunctions, examples include *before, after, since, until.* Lastly, the temporal adverbials considered in our language are syntactically temporal nouns that behave as modifiers, they specifically provide temporal contexts within which the sentence main event or state holds.

This chapter attempts to provide a description of the syntactic behaviour as well as the semantic interpretation of these temporal modifiers and how they interact with other temporal expressions – tense, aspects and aspectual classes. We will be making comparisons between our interpretations with those presented by other commentators such as Brée et al., 1993, Pratt-Hartmann, Francez (2001), Von Stechow (2002) etc.

The rest of this chapter is, therefore, organized as follows. Section 5.1 describes corpus analysis from which we are able to infer the syntactic and semantic behaviours of the various temporal modifiers considered in this chapter. Section 5.2 presents the semantic interpretation of simple tensed unmodified sentences. In section 5.3 we present the semantic interpretation of subordinate clauses and phrases. We discuss the syntactic representation and semantic interpretation of various temporal adverbials, temporal prepositions, and temporal conjunctions in sections 5.4, 5.5 and 5.6 respectively.
5.1 Corpus Analysis

The English language has various temporal prepositions and conjunctions, each with its own syntactic and semantic characteristics. When these modifiers interact with other temporal expressions (e.g. tense and aspects) in a given sentence, they generate varied interpretations. As a result, there is a need to understand how these modifiers behave and why they should be assigned a particular interpretation. We, therefore, need to consider these temporal modifiers in sentence constructs and observe their syntactic structure as well as their semantic interpretation.

Vendler (1967) defined four aspectual classes of verbs based on their behaviour in relation to the temporal context, namely, states, achievements, accomplishments, and activities. Based on the theory of tense as defined by Reichenbach (1947) we have three tenses in English – past, present, and future. Reichenbach’s tenses combine with English aspects – perfect and progressive to produce twelve tense and aspect constructions. We, therefore, examine the various configuration of these temporal expressions by systematically generating sentences by first having the possible aspectual class combination between the main and subordinate clause, these produce sixteen possible combinations. For each of these sixteen possible aspectual class combination, we generate the possible tense/aspect constructions between the main and subordinate clause, this produces one hundred and forty-four construction for each of the aspectual class combination, giving us a total of 2,304 sentences of the possible aspectual class, tense and aspect configuration. The last linguistic factor we consider is the verbal negation. Having the possible verbal polarity between the main and subordinate clause generates 9,216 sentences. For temporal prepositions, the combination of aspectual classes, tense and aspect in the main clause produces sixty possible constructions given that they are complemented by temporal nouns rather than sentences.

Our generated sentences provide an overview of how the various temporal modifiers behave in different tense/aspect constructions. We, however, cannot assume that all possible scenarios are catered for from our generated dataset. We therefore extract two hundred sentences for each preposition and conjunction which we analyse with the aim of finding patterns such as tense agreement, syntactic structure of nominal and sentential complements, distinguishing between temporal instances and other uses of the modifier of interest (e.g. spatial) and a few other observations not apparent from the dataset we have generated. Our
corpus analysis is similar to that carried out by Brée, Smit (1986).

5.2 Tensed Sentences

In this section, we present the syntactic structure and the corresponding semantic interpretation of simple tensed sentences. Consider the following sentence.

(130) Mary kissed John.

Grammar (131) is a semantically annotated grammar that generates sentence (130), where the semantic annotation on the syntactic categories Tense, PN and V, are assigned the same values as the semantic interpretation as the terminal symbol in the body of the rule.

(131) \[ S/\lambda I[\varphi(\psi)] \rightarrow NP/\varphi, VP/\psi \]
\[ NP/\varphi \rightarrow PN/\varphi \]
\[ V'/\varphi(\psi) \rightarrow V/\varphi, NP/\psi \]
\[ VP/\varphi(\psi) \rightarrow Tense/\varphi, V'/\psi \]
\[ Tense/\varphi \rightarrow past/\lambda T \lambda x_6[\exists i_1(T(i_1)(x_6) \land (i_1 < \text{now}) \land (i_1 \subseteq I))] \]
\[ NP/\varphi \rightarrow \text{John}/\lambda P[P(\text{john})] \]
\[ PN/\varphi \rightarrow \text{Mary}/\lambda Q[Q(\text{mary})] \]
\[ V/\varphi \rightarrow \text{kissed}/\lambda x_3 \lambda i_2 \lambda x_4[x_3(\lambda x_5[\text{kiss}(x_4, x_5)(i_2)])] \]

Given the semantically annotated grammar rules in (131), we can recursively generate the semantic interpretation of sentence (130). First we generate the semantic interpretation of the category \( V' \) by applying the semantic interpretation of category \( V \) as seen below.

(132) \[ [v, \text{kissed John}] = [v \text{kissed}][[\text{np, John}]] = \]
\[ \lambda x_3 \lambda i_2 \lambda x_4[x_3(\lambda x_5[\text{kiss}(x_4, x_5)(i_2)])](\lambda P[P(\text{john})]) = \]
\[ \lambda i_2 \lambda x_4[(\text{kiss}(x_4, \text{john})(i_2))] \]

Having generated the semantic interpretation of the category \( V' \) in (132), we generate the semantic interpretation of a verb phrase (VP) by applying the semantic interpretation of Tense to the semantic interpretation of \( V' \) as seen below.

(133) \[ [\text{vp, kissed John}] = [\text{tense past}][[v, \text{kissed john}]] = \]
\[ \lambda T \lambda x_6[\exists i_1(T(i_1)(x_6) \land (i_1 < \text{now}) \land (i_1 \subseteq I))](\lambda i_2 \lambda x_4[x_3(\text{kiss}(x_4, \text{john})(i_2))] = \]
\[ \lambda x_6[\exists i_1(\text{kiss}(x_6, \text{john})(i_1) \land (i_1 < \text{now}) \land (i_1 \subseteq I))] \]
Finally, we generate the semantic interpretation of the sentence \( S \) by applying the semantic interpretation of the noun phrase (NP) to the semantic interpretation of the verb phrase (VP) and lambda abstracting the free variable \( I \) as seen below.

\[
(134) \ [s \text{ Mary kissed John}] = \lambda I[\lambda Q[Q(\text{mary})](\lambda x_6[\exists i_1(\text{kiss}(x_6, \text{john})(i_1) \land (i_1 < \text{now}) \land (i_1 \subseteq I)])] = \\
\lambda I[\exists i_1(\text{kiss}(\text{mary}, \text{john})(i_1) \land (i_1 < \text{now}) \land (i_1 \subseteq I))]
\]

Figure 5.1 shows the parse tree and the generation of the semantic interpretation of sentence (130) given the grammar rules in (131).

We observe in the parse tree in Figure 5.1 that tense is treated as a lexical item with a corresponding semantic interpretation. Its assigned semantic interpretation ensures the appropriate temporal order of the time of event (\( i_1 \)) and the time of speech (\( \text{now} \)). It also provides quantification of the event time. We will see in sections 5.5 and 5.6 that the semantic interpretation assigned to tense changes depending on the temporal modifier with which it interacts.

However, from our set of generated sentences as well as sentences extracted from the Brown corpus, we observe that grammatical aspects do not have specific
semantic interpretations as tenses do. In many cases, their syntactic roles and semantic implications are dependent on the temporal modifiers with which they interact. Consider the following sentences.

(135) John had arrived during the meeting.

(136) John has been sleeping since noon.

(137) John is arriving today.

Several linguists such as Dowty (1979), Richards (1982), Moens, Steedman (1988) describe the perfect as a temporal operator that places focus on the result state of a given culminated or terminated event. While we do not claim this view of the perfective is incorrect, the perfective and indeed every other grammatical aspect construction does not seem to have a static interpretation in all cases. For example, from the above sentences, the perfective in sentence (136) as observed by Brée (1985) is more of a syntactic marker required to distinguish between the temporal and non-temporal use of the preposition *since* and not necessarily of much semantic consequence. The progressive similarly tends to have varied behaviours, for example, sentence (137) does not describe an ongoing event relative to some time of reference as Reichenbach’s interpretation of progressives will suggest, but rather an event yet to occur or to signify an intention. We, therefore, do not have interpretations for the perfect and the progressive as lexical items as we have for tense. We instead provide varied interpretations for the different interactions of tense, aspects and temporal modifiers based on patterns observed from the analysis of our set of generated sentences and Brown corpus sentences.

5.3 Subordinate Clauses and Phrases

Subordinate clauses of temporal conjunctions are syntactically similar to simple tensed sentences; they are however assigned different semantic interpretations. In most cases, temporal conjunctions enforce tense agreement between the main and subordinate clause verbs. Providing an interpretation for a sentence modified by a temporal conjunction will, therefore, result in some redundancy if we are to include tense interpretations to both clauses. As a result, we omit tense interpretation from the semantic interpretation of subordinate clauses. Consider the following sentence.
5.3. SUBORDINATE CLAUSES AND PHRASES

(138) Mary arrived

We assign the semantic interpretation [139] to [138].

(139) $\lambda I[\exists i(arrive(mary)(i) \land (i < \text{now}) \land (i \subseteq I))]$

Suppose we have sentence [138] as a subordinate clause of a temporal conjunction as in sentence [140], we assign the semantic interpretation [141] to it as subordinate clause.

(140) John left before Mary arrived,

(141) $[\lambda s\text{Mary arrived}] = \lambda Q[\exists i(arrive(mary)(i) \land Q(i) \land (i \subseteq I))],$

where the lambda-abstracted variable $Q$ enables application to the interpretation of the temporal modifier and free variable $I$ is the temporal context. Subordinate clauses such as [141] are assigned the semantic type $(i,t,t)$ – that is, functions from functions from intervals to Boolean to Boolean.

Temporal prepositional phrases and temporal adverbials often contain temporal nouns. We differentiate three types of temporal nouns, namely – interval nouns, event nouns, and temporal proper nouns. Interval nouns simply describe temporal intervals, for example, day, hour. Event nouns describe occurrences, for example, meeting, lecture. Temporal proper nouns are like proper nouns as they name temporal intervals, for example, noon. Interval and event nouns behave like common countable nouns and are often preceded by determiners. We, therefore, have a set of temporal determiners which combine with interval and event nouns to form temporal noun phrases as required by the following rule.

(142) $NP/\varphi(\psi) \rightarrow Det/\varphi, N/\psi,$

where the category $NP$ is a temporal noun phrase provided $Det$ and $N$ is a temporal determiner and temporal noun (interval or event nouns) respectively.

The main difference between non-temporal and temporal determiners is their semantic types. While non-temporal determiners are assigned the semantic type $((e,t),(e,t),t))$ – that is, functions from functions from objects to Boolean to functions from functions from object to Boolean to Boolean. Temporal determiners are assigned the semantic type $((i,t),((i,t),t))$ – that is, functions from functions from intervals to Boolean to functions from functions from intervals to Boolean to Boolean. Given therefore the determiner every, its non-temporal use is interpreted as [143] while its temporal use is interpreted as [144].
CHAPTER 5. TEMPORAL MODIFIERS

(143) \[[\text{det} \text{ every}]\] = \(\lambda P \lambda Q[\forall x(P(x) \to Q(x))]\)

(144) \[[\text{det} \text{ every}]\] = \(\lambda P \lambda Q[\forall i(P(i) \land (i \subseteq I) \to Q(i))]\)

where the free variable \(I\) is the temporal context. We reserve variables \(i\) and \(j\) to represent temporal intervals. The interpretations in (145) and (146) are the semantic interpretations of temporal use of indefinite and definite determiners respectively.

(145) \[[\text{det} \text{ a}]\] = \(\lambda P \lambda Q[\exists i(P(i) \land (i \subseteq I) \land Q(i))]\)

(146) \[[\text{det} \text{ the}]\] = \(\lambda P \lambda Q[\exists i(P(i) \land \forall j(P(j) \to (i = j)) \land (i \subseteq I) \land Q(i))]\)

Similarly, temporal nouns are assigned different variables from regular nouns, where the semantic interpretations of temporal nouns are assigned the semantic type \((i,t)\), for example, (147), while non-temporal nouns are assigned the semantic type \((e,t)\), for example, (148).

(147) \[[\text{n} \text{year}]\] = \(\lambda j[\text{year}(j)]\)

(148) \[[\text{n} \text{student}]\] = \(\lambda x[\text{student}(x)]\)

Suppose we are given the temporal noun phrase \textit{every year}, from rule (142), we can generate a corresponding semantic interpretation as given below.

(149) \[[\text{np} \text{ every year}]\] = \[[\text{det} \text{ every}]\][\[[\text{n} \text{year}]\]]

\(\lambda P \lambda Q[\forall i(P(i) \land (i \subseteq I) \to Q(i))][\lambda j[\text{year}(j)]] = \lambda Q[\forall i(\text{year}(i) \land (i \subseteq I) \to Q(i))]\)

Temporal proper nouns on the other hand simply generate temporal noun phrases as required by rule (150). The semantic interpretation of the head of the rule is therefore assigned the same as the body’s.

(150) \(NP/\varphi \to N/\varphi\),

where \(N\) is a temporal proper noun. Given therefore the temporal proper noun \textit{noon}, we interpret it as follows.

(151) \([\text{np noon}]\] = \([\text{n noon}] = \lambda P[P(\text{noon}) \land (\text{noon} \subseteq I)]\)

The semantic interpretation of temporal noun phrases are assigned the semantic type \(((i,t),t)\) – that is, they are functions from functions from intervals to Boolean to Boolean. Table 5.1 provides the list of symbols for the various syntactic categories encountered in this chapter.
5.4 Temporal Adverbials

By Temporal adverbials, we refer exclusively to those temporal modifiers that are not preceded by temporal prepositions or conjunctions, for example,

(152) Maurice plays basketball every day.

(153) Maurice went to the hospital yesterday.

(154) Coldplay are performing tonight.

Over the course of this chapter, we will observe that various temporal modifiers introduce different temporal relations between intervals. However, from the above sentences, temporal adverbials act as temporal containers within which a sentence’s main event or state holds. In this section, we attempt to provide an overview of the syntactic structure of temporal adverbials as well as their semantic interpretations and how they combine with unmodified sentences.

Since they are not preceded by conjunctions or prepositions temporal adverbials considered are syntactically temporal noun phrases, where the head noun is an interval or an event noun. A temporal adverb (TempAdv) is therefore generated as required by the following rule.

(155) TempAdv/φ → NP/φ,

where NP is a temporal noun phrase generated by rule (142). The corresponding semantic interpretation of temporal adverbs is the same as that of the temporal noun phrase its is generated from. Given therefore the temporal noun phrase every year the temporal adverbial generated is assigned the same semantic interpretation given in (149) repeated as (156).
We can now generate a sentence modified by a temporal adverbial given the following production rule.

\[(157) \; S/\lambda I[\varphi(\psi)] \rightarrow S/\psi, \; TempAdv/\varphi\]

Suppose we have an unmodified sentence as in (158) interpreted as (159),

\[(158) \; \text{Every student wrote an exam.}\]

\[(159) \; \left[\left[\text{s} \; \text{every student wrote an exam}\right]\right] = \lambda I[\forall x(\text{student}(x) \rightarrow \exists i \exists y (\text{exam}(y) \land \text{write}(x, y)(i) \land (i < \text{now}) \land (i \subseteq I)))] ,\]

we can generate the semantic interpretation of sentence (160) as seen in (161) by applying the semantic interpretation of the temporal adverb \textit{every year} as given in (156) to the semantic interpretation of the unmodified sentence given in (159) and lambda-abstracting the free variable \(I\).

\[(160) \; \text{Every student wrote an exam every year.}\]

\[(161) \; \left[\left[\text{s} \; \text{every student wrote an exam every year}\right]\right] = \lambda I[[\text{tempAdv} \; \text{every year}]]\left[\left[\text{s} \; \text{Every student wrote an exam}\right]\right] = \lambda I[\lambda Q[\forall i(\text{year}(i) \land (i \subseteq I) \rightarrow Q(i))]] = \lambda I[\forall i(\text{year}(i) \land (i \subseteq I) \rightarrow \forall x(\text{student}(x) \rightarrow \exists i \exists y (\text{exam}(y) \land \text{write}(x, y)(i) \land (i < \text{now}) \land (i \subseteq I)))] ] = \lambda I[\exists i_1 \exists y (\text{exam}(y) \land \text{write}(x, y)(i_1) \land (i_1 < \text{now}) \land (i_1 \subseteq i))]] \]

Temporal adverbials have similar semantic interpretations as the temporal preposition \textit{during}, temporal conjunction \textit{while} and the locative temporal prepositions we will consider later in this chapter. There are however other temporal relations that can be expressed in natural language. The English language provides other temporal modifiers (prepositions and conjunctions) that express these relations.

### 5.5 Temporal Prepositions

Temporal prepositions are modifiers that are exclusively complemented by temporal noun phrases. The generation of temporal prepositional phrases is therefore
governed by the rule (162) such that the semantic interpretation of the generated temporal prepositional phrase is derived by applying the semantic interpretation of the constituent temporal preposition to the semantic interpretation of the complement noun phrase.

\[(162) \quad \text{TPP}/\varphi(\psi) \rightarrow \text{TP}/\varphi, \text{NP}/\psi.\]

We consider a number of temporal prepositions in this section, namely – during, in, on, at, for and by, before, after, since, and until. The syntactic representation and semantic interpretations of these temporal prepositions have been considered by several linguists. We will consider some of the interpretations in literature in comparison to ours as required by a controlled natural language with temporal features.

Although some prepositions such as during have only the temporal use, several prepositions can have different uses other than temporal (e.g., spatial, inferential). In this section, we attempt to describe syntactic markers that distinguish the temporal use of a given preposition from its non-temporal uses.

### 5.5.1 During

During can be said to serve as a temporal container just like temporal adverbials within which the main clause event or state holds. Consider the following sentences.

(163) Mark left during the meeting.

(164) Mary spoke during the conference.

(165) Jake wrote a novel during the war.

(166) Peter owned industries during the depressions.

The above sentences are examples of verbs of each of Vendler’s aspectual classes in interaction with the during prepositional phrase. Although we see that when in the past tense, it is possible to have verbs from each of these aspectual classes produce grammatical sentences in the context of a during phrase, other tense-aspect constructions might, however, affect grammatical correctness. We, therefore, consider the syntactic structure of during sentences based on our analysis of Brown corpus sentences and our set of generated sentences.
Syntactic Analysis

From our analysis we observe that *during* has only a temporal use. As a result, it is mostly complemented by temporal noun phrases. Given the three classes of temporal nouns, we find that *during* is often complemented by event and interval noun phrases and seldom temporal proper nouns, this is illustrated by the following sentences.

(167) Mark kissed Mary during every meeting.


(169) The minister will meet with the president during the week.

<table>
<thead>
<tr>
<th></th>
<th>Temporal Noun</th>
<th>Temporal Proper Noun</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Interval</td>
<td>Event</td>
</tr>
<tr>
<td>Definite</td>
<td>72</td>
<td>44</td>
</tr>
<tr>
<td>Indefinite</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>Universal</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Deictic</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>No determiner</td>
<td>6</td>
<td>13</td>
</tr>
<tr>
<td>Possessive</td>
<td>7</td>
<td>12</td>
</tr>
</tbody>
</table>

Table 5.2: Data of the Syntactic Structure of During Complements

Sentences (167), (168), and (169) show *during* being complemented by event nouns, temporal proper nouns and interval nouns respectively. We observe that a sentence such as (168) appears unnatural, this is evident from our study of the Brown Corpus sentences, where only 3% of the extracted *during* sentences have year complements. Although one can think of a sentence like (170) which can be considered somewhat natural. Examining sentences of a bigger corpus might provide more instances of such instances.

(170) The Democrats lost votes during 1987.

However, we have 51% of the sentences complemented by interval nouns, 41.5% are complemented by event nouns. *During* often attempts to select a particular interval within which it places the main clause event. As a result 58% of the Brown corpus *during* sentences have temporal noun complement preceded by definite determiners, 8% have temporal nouns with indefinite determiners, 2% have
temporal nouns with universal determiners and the rest had possessives, deictic
derminers or no determiners. Table 5.2 provides a data summary of the struc-
ture of during complements. In a similar study, considering during complements
with definite and indefinite determiners, Breé observed that 90% of sentences
considered had definite determiner. In our study, considering a set of sentences
with event and interval nouns, 85% were found to have definite determiners the
rest having universal and indefinite determiners.

Given a during sentence in the past tense, the during phrase provides a tem-
poral context within which the main clause event occurred; this behaviour is
retained in future-tensed sentences. The present tense, on the other hand, de-
cribes several episodes of the described event.

(171) John had arrived during the meeting

The progressive when in the present tense, for example, sentence (172), describes
an event in the future.

(172) The Vice Chancellor is speaking during the seminar

Table 5.3 summarizes the tense, aspect and aspectual combination encountered
from our analysis of the Brown corpus sentences. We provide similar tables for

<table>
<thead>
<tr>
<th></th>
<th>Past</th>
<th>Present</th>
<th>Future</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Perf</td>
<td>Prog</td>
<td>PerfProg</td>
</tr>
<tr>
<td>Activity</td>
<td>1</td>
<td>9</td>
<td>3</td>
</tr>
<tr>
<td>Achievement</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Accomplishment</td>
<td>2</td>
<td>12</td>
<td>1</td>
</tr>
<tr>
<td>State</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.3: Data of the Tense, Aspect, Aspectual Class Configuration During Main Clauses
the rest of the temporal prepositions/conjunctions considered in this chapter. In some cases the tables are relegated to the appendix for the sake of readability.

**Semantic Interpretation**

From our analysis we observe that the preposition *during* permits only event and interval nouns as complements. These temporal nouns are like common countable nouns and are hence preceded by determiners, for example,

(173) John kissed Mary during the meeting.

(174) John kissed Mary during a meeting.

(175) John kissed Mary during every meeting.

Sentences (173), (174) and (175) are examples of *during* interacting with different temporal determiners. As discussed in section 5.3 we generate temporal noun phrases with the production rule (142) repeated in (176). The temporal noun phrase the lecture is therefore interpreted as seen in (177).

(176) \[
NP/\varphi(\psi) \rightarrow Det/\varphi, N/\psi,
\]

(177) \[
\begin{align*}
&\text{\{the lecture\} = \{det the\}(\{a lecture\}) =} \\
&\lambda\Phi[\lambda Q[\exists i(P(i) \land \forall j(P(j) \rightarrow (i = j)) \land (i \subseteq I) \land Q(i))](\lambda i_1[\text{lecture}(i_1)])] = \\
&\lambda Q[\exists i(\text{lecture}(i) \land \forall j(\text{lecture}(j) \rightarrow (i = j)) \land (i \subseteq I) \land Q(i))]
\end{align*}
\]

The semantic interpretation in (177) is assigned the semantic type \((i,t)\) – that is, a function from intervals to booleans. We therefore require an interpretation for *during* that combines with the type \((i,t)\). Our interpretation of the temporal preposition *during* is given as follows.

(178) \[
\begin{align*}
&[\text{tp during}] = \lambda\Psi\lambda\Phi[\Psi(\lambda i[\Phi(i)])]
\end{align*}
\]

The above interpretation of *during* is assigned the semantic type \(((i,t),(i,t),t))\) – that is, it is a function from functions from intervals to Boolean to functions from functions from interval to Boolean to Boolean. The proposition \(\Phi\) holds within the temporal interval \(i\) which is eventually existentially quantified when it combines with a temporal noun phrase to generate a temporal prepositional phrase as required by rule (179) with corresponding semantic interpretation given in (180).
We can now generate the interpretations of during sentences by combining the interpretation of an unmodified sentence and the interpretation of the during phrase as required. The semantic interpretation of the modified sentence is generated by applying the semantic interpretation of the temporal prepositional phrase to the semantic interpretation of the unmodified sentence and lambda-abstracting the free variable.

(181) \( S/\lambda I[\varphi(\psi)] \rightarrow S/\psi, \ TPP/\varphi \)

Given the sentence (182) with interpretation (183),

(182) John wrote an essay

(183) \([s \text{ John wrote an essay}] = \lambda I[\exists i \exists x (\text{essay}(x) \land \text{write(john, } x\text{)}(i) \land (i < \text{now}) \land (i \subseteq I))],\)

we can generate the semantic interpretation of sentence (184) in (185) by applying the semantic interpretation of the temporal prepositional phrase as given in (180) to the semantic interpretation of the unmodified sentence given in (183) and lambda-abstracting the free variable \( I \).

(184) John wrote an essay during the lecture.

(185) \([s \text{ John wrote an essay during the lecture}] = \lambda I[[tpp \text{ during the lecture}][[s \text{ John wrote an essay}]]) = \lambda I[\lambda \Phi[\exists i_1 (\text{lecture}(i_1) \land \forall j (\text{lecture}(j) \rightarrow (i_1 = j)) \land (i_1 \subseteq I) \land \Phi(i_1)))] = \lambda I[\exists i_1 (\text{lecture}(i_1) \land \forall j (\text{lecture}(j) \rightarrow (i_1 = j)) \land \exists i \exists x (\text{essay}(x) \land \text{write(john, } x\text{)}(i) \land (i < \text{now}) \land (i \subseteq i_1) \land (i_1 \subseteq I))]]}
5.5.2 The Locative Temporal Prepositions *In, On* and *At*

Locative temporal prepositions have the common property of providing the temporal location for a given event. When used in the spatial context, they describe quite specifically the spatial position of a given object relative to a reference object. Consider the following sentences.

(186) Mark gave a speech at the university.

(187) Mark gave a speech in Leeds.

(188) Mark gave a speech on the train.

Similarly, the locative temporal prepositions provide the temporal location of a given event within the time of reference provided by the temporal preposition’s complement noun phrase as shown in the following sentences.

(189) John arrived at noon.

(190) John arrived in January.

(191) John arrived on Tuesday.

Sentences (189), (190), (191) all describe the temporal context within which the main clause event occurs. But we know intuitively as English speakers that these prepositions cannot be used interchangeably. We, therefore, need to identify syntactic markers that determine what preposition is appropriate in a given sentence.

The locative prepositions, unlike *during*, tend to favour temporal proper nouns as complements, we observe that they select exactly one identifiable interval in which the sentence event holds. When they are complemented by event noun phrases, we mostly get the spatial context as in (192) or an unnatural reading as in (193).

(192) John arrived at the meeting

(193) ?John arrived on the meeting

Similarly, interval nouns produce unnatural sentences as in sentence (194). We would, however, get a temporal context if the interval noun head of the complement noun phrase has a pre-modifier as observed in sentence (195) and (196).

(194) ?John arrived on the meeting
5.5. TEMPORAL PREPOSITIONS

(194) John arrived at the semester.

(195) John arrived at the end of the semester.

(196) John arrived on the last Thursday of January.

In, as observed by Brée, Smit (1986), confines the main state or event in a reduced temporal space. That is, \textit{in} enforces some temporal inclusion of the main clause within a temporal context. \textit{On} behaves similarly, although it appears the temporal reference introduced by an \textit{on} phrase is relatively smaller than \textit{in’s}.

We have noted that despite the similarity in their semantic functions, we intuitively know that these locative prepositions cannot be used interchangeably. We also know their complements are mostly temporal proper nouns. However, different temporal proper nouns describe different sizes of intervals. For example, a temporal noun such as \textit{1956} describes a whole year as opposed to \textit{14:45} which describes a given time accurate to the minute. These two temporal proper nouns complement two different locative prepositions.

\textit{In} is adapted for temporal proper nouns that describe larger intervals ranging from months to millennia. The reason for this might be observed from its spatial use. \textit{In} when used to convey the spatial context describes an object within a larger container, hence in the temporal context, the preposition tends to choose intervals large enough to act as temporal containers within which a given event occurred as in (197) and (198).

(197) Russia will host the world cup in 2018.

(198) Terrence graduated in December.

\textit{On} has similar semantic interpretation with \textit{in}, it has however been adapted to permit days of the week and dates as complements. Consider the following sentences.

(199) Dudley played golf on Sunday.

(200) Dudley will play on the 14th of March.

\textit{At} selects relatively tiny granules of time, ranging from seconds to hours as in sentence (201), this is similar to Bree’s description, as he considered \textit{at} as a preposition expressing exact positions. We, therefore, claim that \textit{at} provides temporal identification for the main clause event.
CHAPTER 5. TEMPORAL MODIFIERS

(201) The train departs at 19:43.

We mentioned briefly that these locative prepositions permit pre-modified nouns as complements. Consider the following examples.

(202) Christianity became popular in the fourteenth century.

(203) In the first year of Obama’s presidency, the hope of an improved economy was high

(204) Jose arrived at the end of the match.

The above sentences show that locative prepositions are permitted to have interval and event nouns as complements provided they are pre-modified and preceded by a definite determiner given a past or future tensed main verb. These pre-modifiers select specific sub-intervals of the interval described by the temporal noun.

Table 5.4 provides a data summary of the syntactic structures of the complements of temporal in, on and at of a collection of a hundred sentences each. Where def, indef is the number of complements with definite and indefinite determiners respectively. Deictic represents the number of sentences with deictic determiners such as this. The table shows locative prepositions are only complemented by noun phrases with a preceding definite determiner. In this cases, the noun phrase has pre-modified noun heads. We also see that on is complemented by dates, at is complemented by day times and in is complemented by years and months.

<table>
<thead>
<tr>
<th>Temporal Noun</th>
<th>TPN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Def</td>
</tr>
<tr>
<td>on</td>
<td>36</td>
</tr>
<tr>
<td>at</td>
<td>33</td>
</tr>
<tr>
<td>in</td>
<td>48</td>
</tr>
</tbody>
</table>

Table 5.4: Data of the Syntactic Structures of In, On and At Complements

The preposition in has another use different from its locative use discussed in this section. Consider the sentence below,

(205) Joshua finished his thesis in 2 months.

We consider this use in more detail in section [5.5.3]
5.5. TEMPORAL PREPOSITIONS

Interpretation of The Locative Prepositions *In, On and At*

From the study of the syntactic behaviours of the locative prepositions *in*, *on* and *at*, we have also observed that their semantic interpretations are similar given that they provide temporal contexts or identity to their main clause event or state. This view of the locative prepositions is in turn similar to our interpretation of *during*.

Rule (150) repeated in (206) generates a noun phrase from a temporal proper noun. Given therefore a temporal proper noun such as *noon*, with semantic interpretation (207), we assign the noun phrase symbol at the head of the rule the same semantic interpretation as the proper noun in the body of the rule.

(206) $NP/\phi \rightarrow N/\phi,$

(207) $[[np \text{ noon}]] = [[n \text{ noon}]] = \lambda P[P(\text{noon}) \land (\text{noon} \subseteq I)].$

We then assign interpretation (208) to each of the locative prepositions.

(208) $\lambda \Psi \lambda \Phi[\Psi(\lambda i[\Phi(i)])]$  

Given that a temporal prepositional phrase is formed from the combination of a temporal preposition and a noun phrase, rule (209) generates a temporal prepositional phrase from the combination of a locative temporal preposition and temporal proper noun.

(209) $TPP/\phi(\psi) \rightarrow TP/\phi, NP/\psi,$

Here, $TP$ is a locative temporal preposition and $NP$ is a temporal proper noun. The semantic interpretation of the temporal prepositional phrase *at noon* is therefore generated as seen below.

(210) $[[tpp \text{ at noon}]] = [[t \text{ at}][[np \text{ noon}]] = \\
\lambda \Psi \lambda \Phi[\Psi(\lambda i[\Phi(i)])](\lambda P[P(\text{noon}) \land (\text{noon} \subseteq I)]) = \\
\lambda \Phi[\Phi(\text{noon}) \land (\text{noon} \subseteq I)]$

Suppose we have an unmodified sentence such as (211) with its semantic interpretation given in (212) below,

(211) John arrived.

(212) $[[s \text{ John arrived}]] = \\
\lambda I_1[\exists i_1(\text{arrive(john)}(i_1) \land (i_1 < \text{now}) \land (i_1 \subseteq I_1))].$
We generate the semantic interpretation for a modified sentence (214) in (215) by applying the semantic interpretation of the temporal prepositional phrase as given in (210) to the semantic interpretation of the unmodified sentence as given in (212) and lambda-abstracting the free variable $I$ as required by rule (181) repeated in (213).

$$S/\lambda I[\varphi(\psi)] \rightarrow S/\psi, \ TPP/\varphi$$

(213) John arrived at noon

(214) John arrived at noon

$$[[s \text{ John arrived at noon}]] = \lambda I[[tpp \text{ at noon}][[[s \text{ John arrived}}])] = \\
\lambda I[\lambda \Phi[\Phi(\text{noon}) \land (\text{noon} \subseteq I)]] \\
(\lambda \Phi[[\exists i_1 (\text{arrive(john)}(i_1) \land (i_1 < \text{now}) \land (i_1 \subseteq I_1))]]) = \\
\lambda I[[\exists i_1 (\text{arrive(john)}(i_1) \land (i_1 < \text{now}) \land (i_1 \subseteq \text{noon}) \land (\text{noon} \subseteq I))]]$$

The semantic interpretations of sentences complemented by in and on phrases are generated similarly.

### 5.5.3 Durative Prepositions For and In

In section 5.5.2 we considered the locative use of in, in this section, we present its durative use. Like many other temporal prepositions, for and in can provide contexts other than the temporal. We, however, observe that these prepositions produce temporal contexts (in this case durative) when complemented by durative nouns as in sentences (216) and (217) below.

(216) Mark finished the puzzle in 3 minutes.

(217) The Bishop spoke for 2 hours.

The durative noun complements of for and in are of a particular syntactic structure. We often have the complement temporal noun phrases with numerical determiners as in sentences (216) and (217). From the Brown corpus extracted sentences, we observe that when a durative noun has an indefinite determiner preceding it, then it refers to a singular unit of the interval represented by the durative noun.

(218) The Bishop spoke for an hour.

(219) Mark finished the puzzle in a minute.
5.5. TEMPORAL PREPOSITIONS

The most obvious difference between these durative prepositions is the aspectual class of their main clause. Dowty (1979), in fact, used these prepositions as a test in determining the aspectual class of sentence verbs. Similarly, we observe that the durative *in* is compatible with culminating main verbs, while *for* permits only non-culminating main verbs and accomplishment, but rejects achievements as main verbs. Brée et al. (1993) supports this disparity by observing that *for* asserts that the main event or state holds over all sub-intervals of the referenced interval introduced by the *for* phrase, hence its main event is required to be durative. *In*, on the other hand, asserts that at some sub-interval within the reference interval, the main event occurred. *For* therefore enforces a universal quantification over the reference duration while *in* enforces an existential quantification.

The progressive in the present often produces a future-tensed reading. Hence sentences (220) and (221) have similar semantic interpretations.

(220) The Bishop is speaking for an hour.

(221) The Bishop will be speaking for an hour.

The *in* preposition with the past progressive is not very natural as seen in the following sentence.

(222) *The Bishop was speaking in an hour.

This is because the progressive produces a durative aspect which is not compatible with the durative *in*. Although one can think of a sense where the past progressive can be used to indicate intent, our proposed CNL however does not handle this use of the past progressive. In the present and future tense, however, the progressive provides information of the amount of time required for the event to commence from the time of speech as seen in sentences (223) and (224). this use of *in* is what Brée refers to as *end of time extent*. Note that *in* has a similar interpretation in the present progressive and future tenses.

(223) The Bishop is speaking in an hour.

(224) The Bishop will be speaking in an hour.

*In* only permits the perfective and punctual verbs when in the future tense as seen in sentence (227).
(225) *The Bishop had spoken in an hour.

(226) *The Bishop has spoken in an hour.

(227) The Bishop will have finished his speech in an hour.

The future perfect as in sentence (227) is compatible with in because it is interpreted as the main clause being completed within the duration specified by the in phrase. For on the other hand provides the duration of a given event as at a time of reference. As a result, when for is in the perfect, the sentence complement is usually an embedded preposition except in the present tense where the time of reference is equal to the time of speech as in the following sentence.

(228) The Bishop has spoken for an hour.

**Semantic Interpretation**

Given that complements of the durative prepositions in and for are often numerical, we need to specifically consider the interpretation of durative noun phrases. Suppose we are given a durative noun phrase such as 2 hours, we assign it the following semantic interpretation.

\[
\begin{align*}
\text{[np 2 hours]} &= \lambda Q[\exists i_1 (\text{dur}(i_1, 2\text{hour}) \land Q(i_1) \land (i_1 \subseteq I))] \\
\text{dur} &= \text{a binary predicate that takes an interval and its duration as arguments. In the above semantic interpretation, } \text{dur}(i_1, 2\text{hour}) \text{ asserts that the duration of variable } i_1 \text{ is 2 hours.}
\end{align*}
\]

Consider the following sentences.

(230) John spoke for 2 hours

(231) John returned in 2 hours

From the discussion thus far, the semantic interpretation of sentence (230) requires a universal quantification over the sub-intervals of the reference interval. The semantic interpretation of sentence (231) on the other hand, requires an existential quantification. The semantic interpretation of sentences (230) and (231) are therefore given in (232) and (233) respectively.

\[
\begin{align*}
\text{[s John spoke for 2 hours]} &= \lambda I[\exists i (\text{dur}(i, 2\text{hour}) \land \forall j ((j \subset i) \rightarrow \text{speak(john)}(j) \land (j < \text{now})) \land (i \subseteq I))] \\
\end{align*}
\]
5.5. TEMPORAL PREPOSITIONS

\[(233) \text{[}s \text{ John returned in 2 hours} = \lambda I [\exists i (\text{dur}(i, 2\text{hour}) \land \exists j (\text{return}(\text{john})(j) \land (j < \text{now}) \land (j \subset i) \land (i \subseteq I))]\]

Let us consider how semantic interpretations (232) and (233) can be systematically generated. Like with the locative prepositions and during, the durative prepositions in and for provide containment relationship. We therefore, simply assign a similar semantic interpretation for both prepositions.

\[(234) \text{[} \text{tp in} \text{]} = \lambda \Psi \lambda \Phi [\Psi(\lambda i [\Phi(i)])] \]

\[(235) \text{[} \text{tp for} \text{]} = \lambda \Psi \lambda \Phi [\Psi(\lambda i [\Phi(i)])] \]

Given the production rule (236) for temporal prepositional phrases, we generate a corresponding semantic interpretation for in and for phrases in (237) and (238) respectively.

\[(236) \text{TPP} / \varphi(\psi) \rightarrow \text{TP} / \varphi, \text{NP} / \psi \]

\[(237) \text{[} \text{tp in 2 hours} \text{]} = [\text{tp in}](\text{[} \text{tp 2 hours} \text{]} = \lambda \Psi \lambda \Phi [\Psi(\lambda i [\Phi(i)])](\lambda Q [\exists i_1 (\text{dur}(i_1, 2\text{hour}) \land Q(i_1) \land (i_1 \subseteq I))] = \lambda \Phi [\exists i_1 (\text{dur}(i_1, 2\text{hour}) \land \Phi(i_1) \land (i_1 \subseteq I))] \]

\[(238) \text{[} \text{tp for 2 hours} \text{]} = [\text{tp for}](\text{[} \text{tp 2 hours} \text{]} = \lambda \Psi \lambda \Phi [\Psi(\lambda i [\Phi(i)])](\lambda Q [\exists i_1 (\text{dur}(i_1, 2\text{hour}) \land Q(i_1) \land (i_1 \subseteq I))] = \lambda \Phi [\exists i_1 (\text{dur}(i_1, 2\text{hour}) \land \Phi(i_1) \land (i_1 \subseteq I))] \]

Main clauses have been interpreted with an existentially quantified temporal interval thus far in this chapter, for example, the unmodified sentence (239) is assigned the semantic interpretation (240).

\[(239) \text{John returned} \]

\[(240) \text{[} s \text{ John returned} \text{]} = \lambda I [\exists i (\text{return}(\text{john})(i) \land (i < \text{now}) \land (i \subseteq I))] \]

Since durative in requires an existentially quantified temporal interval, we can apply rule (241) and simultaneously generate the semantic interpretation of a modified sentence as in (242) by applying the semantic interpretation of the temporal preposition given in (237) to the semantic interpretation of the sentence given in (240) and lambda-abstracting the free variable $I$. 

(241) \( S/\lambda I[\varphi(\psi)] \rightarrow S/\psi, \ TPP/\varphi \)

(242) \([s, \text{John returned in 2 hours}] = \lambda I[[\text{TPP in 2 hours}](s, \text{John returned})] = \lambda I[\lambda \Phi[\exists i_1(\text{dur}(i_1, 2 \text{hour}) \land \Phi(i_1) \land (i_1 \subseteq I))] = \lambda I[\lambda I_1[\exists i_1(\text{dur}(i_1, 2 \text{hour}) \land \exists i(\text{return}(john)(i) \land (i < \text{now}) \land (i \subseteq i_1)) \land (i_1 \subseteq I))]] \)

In order to generate the semantic interpretation for a for sentence as given in (232), we need another semantic interpretation for tense different from the one assigned to the Tense symbol in grammar rules (131) and repeated in (243). We therefore assign the semantic interpretation in (244) to the past tense, given that the quantification for the event interval of the main clause is introduced by the interpretation of tense.

(243) \([\text{tense past}] = \lambda T \lambda x_1[\exists i(T(i)(x_1) \land (i < \text{now}) \land (i \subseteq I))] \)

(244) \([\text{tense past}] = \lambda T \lambda x_1[\forall i((i \subset I) \rightarrow T(i)(x_1) \land (i < \text{now}))] \)

We generate a verb phrase by combining tense with the pre-tensed verbal category \( V' \) as required by rule (246). The corresponding semantic interpretation is therefore generated by applying the semantic interpretation of tense given in (244) to the interpretation of the untensed verbal category (245) as seen in (247).

(245) \([v, \text{spoke}] = \lambda v \lambda i[\text{spoke}(v)(i)] \)

(246) \( VP/\varphi(\psi) \rightarrow Tense/\varphi, \ V'/\psi \)

(247) \([v_p \text{spoke}] = \lambda T \lambda x_1[\forall i((i \subset I) \rightarrow T(i)(x_1) \land (i < \text{now}))][\lambda v \lambda i_1[\text{spoke}(v)(i_1)]] = \lambda x_1[\forall i((i \subset I) \rightarrow \text{spoke}(x_1)(i) \land (i < \text{now}))] \)

We now generate the semantic interpretation for an unmodified sentence such as (248) given the production rule (249).

(248) \( \text{John spoke.} \)

(249) \( S/\lambda I[\varphi(\psi)] \rightarrow NP/\varphi, \ VP/\psi \)
5.5. TEMPORAL PREPOSITIONS

(250) \([s \text{ John spoke}] = \lambda I[[\text{tp}] \text{ John }][[\text{tp} \text{ spoke}]]\) = \\
\lambda I[\lambda P[P(\text{john})](\lambda x_1[\forall i((i \subset I) \rightarrow \text{speak}(x_1)(i) \land (i < \text{now}))])] = \\
\lambda I[\forall i((i \subset I) \rightarrow \text{speak}(\text{john})(i) \land (i < \text{now}))]

Given rule (241) we can now generate the semantic interpretation of sentence (251) as seen in (252).

(251) John spoke for 2 hours

\([s \text{ John spoke for 2 hours}] = \lambda I[[\text{tp} \text{ for 2 hours}][[[s \text{ John spoke}]]]\) = \\
\lambda I[\lambda \Phi[\exists i_1(\text{dur}(i_1, 2\text{hour}) \land \Phi(i_1) \land (i_1 \subseteq I))] \\
(\lambda I[\forall i((i \subset I) \rightarrow \text{speak}(\text{john})(i) \land (i < \text{now}))])] = \\
\lambda I[\exists i_1(\text{dur}(i_1, 2\text{hour}) \land \forall i((i \subset i_1) \rightarrow \text{speak}(\text{john})(i) \land (i < \text{now})) \land (i_1 \subseteq I))]

5.5.4 Before and After

The most commonly used temporal modifiers for describing temporal intervals in a sequence are before and after. These modifiers are mostly used in temporal contexts, but there are instances where they can be used in spatial contexts. Compare the following sentences.

(253) The defendant was allowed to testify before the grand jury.

(254) The defendant was allowed to testify before the recess.

The difference between the type of contexts the prepositions in the above sentences provide is due to the type of noun phrase complement. While sentence (253) provides a spatial complement because of its object noun complement, sentence (254) with a temporal noun complement provides a temporal context.

After can also be used in the spatial context, although in a different manner from before. Consider the following sentence.

(255) Turn right after 200 yards.

The spatial instance of after is often used in imperative statements as seen in sentence (255). Before and after will however always provide temporal contexts when complemented by sentential structures. From our Brown corpus analysis, we find that 21% of the extracted sentences are instances of the spatial use of before, while the remaining 79% are temporal. We will study the behaviour of these 79% temporal instances of before. Due to the general imperative structure of the spatial instance of after, our extracted sentences from the Brown corpus only included temporal instances of after.
Syntactic Analysis

Although we refer here to *before* and *after* as temporal prepositions, they in fact permit both nominal and sentential complements as seen in sentences (256) and (257) below.

(256) John had cleaned the windows before Mary arrived.

(257) The guests arrived before noon.

(258) Luke had never met Joe before.

Sentence (258) unlike (256) and (257) does not have a complement, in this case, *before* has a deictic reference pointing to the time of speech.

Among 158 temporal instances of *before* extracted from the Brown corpus, we find that 54.4% of the sentences have sentences as complements, 16.5% are event nouns, 15.2% are temporal proper nouns and 13.9% have deictic temporal references. Table 5.5 describes the syntactic structure of the non-sentential complements of *before*, where def refers to temporal nouns with definite determiners, indef - indefinite determiners and Parasitic are those structures with parasitic gap followed by a progressive verb phrase as in sentence (259).

<table>
<thead>
<tr>
<th>Def</th>
<th>Indef</th>
<th>Deictic</th>
<th>None</th>
<th>count</th>
<th>Parasitic</th>
<th>Date</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>3</td>
<td>8</td>
<td>7</td>
<td>15</td>
<td>16</td>
<td>13</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.5: Data of the Syntactic Structure of *Before* Non-sentential Complements

(259) Mike left before hearing the announcement.

In sentence (259), the subject of the subordinate clause can be said to be the same as the subject of the main clause. We can say that there is a parasitic gap between the preposition and the subordinate clause subject.

*After* on the other hand permits similar syntactic structures in its subordinate phrase as *before*. Deictic references are however less common as *after* complements than *before* complements. From two hundred Brown corpus *after* sentences, we observe that 48% have sentential complements, 28% are temporal nouns, and 24% are temporal proper nouns. From the study of two hundred *after* sentences consider the distribution of the syntactic structure of *after* non-sentential complements in Table 5.6.
5.5. TEMPORAL PREPOSITIONS

<table>
<thead>
<tr>
<th>Def</th>
<th>Indef</th>
<th>Deictic</th>
<th>None</th>
<th>count</th>
<th>parasitic</th>
<th>Date</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>29</td>
<td>21</td>
<td>7</td>
<td>7</td>
<td>29</td>
<td>33</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 5.6: Data of the Syntactic Structure of After Non-sentential Complements

Semantic Interpretation

In this section we present the semantic interpretation of before and after in instances when they behave like temporal prepositions. Intuitively, the temporal modifiers before and after simply provide temporal order between two intervals. We, however, employ a similar interpretation to Pratt-Hartmann, Francez (2001) interpretation of before and after. According to Pratt-Hartmann and Francez, a given event occurs within a temporal context \( I \). Given sentence (260), the main event time and the time of reference introduced by the before phrase are contained in the temporal context denoted as \( I \).

(260) John arrived before the lecture.

*Before* therefore places the main clause event within some temporal interval from the arbitrary start of \( I \) to the start of the time of reference. Sentence (260) can therefore be interpreted as given a temporal context within which John arrived and the lecture occurred, John’s arrival is at some point within the beginning of the temporal context and the beginning of the lecture. The semantic interpretation of sentence (260) is therefore given as follows;

(261) \( \lambda I [\exists i (\text{lecture}(i) \land \forall j (\text{lecture}(j) \rightarrow (i = j)) \land \exists i_1 (\text{arrive(john)}(i_1) \land (i_1 < \text{now}) \land (i_1 \subseteq \text{init}(i, I)) \land (i \subseteq I))] \)

where \text{init} selects the sub-interval which begins at the start of \( I \) and ends at the start of \( i \). We, therefore, assign the following semantic interpretation to before.

(262) \( \llbracket_{tp} \text{before} \rrbracket = \lambda \Psi \lambda \Phi [\Psi (\lambda i [\Phi (\text{init}(i, I))])] \)

Given that we consider after as the complement of before, after is assigned the following interpretation.

(263) \( \llbracket_{tp} \text{after} \rrbracket = \lambda \Psi \lambda \Phi [\Psi (\lambda i [\Phi (\text{comp}(i, I))])] \)

where \text{comp} selects the sub-interval which begins at the end of \( i \) and ends at the end of \( I \).
From production rule (264), we generate the semantic interpretation of a *before* phrase in (266), by applying the semantic interpretation of *before* as given in (262) with the semantic interpretation of the subordinate clause given in (265).

\[(264)\quad \text{TPP/ϕ(ψ)} \rightarrow \text{TP/ϕ, NP/ψ}\]

\[(265)\quad [\text{tp the lecture}] = \lambda Q[\exists i_1(\text{lecture}(i_1) \land \forall j(\text{lecture}(j) \rightarrow (i_1 = j)) \land (i_1 \subseteq I) \land Q(i_1))]\]

\[(266)\quad [\text{tpp before the lecture}] = [\text{tp before}](\text{tp the lecture}) = \lambda \Psi \lambda \Phi[\Psi(\lambda i[\Phi(\text{init}(i, I))])]\]

Given the production rule (267), we can generate the semantic interpretation of sentence (260) as seen in (269), by applying the semantic interpretation of the temporal prepositional phrase as given in (266) to the semantic interpretation of the unmodified sentence as given in (268) and then lambda-abstract the free variable.

\[(267)\quad S/\lambda I[\varphi(\psi)] \rightarrow S/\psi, \text{TPP}\varphi\]

\[(268)\quad [\text{s John arrived}] = \lambda I[\exists i(\text{arrived}(\text{john})(i) \land (i < \text{now}) \land (i \subseteq I))]\]

\[(269)\quad [\text{s John arrived before the lecture}] = \lambda I[[\text{tpp before the lecture}](\text{tp the lecture})] = \lambda \Psi \lambda \Phi[\exists i_1(\text{lecture}(i_1) \land \forall j(\text{lecture}(j) \rightarrow (i_1 = j)) \land (i_1 \subseteq I) \land \Phi(\text{init}(i_1, I)))]\]

\[(270)\quad \text{John arrived after the lecture.}\]

\[(271)\quad [\text{s John arrived after the lecture}] = \lambda I[\exists i_1(\text{lecture}(i_1) \land \forall j(\text{lecture}(j) \rightarrow (i_1 = j)) \land \exists i(\text{arrive}(\text{john})(i) \land (i < \text{now}) \land (i \subseteq \text{init}(i_1, I)) \land (i_1 \subseteq I))]\]
5.5. TEMPORAL PREPOSITIONS

In this section we have shown the generation of a *before* sentence with a nominal complement. We discuss *before* and *after* as temporal conjunctions in section 5.6.1.

5.5.5 By

The preposition *by* has a similar semantic interpretation to *before*. The preposition *by* asserts that the main event has been completed as of the time of reference.

Syntactic Analysis

The complements of *by* phrases are usually temporal proper nouns as in sentence (272). But just as with locative prepositions previously discussed, *by* also allows complements that describe subintervals of noun phrases (pre-modified temporal nouns preceded by definite determiners) as in sentences (273) and (274). Regular event nouns produce unnatural constructions as in sentence (275). Brée et al. (1993) also observes this about the temporal preposition *by*.

(272) The President will meet the delegates by noon.

(273) The president will meet the delegates by the end of the conference.

(274) The opposing team will equalise by the second half

(275) ?The President will meet the delegates by the meeting.

*By* is quite selective with the main clause tense it permits. Although past-tensed main clause is not particularly ungrammatical, the preposition *by* will be more appropriate in the perfective as in sentence (277).

(276) ?The president met the delegates by noon.

(277) The president had met the delegates by noon.

The perfective is more appropriate because the preposition *by* tends to assert the completion of the main event as of a given time of reference, which is similar to Dowty (1979) description of the perfective.

The past progressive is ungrammatical in *by* sentences. When in the present tense it tends to provide a future reading. Consider the sentences below:
CHAPTER 5. TEMPORAL MODIFIERS

<table>
<thead>
<tr>
<th>Definite</th>
<th>Deictic</th>
<th>Year</th>
<th>Month</th>
<th>Dates</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>6</td>
<td>12</td>
<td>8</td>
<td>15</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 5.7: Data summary of the syntactic structure of *By* complements

(278) The train departs by midnight.

(279) ? The train has departed by midnight.

(280) The train is departing by midnight.

As observed in sentence (279), the present perfect is appears unnatural except when further modified to imply multiple episodes as in sentence (281).

(281) The train has always departed by midnight.

The simple present tense is not very natural and would rather be complemented by a locative preposition or be in the progressive as in sentence (280).

The future tense, however, asserts that the main event will be complete before the temporal reference introduced by the *by* phrase. Consider the following sentences.

(282) The train will depart by midnight.

(283) The train will be departing by midnight.

(284) The train will have departed by midnight.

**Semantic interpretation**

Having established the similarity between the semantic interpretation of *before* and *by*, we assign the semantic interpretation (285) to the temporal preposition *by*.

\[
(285) \llbracket_{tp} \text{by} \rrbracket = \lambda \Psi. \lambda \Phi. \Psi(\lambda i. [\Phi(\text{init}(i, I))])
\]

Given a temporal proper noun such as *midnight*, with its interpretation given in (287), we generate the semantic interpretation of the temporal preposition phrase by applying the semantic interpretation of temporal preposition to the semantic interpretation of the noun phrase as required by rule (286).
5.5. TEMPORAL PREPOSITIONS

\[ \text{(286) } TPP / \varphi(\psi) \rightarrow TP / \varphi, NP / \psi \]

\[ \text{(287) } [\text{np midnight}] = \lambda P[P(\text{midnight}) \land (\text{midnight} \subseteq I)] \]

\[ \text{(288) } [\text{tpp by midnight}] = \text{(tp by)[(np midnight)]} = \lambda \Psi \lambda \Phi[\Psi(\lambda i[\Phi(\text{init}(i, I))])(\lambda P[P(\text{midnight}) \land (\text{midnight} \subseteq I)]) = \lambda \Phi[\Phi(\text{init}(\text{midnight}, I) \land (\text{midnight} \subseteq I))] \]

Given an unmodified sentence such as (289) interpreted as in (290),

\[ \text{(289) } \text{John had arrived} \]

\[ \text{(290) } [\text{s} \text{ John had arrived}] = \lambda I[\exists i(\text{arrived(john)}(i) \land (i < \text{now}) \land (i \subseteq I))] \]

We generate the semantic interpretation of sentence (291) as seen in (293), by applying the semantic interpretation of the temporal prepositional phrase to the semantic interpretation of the unmodified sentence and then lambda-abstract the free variable as required by rule (292).

\[ \text{(291) } \text{John had arrived by midnight,} \]

\[ \text{(292) } S / \lambda I[\varphi(\psi)] \rightarrow S / \psi, TPP \varphi \]

\[ \text{(293) } [\text{s} \text{ John had arrived by midnight}] = \lambda I[[\text{tpp by midnight}][\text{s} \text{ John had arrived}]] = \lambda I[\lambda \Phi[\Phi(\text{init}(\text{midnight}, I) \land (\text{midnight} \subseteq I))] (\lambda I_1[\exists i_1(\text{arrived(john)}(i_1) \land (i_1 < \text{now}) \land (i_1 \subseteq I_1))))] = \lambda I[\exists i_1(\text{arrive(john)}(i_1) \land (i_1 < \text{now}) \land (i_1 \subseteq (\text{init}(\text{midnight}, I))) \land (\text{midnight} \subseteq I))] \]

5.5.6 Until

Until much like before and after can be complemented by either temporal noun phrases or sentences, until however has only a temporal use. We consider the behaviour of until and till in this section as they both have similar syntactic behaviours and semantic interpretations. Until is one of the more widely studied temporal modifiers as its semantic interpretation and syntactic representation have been studied by a number of linguists such as Karttunen (1973), Brée (1985), Brée, Smit (1986), Pratt-Hartmann, Francez (2001).
Syntactic Analysis

*Until* permits a wide range of syntactic structures as complement, most common are temporal proper nouns as in sentence (294), premodified temporal nouns as in sentence (295), and sentences as in sentence (296):

(294) John slept until 5:45pm.

(295) Mark continued composing until the last year of his life

(296) John slept until Mary arrived.

The main clause requires a durative verb; this is because like *for*, *until* enforces universal quantification of the sub-intervals of an arbitrary reference interval that is terminated at the time introduced by the *until* phrase. Therefore, given a culminating verb, we get an unnatural sentence as in sentence (297) below.

(297) ?Job drank a pint of beer until midnight.

Similarly, we observe that the perfect is not permitted in an *until* sentence main clause giving that the perfect coerces culmination. The progressive on the other hand when in interaction with an accomplishment strips off its culmination making it resemble an activity as in sentence (298).

(298) Job was drinking a pint of beer until midnight.

Table 5.8 provides a data summary of the non-sentential complements of *until*, where *def* are the noun phrases complements with definite determiners, *indef* are those with indefinite determiners and *embedded* refer to situation where *until* is followed by another temporal preposition, for example, *until before*, *until after*.

<table>
<thead>
<tr>
<th>Def</th>
<th>Indef</th>
<th>Deictic</th>
<th>None</th>
<th>Year</th>
<th>Month</th>
<th>Date</th>
<th>embedded</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0</td>
<td>13</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>4</td>
<td>19</td>
</tr>
</tbody>
</table>

Table 5.8: Data of the Syntactic Structure of *Until* Non-sentential Complements

Semantic interpretation

*Until* asserts that a particular event persists and is only terminated at the interval introduced by the *until* phrase. Given therefore an *until* sentence such as (299), we assign the semantic interpretation in (300).
Mary slept until noon

\[
\lambda I [\exists i \forall j((i \subseteq \text{init}(\text{noon}, I)) \land (j \subset i) \rightarrow 
\text{sleep(mary)}(j) \land (j < \text{now}) \land (\text{noon} \subseteq I))]
\]

*Until* tends to place the main clause event prior to the time introduced by its subordinate clause and phrase, we therefore assign the same semantic interpretation for *until* as we have for *before* as given below.

\[
\lambda \Psi \lambda \Phi [\Psi(\lambda i [\Phi(\text{init}(i, I))])]
\]

We generate the semantic interpretation of a temporal prepositional phrase as required by rule (302) as seen in (303) by applying the semantic interpretation of the temporal preposition to the semantic interpretation of the complement noun phrase.

\[
\text{TPP}/\varphi(\psi) \rightarrow \text{TP}/\varphi, \text{NP}/\psi
\]

\[
\lambda \Psi \lambda \Phi [\Psi(\lambda i [\Phi(\text{init}(i, I))])](\lambda P[P(\text{noon}) \land (\text{noon} \subseteq I)]) = 
\lambda \Phi[\Phi(\text{init}(\text{noon}, I)) \land (\text{noon} \subseteq I)]
\]

*Until*, however demands certain semantic requirements from its main clause, such as an arbitrary interval of reference within which the main event occurs. We therefore require an interpretation for tense different from (243) and (244). Consider the interpretation for tense with an existentially quantified time of reference within which the main event holds below.

\[
\lambda T \lambda x[\exists i_1 \forall j((i_1 \subseteq I) \land (j \subset i_1) \rightarrow T(j)(x) \land (j < \text{now}))]
\]

Similar to the generation of the semantic interpretation of simple tensed sentences in section 5.2, we can generate the semantic interpretation of the main clause of sentence (299) as seen in (305) if (304) is applied as the semantic interpretation of the tense.

\[
\lambda I [\exists i_1 \forall j((i_1 \subseteq I) \land (j \subset i_1) \rightarrow \text{sleep(mary)}(j) \land (i < \text{now}))]
\]
As required by rule (306), we generate the semantic interpretation of sentence (299) as seen in (307) by applying the semantic interpretation of the temporal prepositional phrase to the semantic interpretation of the main clause and lambda-abstracting the free variable $I$.

(306) $S/\lambda I[\varphi(\psi)] \rightarrow S/\psi$, $TPP\varphi$

(307) $\llbracket s\text{ Mary slept until noon} \rrbracket = \lambda I[\llbracket TPP\text{ until noon} \rrbracket(\llbracket s\text{ Mary slept} \rrbracket)] = \\
\lambda I[\lambda \Phi[\Phi(\text{init}(noon,I)) \land (noon \subseteq I)] \\
(\lambda I[\exists i_1 \forall j((i_1 \subseteq I_1) \land (j \subset i_1) \rightarrow \text{sleep}(mary)(j) \land (j < \text{now}))])] = \\
\lambda I[\exists i_1 \forall j((i_1 \subseteq (\text{init}(noon,I))) \land (j \subset i_1) \rightarrow \\
\text{sleep}(mary)(j) \land (j < \text{now}) \land (noon \subseteq I))]$

5.5.7 Since

There are two observable uses of since, namely – the temporal use as in sentence (308) and inferential use as in sentence (309).

(308) Yolanda has lived in London since the 1980s

(309) Since Yolanda lives in London Adam will have a place to sleep.

Brée, Smit (1986) observes the syntactic marker that distinguishes these two uses of since. The temporal since often has a perfective main clause, while the inferential since does not. Since provides information on the starting point of the interval the main clause occurred.

Syntactic Analysis

Although the temporal since often has a perfective verb in the main clause, the main clause verbs are often states or achievements as in sentences (310) and (311) respectively. We can have the activities and accomplishments in the perfect progressive, and passive voiced main clause as in sentence (313)

(310) Rose had arrived since 16:30.

(311) Dudley has been a boxer since 2012.

(312) John has been sleeping since Mary left.
The constitution had been written since the beginning of the 19th century. 

\textit{Since} like \textit{before, after} and \textit{until} can be complemented by either a temporal noun or a sentence. When the main clause is in present perfect, the subordinate clause is often in the past tense as in sentence (312). \textit{Since} is not used in referencing the future. When in the past tense the sentence describes the event time and the reference time as being prior to the time of speech as in sentence (310).

\textbf{Semantic Interpretation}

\textit{Since} appears to require stative verbs in its main clause. Although the perfect is often known to enforce a culmination on events, it appears in the case of \textit{since} to convert achievements to states, that is, directly referring to the result state of the culmination of the event described by the verb, therefore making the main clause verb durative. That is, the result state of the main event remains true from the temporal interval introduced by the \textit{since} phrase. \textit{Since} therefore describes the beginning of the temporal interval the main event or state holds.

Brée, Smit (1986) considers \textit{since} and \textit{until} as symmetrical. As a result given that \textit{until} is assigned the same semantic interpretation as \textit{before} as seen in (301), we assign \textit{since} the same semantic interpretation as \textit{after} as given in (314)

\begin{equation}
\llbracket \text{tp} \text{ since} \rrbracket = \lambda \Psi \lambda \Phi [\Phi(\lambda i[\Psi(\text{comp}(i, I))])],
\end{equation}

Similar to \textit{until}, we require an arbitrary time of reference which we universally quantify over its sub-intervals. We therefore require the tense interpretation used in the generating \textit{until} main clauses (304) repeated below().

\begin{equation}
\llbracket \text{tense past} \rrbracket = \lambda T \lambda x [\exists i_1 \forall j((i_1 \subseteq I) \land (j \subset i_1) \rightarrow T(j)(x) \land (j < \text{now})])
\end{equation}

Given therefore a \textit{since} sentence such as (316), we assign it the semantic interpretation in (317)

\begin{equation}
\llbracket \text{John had finished the essay since Tuesday} \rrbracket = \lambda I [\exists i_1 \forall j((j \subset i_1) \rightarrow \exists x(\text{essay}(x) \land \forall y(\text{essay} \rightarrow x = y) \land \text{finished(john, x)}(j) \land (j < \text{now})) \land (i_1 \subseteq \text{comp(tuesday, I)}) \land (\text{tuesday, I}))]
\end{equation}

Given the semantic interpretation (319) of the nominal complement in sentence (316), we generate the semantic interpretation the since phrase by applying the
semantic interpretation of since given in (314) to the semantic interpretation of the noun phrase given in (319) as shown in (320)

(318) $TPP/\varphi(\psi) \rightarrow TP/\varphi, NP/\psi$

(319) $[[\text{tp Tuesday}]] = \lambda P[P(\text{tuesday}) \land (\text{tuesday} \subseteq I)]$

(320) $[[\text{tpp since Tuesday}]] = [[[\text{tp since}]]([[\text{tp Tuesday}]])]$  
$= \lambda \Psi \lambda \Phi (\lambda i[\Phi(\text{comp}(i, I))])(\lambda P[P(\text{tuesday}) \land (\text{tuesday} \subseteq I)])$  
$= \lambda \Phi[\Phi(\text{comp}(\text{tuesday}, I)) \land (\text{tuesday} \subseteq I)]$

We can now generate the semantic interpretation of sentence (316). Given the production rule (321), we apply the semantic interpretation of the temporal prepositional phrase given in (320) to the semantic interpretation of the main clause given in (322).

(321) $S/\lambda I[\varphi(\psi)] \rightarrow S/\psi, TPP\varphi$

(322) $[[\text{s John had finished the essay}]] =$  
$\lambda I[\exists i_1 \forall j((j \subseteq i_1) \rightarrow \exists x(\text{essay}(x) \land \forall y(\text{essay}(y) \rightarrow y = x) \land \text{finish}(\text{john}, x)(j) \land (j < \text{now}) \land (i_1 \subseteq I)))]$

(323) $[[\text{s John had finished the essay since Tuesday}]] =$  
$[[\text{s John had finished the essay}]]([[\text{tpp since Tuesday}]]) =$  
$\lambda I[\exists i_1 \forall j((j \subseteq i_1) \rightarrow \exists x(\text{essay}(x) \land \forall y(\text{essay}(y) \rightarrow y = x) \land \text{finished}(\text{john}, x)(j) \land (j < \text{now})) \land (i_1 \subseteq \text{comp}(\text{tuesday}, I)) \land (\text{tuesday}, I)))]$

5.6 Temporal Conjunctions

Temporal conjunctions are those temporal modifiers that are complemented by sentential structures such that the interval within which the complement sentence event or state holds provides time of reference information required by the temporal conjunction and its main clause. In this section, we consider the syntactic behaviours and semantic interpretations for the temporal conjunctions before, after, since, until and while.
5.6. TEMPORAL CONJUNCTIONS

5.6.1 Before and After

In section 5.5.4 we considered the syntactic behaviour and semantic interpretation of *before* and *after* when they are complemented by temporal nouns, that is, when they function as temporal prepositions. In this section, we consider their behaviour as temporal conjunctions. We noted that about 54.4% of the considered *before* sentences and 48% of *after* sentences are complemented by sentences. Tables 5.9 and 5.10 summarises the distribution of the syntactic structure of the sentential complements of *before* and *after* respectively.

<table>
<thead>
<tr>
<th></th>
<th>Past</th>
<th>Present</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Activity</strong></td>
<td>7</td>
<td></td>
</tr>
<tr>
<td><strong>Achievement</strong></td>
<td>22</td>
<td></td>
</tr>
<tr>
<td><strong>Accomplishment</strong></td>
<td>14</td>
<td></td>
</tr>
<tr>
<td><strong>State</strong></td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.9: Data of the Syntactic Structure of *Before* Sentential complements

<table>
<thead>
<tr>
<th></th>
<th>Past</th>
<th>Past Perfect</th>
<th>Present</th>
<th>Present Perfect</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Activity</strong></td>
<td>2</td>
<td>0</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td><strong>Achievement</strong></td>
<td>15</td>
<td>8</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td><strong>Accomplishment</strong></td>
<td>19</td>
<td>0</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td><strong>State</strong></td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.10: Data of the Syntactic Structure of *After* Sentential Complements

Sentences (324), (325) and (326) show the subordinate clause of *before* sentences in the past, present and future tense respectively. We observe the requirement of tense agreement between the main and subordinate clauses. Given a main clause in the future tense, however, we see the subordinate clause remains in the present tense.
(324) Mike spoke before/after Mary arrived.

(325) Mike speaks before/after the meeting begins.

(326) Mike will speak before/after Mary arrives.

While before and after permit similar syntactic structures in their main and subordinate clauses, one obvious difference between them is after permits the perfective in its subordinate clause as opposed to before as seen in sentence (327).

(327) John arrived after Mary had left.

Also, Before and after intuitively appear to be symmetrical. (Brée, Smit, 1986) however notes that given a before sentence there is no requirement that the subordinate clause occurs; this is however not the case with after. Consider sentences (328) and (329).

(328) Dillinger left the house before we could arrest him

(329) After Dillinger left the house, we arrested him.

Sentence (328) denies that Dillinger was arrested while (329) asserts that he was. We, however, do not consider this in our language, we assume that both conjunctions assert the events or states in their main and subordinate clauses.

Semantic Interpretation

The semantic interpretation of before and after as temporal conjunctions is the same as when they behave as temporal prepositions (that is, (262) and (263) respectively). In section 5.2 we noted that temporal noun phrases and subordinate clauses are assigned the same semantic type that is, $(i, t), t)$. We can therefore apply the semantic interpretation of temporal conjunction to the semantic interpretation of its subordinate clause to generate the semantic interpretation of a temporal prepositional phrase. Given therefore the following sentence,

(330) John arrived before Mary left,

where the semantic interpretation of the subordinate clause is assigned (331) below.
5.6. TEMPORAL CONJUNCTIONS

(331) \[ \text{[Mary left]} = \lambda Q[\exists i (\text{leave(mary)}(i) \land (i \subseteq I) \land Q(i))] \]

Rule (332) enables generation of the semantic interpretation of a temporal propositional phrase, given the semantic interpretation of a temporal conjunction and its complement subordinate clause.

(332) \( \text{TPP}/\varphi(\psi) \rightarrow \text{TC}/\varphi, S/\psi \)

we therefore generate the semantic interpretation sentence (330) as given in (334) by applying rule (333).

(333) \( S/\lambda I[\varphi(\psi)] \rightarrow S/\psi, \text{TPP}\varphi \)

(334) \[ \text{[John arrived before Mary left]} = \lambda I[\exists i (\text{leave(mary)}(i) \land \exists i_1 (\text{arrive(john)}(i_1) \land (i_1 < \text{now}) \land (i_1 \subseteq \text{init}(i, I)) \land (i \subseteq I))] \]

5.6.2 Until

\text{Until} like \text{before} and \text{after} also permits sentential structures as complement such that tense agreement is required between the main clause and subordinate clause verbs. The subordinate clauses of \text{until} are often punctual verbs (i.e., achievements), for example, sentence (335); this is because the temporal intervals introduced by the \text{until} phrases provides a termination time for the main clause event.

(335) Mary spoke until John left.

Stative verbs are also permitted as \text{until} complements are seen in the following sentence

(336) The company will continue to purchase property until they own half the city.

Table [5.11] provide data summary of the syntactic structure of \text{until} sentences.

Semantic interpretation

As with \text{before} and \text{after}, \text{until} retains the same semantic interpretation (301) when complemented with a sentence or a temporal noun phrase. Given therefore a sentence such as (337),
CHAPTER 5. TEMPORAL MODIFIERS

<table>
<thead>
<tr>
<th></th>
<th>Past</th>
<th>Past Perfect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Achievement</td>
<td>11</td>
<td>3</td>
</tr>
<tr>
<td>Accomplishment</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>State</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Present</td>
<td>Present Perfect</td>
</tr>
<tr>
<td>Activity</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Achievement</td>
<td>27</td>
<td>2</td>
</tr>
<tr>
<td>Accomplishment</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>State</td>
<td>23</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.11: Data of the Syntactic Structure of Until Sentential Complements

(337) John slept until Mary left,

Where the semantic interpretation of the subordinate clause is given as follows,

(338) $\llbracket_{s} \text{Mary left} \rrbracket = \lambda Q[\exists i(\text{leave(mary)}(i) \land (i \subseteq I) \land Q(i))]$.

Rule (339) enables the generation of the semantic interpretation of a temporal prepositions phrase, given the semantic interpretation of the temporal conjunction – until and its complement

(339) $TPP/\varphi(\psi) \rightarrow TC/\varphi, S/\psi$

Given the semantic interpretation of the main clause in (341), we can generate the semantic interpretation of sentence (337) as seen in (341) as required by rule (340).

(340) $S/\lambda I[\varphi(\psi)] \rightarrow S/\psi, TPP/\varphi$

(341) $\llbracket_{s} \text{John slept} \rrbracket =$

$\lambda I[\exists i \forall j((j \subset i) \land (i \subseteq I) \rightarrow \text{sleep(john)}(j) \land (j < \text{now}))]$

(342) $\llbracket_{s} \text{John slept until Mary left} \rrbracket = \lambda I[\llbracket_{\text{tpp until Mary left}}(\llbracket_{s} \text{John slept}) \rrbracket] =$

$\lambda I[\lambda \Phi[\exists i(\text{leave(mary)}(i) \land \Phi(\text{init}(i, I)) \land (i \subseteq I))]]$

$(\lambda I[\exists i \forall j((j \subset i) \land (i \subseteq I) \rightarrow \text{sleep(john)}(j) \land (j < \text{now}))])] =$

$\lambda I[\exists i(\text{leave(mary)}(i) \land 
\exists i_1 \forall j((j \subset i) \land (i_1 \subseteq (\text{init}(i, I))) \rightarrow \text{sleep(john)}(j) \land (j < \text{now})) \land (i \subseteq I))]]$
5.6. TEMPORAL CONJUNCTIONS

5.6.3 While

*While* is strictly a temporal conjunction, that is, it can only permit sentential complements. *While* can be said to be similar to *during* in that the main clause state or event holds within the interval introduced by the subordinate clause.

**Syntactic Analysis**

The subordinate clause of *while* is expected to be durative since we expect it to contain the main clause event or state. Achievements are therefore not permitted as the subordinate clause verb. Activities are permitted as they inherently imply an extended interval in which they occur as in sentence (343).

(343) Vince played the guitar while Gabriel sang.

Accomplishments can be viewed as activities with an eventual culmination. When we have an accomplishment in the subordinate clause, the interval of interest is the activity without necessarily including the culmination. Consider the following sentence.

(344) Vince played the guitar while Gabriel drank a glass of wine.

We know the event *drink a glass of wine* is true only when the glass has been emptied by Gabriel, however as a *while* complement, we are more interested in the interval before the culmination.

Because the subordinate clause requires a durative verb, we often have the progressive but never the perfective. Stative verbs mostly seem unnatural in the subordinate clause as well.

The Main clause of *while* sentences does not have many restrictions. The only obvious ones are that the perfective is unnatural; this is because the perfect refers to the interval after the termination of the event. Therefore having the perfect in the main clause of *while* sentences means the interval after the main event is within the subordinate clause interval which is not the temporal relationship *while* intends to provide.

**Semantic Interpretation**

Just like *during*, *while* acts as some temporal container within which the main clause event or states holds. A more pragmatic way to look at *while* is that the
main event occurred simultaneously to the subordinate clause event. There is, however, the impression that the interval of the main clause event is completely contained in the subordinate clause event. We therefore interpret *while* as in (345).

\[(345) \\[\text{tp while}\] = \lambda \Psi \lambda \Phi[\Psi(\lambda i[\Phi(i)])] \]

Consider the following sentence.

(346) John arrived while Mary slept.

We can generate the semantic interpretation of the *while* phrase given the production rule (347), by applying the semantic interpretation of the temporal conjunction to the semantic interpretation of the subordinate clause

\[(347) \text{TPP}/\varphi(\psi) \rightarrow \text{TC}/\varphi, \text{S}/\psi, \]

\[(348) \\[\text{tp while Mary slept}\] = \\[\text{tp while}\](\\[\text{s Mary slept}\]) = \lambda \Psi \lambda \Phi[\Psi(\lambda i[\Phi(i)])](\lambda Q[\exists i_1(\text{sleep}(\text{mary})(i_1) \land Q(i_1) \land (i_1 \subseteq I)]) = \lambda \Phi[\exists i_1(\text{sleep}(\text{mary})(i_1) \land \Phi(i_1) \land (i_1 \subseteq I))] \]

We therefore generate the semantic interpretation of sentence (346) by applying the semantic interpretation of a sentence to the semantic interpretation of the temporal prepositional phrase.

\[(349) \text{S}/\lambda I[\varphi(\psi)] \rightarrow \text{S}/\psi, \text{TPP}/\varphi \]

\[(350) \\[\text{s John arrived while Mary slept}\] = \lambda I[\\[\text{tp while Mary slept}\](\\[\text{s John arrived}\])] = \lambda I[\exists i_1(\text{sleep}(\text{mary})(i_1) \land \Phi(i_1) \land (i_1 \subseteq I))] = \lambda I[\exists i_1(\text{sleep}(\text{mary})(i_1) \land \exists i(\text{arrive}(\text{john})(i) \land (i < \text{now}) \land (i \subseteq I)))] = \lambda I[\exists i_1(\text{sleep}(\text{mary})(i_1) \land \exists i(\text{arrive}(\text{john})(i) \land (i \subseteq i_1)) \land (i_1 \subseteq I))] \]

5.7 Discussion

In this chapter, we discussed our corpus analysis, the syntactic structure and semantic interpretation of tensed sentences as well as the semantic interpretation of subordinate clauses and phrases. We then discussed the syntactic behaviour and semantic interpretation of temporal adverbials, temporal prepositions, and
temporal conjunctions. We are therefore able to provide syntactic and semantic rules for the generation of temporal prepositional phrases and modified sentences based on analysis of our set of generated sentences and sentences from Brown Corpus.

We observe that the considered temporal prepositions and conjunctions are assigned one of three semantic interpretations as given below.

\[(351) \lambda \Psi \lambda \Phi [\Psi (\lambda i [\Phi (i)])] \]
\[(352) \lambda \Psi \lambda \Phi [\Phi (\lambda i [\Psi (\text{init}(i, I))])] \]
\[(353) \lambda \Psi \lambda \Phi [\Phi (\lambda i [\Psi (\text{comp}(i, I))])] \]

where \((351)\) provides containment relationship between temporal intervals, \((352)\) and \((353)\) provide temporal order such that \((352)\) and \((353)\) are symmetrical.

Although we can have the same semantic interpretation for two modifiers, we observe that each modifier imposes specific semantic and syntactic requirements on their main clause and subordinate clause/phrases. There is a large literature on the syntactic representation and semantic interpretation of temporal expressions, most however consider a class of temporal expressions in isolation. For example, Heny (1982) and Richards (1982) each attempt to provide a definitive semantic interpretation for grammatical aspects. Pratt-Hartmann, Francez (2001) presents a solid foundation on the semantic interpretation of a number of temporal modifiers, this analysis, however, provides little description on the effects on tense/aspects constructions on the semantic interpretation of temporal modifiers. Our analysis, however, is one which considers syntactic behaviour and semantic interpretation of various temporal modifiers taking into consideration a number of factors particularly the tense, aspect and aspectual class configuration permitted in the main and subordinate clause; the effect a given configuration can have on the semantic interpretation of a particular temporal modifier.

In section 4.4, we briefly introduced a controlled natural language with temporal features. In order to design such a language, we are required to understand the interactions of temporal expressions as we have done in this chapter. As defined in chapter 1, a controlled natural language is a computer processable subset of a natural language with explicitly regimented syntax and vocabulary such that member sentences are unambiguously translated to some formal logic. Given therefore the syntactic and semantic analysis of temporal expressions and
their interactions, we can define a controlled natural language where member sentences include temporal expressions – tenses, aspects and temporal modifiers, such that their semantic interpretations unambiguously express the interaction of contained temporal expressions in some formal language. The following sentences are valid sentences in our language.

(354) John wrote poems.

(355) John wrote poems everyday.

(356) John wrote poems everyday in January.

Sentences (354), (355) and (356) are examples of a simple tensed sentence, a sentence modified by a temporal adverbial and a sentence with embedded temporal modifiers respectively.

We consider the semantic and syntactic restrictions imposed by these temporal modifiers in the design of our CNL. In the next chapter, we present the syntactic structure of our language and how semantic interpretations are generated.
Chapter 6

Design of a CNL with Temporal Features

As discussed in chapter 2, a language comprises of a vocabulary and a set of rules that determines how words in the said vocabulary combine to form sentences. In the same chapter, we considered the formal interpretations of sentences and how these interpretations can be automatically generated from the interpretation of their syntactic constituents – Montague semantics.

In this chapter, we consider the development of a controlled natural language with English as its base language. This language is expected to be able to parse and provide semantic interpretations for sentences with temporal expressions. Just as with every other language, our language possesses a vocabulary (or lexicon) and a set of grammar rules. The grammar rules governing the generation of the various syntactic structures in our language are defined in definite clause grammar. Corresponding semantic interpretations are generated with the aid of Montague semantics. The language has been developed in Prolog, as a result, the syntactic rules provided in this chapter resemble Prolog syntax. A subset of the code for our language is provided in the appendix. We have omitted parts of the code that computes the lambda calculus – $\alpha$ and $\beta$ reductions and pretty printing of our semantic interpretations.

We describe our language’s lexicon in section 6.1. Section 6.2 presents the general form of the non-terminal symbols in our language as well as the production rules for generating them. Section 6.3 presents the semantic interpretation of the various syntactic categories contained in our language, showing how they are generated from the semantic interpretation of their constituents.
6.1 Lexicon

A lexicon refers to the set of terminal symbols of a given language such that each member is of a part of speech (POS). Every member sentence of a language is strictly formed from the combination of member lexical items of the language’s lexicon.

Lexical items in our language have the following general form:

\[ \text{lex}(\Phi, \Psi), \]

where \( \Phi \) is its part of speech and \( \Psi \) is a list of feature-value pairs. Given therefore the participle inflection of a verb, for example, written, we define it in our lexicon as follows.

\[ \text{lex}(\text{verb}, [\text{nlpr:write}, \text{syn:[written]}, \text{tense:participle}, \text{num:sgl}, \text{class:acc}, \text{type:tran}]) \]

Following the general form in (357), lexical item definition (358) provides part of speech information, in this case verb. The following list of features contains information such as the non-logical primitive (nlpr) of the lexical item as applicable, the value of the syntax feature is the inflection of interest, the value of the tense feature ranges over the three simple tenses in English – present, past, future and the two grammatical aspects – participle and progressive, the number (num) feature which is assigned either singular (sgl) or plural (pl) values, the class feature ranges over the four Vendler aspectual classes of verbs and finally, the type feature which is assigned either transitive (tran) or intransitive (intran).

In our language, we distinguish six different parts of speech, namely – determiners, nouns, verbs, auxiliary verbs, temporal prepositions/conjunctions and tense. Just as we have shown with verbs, each of these parts of speech has varied feature-value pairs used in their definition as lexical items. We categorise these parts of speech into two classes – function words and content words. Function words can be considered words that express grammatical and semantic relationships. Determiners, auxiliary verbs, temporal prepositions/conjunctions and tense are function words. Content words, on the other hand, describe entities, events, states, and intervals. Verbs and nouns are therefore content words.

The difference between content and function words in our lexicon is that content words possess the nlpr feature while function words do not. Content
words are represented in our semantic interpretation with the value of their \textit{nlpr} feature which is the bare form of the said inflection of interest as seen in (358). Since function words express grammatical and semantic relationships, they do not require the \textit{nlpr} feature. Given therefore a determiner such as every, we assign it lexical entry (359) in our lexicon, as opposed to the definition of content words as seen in (358).

\begin{equation}
\text{lex(det,\{syn:\textit{[every]}, num:sgl, type:uni, class:nontemp\})}
\end{equation}

Section \ref{sec:Appendix} in the appendix provides the prolog code showing how lexical items are defined in our language. Table \ref{tab:6.1} summarises the features possessed by each part of speech.

### 6.2 Grammar Rules

The grammar of a given language contains sets of rules that determine the structure of its member sentences. Grammar rules for our language are given in definite clause grammar.

In this section, we present the general form of each of the syntactic categories of interest. We also show the production rules for generating these categories. Note that although these production rules resemble Prolog code, they have however been modified for the sake of readability. That is, specific variables such as the value of the semantic feature, and variables we do not show how they unify

<table>
<thead>
<tr>
<th>POS</th>
<th>Syntax \textit{(syn)}</th>
<th>Symbol \textit{(nlpr)}</th>
<th>Number \textit{(num)}</th>
<th>Type \textit{(type)}</th>
<th>Class \textit{(class)}</th>
<th>Tense \textit{(tense)}</th>
<th>Polarity \textit{(pol)}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determiners</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(det)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nouns</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(noun)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Verbs</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(verb)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Auxiliary verbs</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(aux)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prepositions</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(tp)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tense</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>(tense)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.1: POS and their Specific Features
are given in letters of the Greek alphabet. Section 6.3 provides the semantic interpretation of the categories discussed in this section, a subset of the Prolog code is given in the appendix.

### 6.2.1 Determiners

Determiners are words that occur at the beginning of noun phrases, that is, they precede noun heads. They provide information about how specific or general a common noun is. In our language, we distinguish four types of determiners namely, nominal negation, definite, indefinite and universal determiners. Each of these types of determiners can either be singular or plural which enables enforcement of number agreement with the noun head they precede. A determiner can also be classed as temporal or non-temporal. In this section, we provide the syntactic rule for generating the non-terminal symbol \texttt{det} (determiners) from an arbitrary determiner lexical item.

Given for example the singular non-temporal definite determiner \textit{the}, given in our lexicon as follows,

\[(360) \text{lex} (\text{det}, [\text{syn:}\{\text{the}\}, \text{num:sgl}, \text{class:nonTemp}, \text{type:def}])\],

a suitable non-terminal symbol \texttt{det} (determiners) will be one that retains the class, number and type information about the corresponding lexical item it is generated from. We, therefore, have the following general form for determiners in our grammar

\[(361) \text{det}([\text{sem:}\Phi, \text{num:X}, \text{class:Y}, \text{type:Z}]),\]

such that \(\Phi\) is the assigned semantic interpretation of the determiner, \(\theta\) can either be assigned values \texttt{sgl} (singular) or \texttt{pl} (plural), \(\delta\) can either be \texttt{temp} (temporal) or \texttt{nonTemp} (non-temporal), and \(\gamma\) ranges over the four types of determiners (\texttt{neg}, \texttt{indef}, \texttt{def} and \texttt{uni}).

Given therefore a lexical item whose POS is a determiner, we have the following DCG production rule with a \texttt{det} head.

\[(362) \text{det}([\text{sem:}\Phi, \text{num:X}, \text{class:Y}, \text{type:Z}]) \rightarrow \text{lex} (\text{det}, [\text{syn:}\{\psi\}, \text{num:X}, \text{class:Y}, \text{type:Z}])\]

The above rule lets us generate the non-terminal symbol \texttt{det} with \(X\) number, of class \(Y\) and type \(Z\) which are assigned the exact same values as corresponding
features of the lexical item in the body of the production rule. The value of the \textit{sem} feature is determined by the value of the type feature ($Z$) and class feature ($Y$) (that is, the type of the determiner and whether or not it is a temporal or non-temporal determiner), and $\psi$ is the particular inflection of interest. We present the semantic interpretation of the various types of determiners in section 6.3.1. Table 6.2 shows the features of the syntactic category \textit{det} and their range of values.

### 6.2.2 Nouns and Noun phrases

We generate the non-terminal symbol $n$ (nouns) from the lexical items whose part of speech is noun. Table 6.1 shows that lexical items of the noun POS are defined with the \textit{syntax} (\textit{syn}), \textit{non-logical primitive} (\textit{nlpr}), \textit{number} (\textit{num}) and \textit{type} (\textit{type}) features. The number feature (\textit{num}) can be assigned singular (\textit{sgl}) or plural (\textit{pl}) values and type (\textit{type}) ranges over proper (\textit{proper}), count (\textit{count}), day (\textit{day}), month (\textit{month}), time (\textit{time}), interval (\textit{interval}) and year (\textit{year}). In this section, we present the syntactic rules for generating the syntactic category noun from lexical items whose part of speech is noun, and noun phrases from the combination of \textit{det} (determiners) and \textit{n} (nouns) syntactic categories or directly from proper nouns.

Given the singular common countable noun such as \textit{boy}, defined in our lexicon thus,

(363) \texttt{lex(noun, [nlpr:boy, syn:[boy], num:sgl, type:count])},

its corresponding syntactic category \textit{n} requires a set of features that can retain the number (\textit{num}) and type (\textit{type}) values of the lexical item it is generated from.
As a result, we have the following general form for the syntactic category \( n \) in our grammar.

\[
(364) \ n([\text{sem} : \Phi, \ \text{num} : \theta, \ \text{type} : \delta]),
\]

where \( \Phi \) is the assigned semantic interpretation of the noun, \( \theta \) can be assigned the values \( \text{sgl} \) or \( \text{pl} \), \( \delta \) can be assigned any of the values \( \text{count}, \ \text{event}, \ \text{proper}, \ \text{day}, \ \text{month}, \ \text{time}, \ \text{interval} \) and \( \text{year} \). Production rule (365) shows the generation of the syntactic category \( n \) from a given lexical item of the noun POS.

\[
(365) \ n([\text{sem} : \Phi, \ \text{num} : X, \ \text{type} : Y]) \rightarrow \n
\text{lex(noun, [nlpr:} \delta, \ \text{syn:[} \gamma, \ \text{num:} X, \ \text{type:} Y])}
\]

The above rule lets us generate the non-terminal symbol \( n \) given a lexical item of inflection \( \gamma \), such that the values assigned to the features \( \text{num} \) and \( \text{type} \) are the exact same as those of the lexical item’s in the body of the rule and \( \delta \) is the bare form of the inflection of interest, hence information about nouns can be passed up the syntactic tree as required. Table 6.3 shows the features of the syntactic category \( n \) and their range of values.

Noun phrases (np) in our language can be generated in two ways. First, determiners combine with common nouns to form noun phrases. In this case, we are required to enforce number agreement between the constituent determiner (det) and noun (n). Also, temporal nouns (i.e. nouns of the type \text{interval}, \text{event}) are required to be preceded only by temporal determiners, while non-temporal nouns (nouns of type \text{count}) are preceded by non-temporal determiners.
6.2. GRAMMAR RULES

Second, proper nouns and temporal proper nouns (nouns of type time, month, dates, year) simply generate noun phrases. We have the following general form for the syntactic category np in our grammar.

\[(366) \text{np(}[\text{sem:} \Phi, \text{num:} \theta, \text{type:} \delta]\),\]

where \(\Phi\) is the assigned semantic interpretation of the noun phrase, \(\theta\) can be assigned the values sgl (singular) or pl (plural) and \(\delta\) ranges over the types of nouns given in Table 6.3. Production rule \((367)\) shows the generation of noun phrases from the combination of determiners (det) and nouns n. Production rule \((368)\) shows the generation of noun phrases from proper nouns.

\[(367) \text{np(}[\text{sem:} \Phi, \text{num:} X, \text{type:} Y]\)\rightarrow\]
\[\text{det(}[\text{sem:} \Psi, \text{num:} X, \text{class:} \delta, \text{type:} \gamma]\),\]
\[\text{n(}[\text{sem:} \Gamma, \text{num:} X, \text{type:} Y]\)\]

\[(368) \text{np(}[\text{sem:} \Phi, \text{num:} X, \text{type:} Y]\)\rightarrow\]
\[\text{n(}[\text{sem:} \Phi, \text{num:} X, \text{type:} Y]\)\]

We apply production rule \((367)\) given a noun of the type count (e.g., boy), interval (e.g., hour) or event (e.g., meeting) such that nouns of the type count are preceded by non-temporal determiners, while those of type interval and event are preceded by temporal determiners. In production rule \((368)\) \(Y\) can be assigned proper, year, month, date, or time values.

6.2.3 Tense

There are three simple tenses in English – past, present and future. Syntactically these tenses are mostly introduced by morphemes. In our lexicon we do not have lexical item definitions for tenses. As a result the production rule generating the syntactic category tense has an empty string in its body. We have the following general form for the syntactic category tense in our grammar,

\[(369) \text{tense(}[\text{sem:} \Phi, \text{type:} \delta]\)\]

where \(\delta\) can be assigned either past or future values. The following production rule shows the generation of tense for the past tense.

\[(370) \text{tense(}[\text{sem:} \Phi, \text{type:} \text{past}]\rightarrow[]\)\]

We show in the next section how tense is being applied to tensed verbs. The semantic interpretations of tenses are given in section 6.3.3.
6.2.4 Verbs and Verb Phrases

In this section we consider the syntactic rules for generating the categories \(v\) (verbs), \(aux\) (auxiliary verbs) and \(vp\) (verb phrases).

We generate the syntactic category \(v\) from the lexical item whose part of speech is verb. As seen in Table 6.1 lexical items of the verb POS are defined with the syntax (\(syn\)), non-logical primitive (\(nlpr\)), number (\(num\)), class (\(class\)), type (\(type\)), and tense (\(tense\)) features, where the \(num\) feature can be assigned singular (\(sgl\)) or plural (\(pl\)) values; the \(class\) feature ranges over Vendler’s four aspectual classes of verbs, namely—state (\(state\)), accomplishment (\(acc\)), achievement (\(ach\)), activity (\(act\)); \(type\) can be assigned either transitive (\(tran\)) or intransitive (\(intran\)) values; the \(tense\) feature ranges over the values \(past\), \(present\), \(progressive\), \(participle\) and \(untensed\).

Given therefore the past-tensed intransitive verb \(wrote\), defined in our lexicon as follows,

\[
(371) \text{lex(verb, } [\text{nlpr:write, syn:[wrote], num:sgl, tense:past, class:act, type:intran}])
\]

its corresponding \(v\) category is required to retain the \(num\), \(tense\), \(class\) and \(type\) features. We therefore, have the following general form for the syntactic category \(v\) in our grammar.

\[
(372) v(\{\text{sem:}\Psi, \text{num:}\delta, \text{tense:}\gamma, \text{class:}\rho, \text{type:}\theta\}),
\]

where \(\delta\) is assigned \(sgl\) or \(pl\) values, \(\gamma\) is assigned \(past\), \(present\), \(future\), \(progressive\), \(participle\) or \(untensed\), \(\rho\) ranges of the four aspectual classes of verbs (\(act\), \(acc\), \(ach\), \(state\)) and \(\theta\) is assigned \(tran\) or \(intran\). The following production rule shows the generation of the syntactic category \(v\) from an arbitrary lexical item of the verb part of speech.

\[
(373) v(\{\text{sem:}\Psi, \text{num:}\dot{W}, \text{tense:}\dot{X}, \text{class:}\dot{Y}, \text{type:}\dot{Z}\}) \rightarrow \\
\text{lex(verb, } [\text{nlpr:}\Phi, \text{syn:[}\psi\text{], num:}\dot{W}, \text{tense:}\dot{X}, \text{class:}\dot{Y}, \text{type:}\dot{Z}])
\]

The production rule in (373) shows the generation of the non-terminal symbol \(v\) given a lexical item such that the values assigned to the features \(num\), \(tense\), \(class\), \(type\) are the exact same as those of the lexical item’s in the body of the rule, and \(\psi\) is the inflection of interest. Consider below the generation of the syntactic category \(v\) given the past-tensed, singular, intransitive verb \(wrote\)
6.2. GRAMMAR RULES

Table 6.4 shows the features of the category v and their range of values.

<table>
<thead>
<tr>
<th>POS</th>
<th>Feature</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>verb</td>
<td>num</td>
<td>sgl, pl</td>
</tr>
<tr>
<td></td>
<td>type</td>
<td>intran, tran</td>
</tr>
<tr>
<td></td>
<td>class</td>
<td>state (state), act (activity), acc (accomplishment), ach (achievement)</td>
</tr>
<tr>
<td></td>
<td>tense</td>
<td>past, present, future, progressive, participle</td>
</tr>
</tbody>
</table>

Table 6.4: Verb Features and Range of Values

The syntactic category aux is generated in our language from lexical items whose part of speech is auxiliary verb. The syntactic category aux is defined in our language with the syntax (syn), type (type), polarity (pol) and tense (tense) features, such that the type feature can be assigned have, be, or do values and the tense feature ranges over the three simple tenses in English – past, present and future, and pol is assigned positive (pos) or negative (neg). Given for example the auxiliary verb had with the following representation in our lexicon,

(375) \[ \text{lex}(\text{aux}, [\text{syn}:[\text{had}], \text{num}:\text{sgl}, \text{tense}:\text{past}, \text{pol}:\text{pos}, \text{type}:\text{have}]) \]

its corresponding aux category is given as follows,

(376) \[ \text{aux}([\text{num}:\text{sgl}, \text{tense}:\text{past}, \text{pol}:\text{pos}, \text{type}:\text{have}]). \]

We therefore have the following general form for auxiliary verbs in our grammar,

(377) \[ \text{aux}([\text{num}:\delta, \text{tense}:\gamma, \text{pol}:\theta, \text{type}:\psi]) \]
where $\delta$ is assigned sgl or pl, $\gamma$ ranges over the three simple tenses in English, $\theta$ can be assigned pos (positive) or neg (negative) and $\psi$ can be assigned either have, be or do.

(378) $\text{aux([num:}W, \text{tense:}X, \text{pol:}Y, \text{type:}Z]) \rightarrow$

$\text{lex(aux,[syn:[}\psi], \text{num:}W, \text{tense:}X, \text{pol:}Y, \text{type:}Z])$

The above production rules show that the features of the aux category in the head of the rule are assigned the same values as those of the lexical item’s in the body. Table 6.5 provides the features of the aux syntactic category and their range of values.

We have several rules for generating verb phrases (vp) in our language. Before we present these rules, consider the general form of the category vp in our grammar,

(379) $\text{vp([sem:}\Phi, \text{num:}\delta, \text{class:}\gamma, \text{tense:}\theta])},$

where the features $\text{num}$, and $\text{class}$ range over corresponding features for the syntactic category v as summarised in Table 6.4. The $\text{tense}$ feature however range over the simple English tenses and complex tense constructions such as past progressive, future participle, etc. Generally verb phrases are generated directly from intransitive verbs or from the combination of verbs and noun phrases. In our language, we also consider the generation of verb phrases given tensed verb heads and a preceding auxiliary verbs. As a result we have several production rules for generating verb phrases in our language. First, given an untensed intransitive verb, we define a production rule that directly generates a verb phrase as seen in the following production rule.
6.2. GRAMMAR RULES

(380) \( \text{vp}([\text{sem}:\Phi, \text{num}:X, \text{class}:Y, \text{tense:untensed}]) \rightarrow \text{v}([\text{sem}:\Phi, \text{num}:X, \text{class}:Y, \text{tense:untensed, type:intran}]) \),

Similarly, given an untensed transitive verb, a verb phrase is generated from its combination with a noun phrase thus,

(381) \( \text{vp}([\text{sem}:\Phi, \text{num}:X, \text{class}:Y, \text{tense:untensed}]) \rightarrow \text{v}([\text{sem}:\Gamma, \text{num}:X, \text{class}:Y, \text{tense:untensed, type:tran}], \text{np}([\text{sem}:\Psi, \text{num:}, \text{type}\delta])) \).

Production rule (381) allows the generation of verb phrases from a verb and a noun phrase provided the value of the tense feature is untensed. Rule (381) also requires the noun phrase type feature to be either count (countable nouns) or proper (proper nouns).

We introduce two intermediate syntactic categories required in the generation of more complex verb phrases – \( \text{vb}, \text{vbar} \). The category \( \text{vb} \) is generated from verbs where the tense feature is assigned the value progressive or participle. Given therefore the syntactic category \( \text{v} \) of the intran type (that is, an intransitive perfective or progressive verb e.g written, writing), its corresponding \( \text{vb} \) category is generated directly as required by rule (382). Suppose the syntactic category \( \text{v} \) is assigned the tran type value (that is, a transitive perfective or progressive verb), then a corresponding \( \text{vb} \) requires the combination of the verb with a noun phrase as shown in rule (383).

(382) \( \text{vb}([\text{sem}:\Phi, \text{num}:X, \text{class}:Y, \text{tense:Z}]) \rightarrow \text{v}([\text{sem}:\Gamma, \text{num}:X, \text{class}:Y, \text{tense:Z, type:intran}]) \),

(383) \( \text{vb}([\text{sem}:\Phi, \text{num}:X, \text{class}:Y, \text{tense:Z}]) \rightarrow \text{v}([\text{sem}:\Gamma, \text{num}:X, \text{class}:Y, \text{tense:Z, type:tran}], \text{np}([\text{sem}:\Psi, \text{num:}, \text{type}\delta])) \),

where \( Z \) is assigned either the progressive or participle values and \( \delta \) is assigned either proper or count values.

The syntactic category \( \text{vbar} \) is generated in various ways. The category \( \text{vbar} \) forms a verb phrase \( \text{vp} \) when preceded by the tense category. The simplest rule for generating \( \text{vbar} \) is directly from an intransitive verb (\( \text{v} \)) with its tense feature assigned either the past or future as seen in rule (384).
where \( Y \) is assigned past or future.

Given a transitive verb, we generate the \( \text{vbar} \) category by combining the \( \text{v} \) category of interest with an \( \text{np} \) of type proper or count.

\[
\text{vbar ([sem: } \Phi, \text{ num:X, class:Z, tense:Y]}) \rightarrow \text{v ([sem: } \Phi, \text{ num:X, tense:Y, class:Z type:intran]}) \]

\[
\text{vbar ([sem: } \Phi, \text{ num:X, class:Z, tense:Y]}) \rightarrow \text{v ([sem: } \Psi, \text{ num:X, tense:Y, class:Z type:tran]}) , \text{np ([sem: } \Gamma, \text{ num:_, type:} \delta]) \]

where \( \delta \) is assigned either proper or count and \( Y \) is assigned past or future.

Finally, \( \text{vbar} \) is generated from the combination of the \( \text{aux} \) category and the \( \text{vb} \) category such that an \( \text{aux} \) of type have combines with a \( \text{vb} \) of participle tense value. Similarly, an \( \text{aux} \) of type be combines with a \( \text{vb} \) of progressive tense value. The resulting \( \text{vbar} \) category tense feature is assigned the combination of the \( \text{tense} \) value of the \( \text{aux} \) category and the \( \text{tense} \) value of the \( \text{vb} \) category. We are therefore able to generate complex tense-aspect structures such as past participle, future progressive, etc. Rules (386) and (387) illustrate the generation of \( \text{vbar} \) from the combination of an \( \text{aux} \) category and an \( \text{vb} \) category.

\[
\text{vbar ([sem: } \Phi, \text{ num:X, class:R, tense:T]}) \rightarrow \text{aux ([num:X, tense:Y, pol:pos, type:have]) , \text{vb ([sem: } \Phi, \text{ num:X, class:R, tense:participle]})} \\
\text{vbar ([sem: } \Phi, \text{ num:X, class:R, tense:T]}) \rightarrow \text{aux ([num:X, tense:Y, pol:pos, type:be]) , \text{vb ([sem: } \Phi, \text{ num:X, class:R, tense:progressive]})} \\
\]

We now provide production rules for generating verb phrases from the combination of \( \text{tense} \) and \( \text{vbar} \) as required by rule (388) and (389).

\[
\text{vp ([sem: } \Phi, \text{ num:X, class:Y, tense:Z]}) \rightarrow \text{tense ([sem: } \Psi, \text{ tense:Z]) , \text{vbar ([sem: } \Gamma, \text{ num:X, class:Y, tense:Z]})} \\
\text{vp ([sem: } \Phi, \text{ num:X, class:Y, tense:} \phi]) \rightarrow \text{tense ([sem: } \Psi, \text{ tense:} \phi]) , \text{vbar ([sem: } \Gamma, \text{ num:X, class:Y, tense:} \delta])} \\
\]
Since category aux tense feature can only be assigned the simple English tense – past, present, future, rule (388) generates a vp category provided the values of the tense feature of both constituent categories are equal hence applying the correct tense to the verb with the appropriate morpheme. Rule (389) combines tense and vbar categories given a vbar with a complex tense configuration such as pastprogressive. In this case, the tense value of the tense category is expected to be equal to the simple tense prefix of the tense value of the vbar category. A vbar category with a pastprogressive tense value is therefore preceded by a tense category with a past tense value.

In chapter 5 we showed that temporal conjunctions are complemented by sentences. In our language, the verb phrases of these complements are assigned a different semantic interpretation from those of the main clause. As a result, we define a syntactic category specifically for the verb phrases of subordinate clauses of temporal conjunctions. Consider the general form of this verb phrase.

\[
\text{(390) compVerb([sem:\Phi, num:X, class:Y, tense:Z])} \rightarrow \\
\text{senTemp([sem:\Gamma])}, \\
\text{vbar([sem:\Psi, num:X, class:Y, tense:Z])},
\]

where the category provides particular semantic interpretations of the appropriate semantic type for complements of temporal conjunctions. We discuss this in the section on the semantics of our language.

### 6.2.5 Temporal Modifiers

By temporal modifiers, we refer to temporal prepositions, conjunctions, adverbs and temporal prepositional phrases. As discussed in chapter 5 temporal adverbs are simply generated from temporal noun phrases. We therefore have the following general form for temporal adverbials in our grammar.

\[
\text{(391) tempAdv([sem:\Phi, num:\delta])}
\]

Given an np cateory of the type interval, we generate the tempAdv category as follows,

\[
\text{(392) tempAdv([sem:Sem, num:X])} \rightarrow \\
\text{np([sem:Sem, num:X, type:interval])}
\]
The non-terminal symbol \( tp \) (temporal prepositions/conjunctions) is generated from lexical items whose part of speech is temporal preposition. From Table 6.1 we know that the temporal preposition part of speech requires syntax and type features, where the type feature ranges over the individual temporal prepositions and conjunctions discussed in the previous chapter. Given therefore the temporal conjunction \( \text{before} \), we define it thus in our lexicon,

\[
(393) \text{lex}(tp, [\text{syn:}[\text{before}], \text{type:before}]),
\]

its corresponding \( tp \) syntactic category is required to retain the type value. We therefore have the following general definition for the \( tp \) category.

\[
(394) \text{tp}([\text{sem:}\Phi, \text{type:}\delta]),
\]

where \( \delta \) ranges over the temporal prepositions and conjunctions considered in our language. The following production rule shows the generation of the syntactic category \( tp \) from a given lexical item of the \( tp \) part of speech.

\[
(395) \text{tp}([\text{sem:}\Phi, \text{type:}\gamma]) \rightarrow \text{lex}(tp, [\text{syn:}[\gamma], \text{type:}\gamma])
\]

Temporal prepositional phrases (\( tpp \)) are either generated when temporal prepositions combine with temporal nouns or temporal conjunctions combine with sentences. The following production rules show the generation of temporal prepositional phrases from the combination of temporal prepositions and temporal noun phrases.

\[
(396) \text{tpp}([\text{sem:}\Phi, \text{type:R}]) \rightarrow \text{tp}([\text{sem:}\Psi, \text{type:R}]), \text{np}([\text{sem:}\Gamma, \text{num:}\delta, \text{type:}\theta])
\]

Production rules for generating the \( tpp \) category are peculiar to the value of the type feature of the category. From our theory on the various behaviours of temporal modifiers in chapter 5 we define several rules for the ten temporal prepositions/conjunctions used in our language.

The locative prepositions \( in, on \) and \( at \) as discussed in the previous chapter are complemented by different types of temporal nouns. \( In \) is complemented by years and month, \( on \) is complemented by days and \( at \) is complemented by time. Production rules (397), (398) and (399) define the generation of temporal prepositional phrases from the combination of temporal preposition \( in, on \) and \( at \) respectively.
Before considering the generation of temporal prepositional phrases from temporal conjunctions, we present first the generation of the syntactic category, `compSen`, which defines sentential complements of temporal conjunctions. At the end of section 6.2.4, we introduced the category `compVerb` which is a verbal category specifically for generating subordinate clause verb phrases. The category `compSen` is therefore generated through the combination of the `np` category and the `compVerb` category as seen in the production rule below.

\[
\text{compSen}([\text{sem:}\Phi, \text{num:}X, \text{class:}Y, \text{tense:}Z]) \rightarrow \\
\text{np}([\text{sem:}\Psi, \text{num:}X, \text{type:}\delta]), \\
\text{compVerb}([\text{sem:}\Gamma, \text{num:}X, \text{class:}Y, \text{tense:}Z]),
\]

where \(\delta\) is assigned `count` or `proper` values.

Rule (401) defines the generation of a temporal preposition from the combination of a temporal conjunction and a complement sentence.

\[
\text{tpp}([\text{sem:}\Phi, \text{tense:}Z, \text{type:}R]) \rightarrow \\
\text{tp}([\text{sem:}\Psi, \text{type:}R]), \\
\text{compSen}([\text{sem:}\Gamma, \text{num:}\delta, \text{class:}\psi, \text{tense:}Z]),
\]

The temporal conjunctions `before`, `after`, `until` and `since` however permit nominal and sentential complements. Given therefore a `tp` of type `before`, `after`, `until`, `since`, we can apply either rules (396) or (401) in order to generate a temporal prepositional phrase. There are however restrictions of the type of temporal nouns and sentences that combine with these temporal conjunctions. For example the temporal modifier `before` is complemented by interval nouns, event nouns, all the temporal proper nouns and simple tensed sentences, but reject perfective complements. `After` on the other hand permits all forms of complements `before` permits including the perfect.
6.2.6 Sentences

In our language, we distinguish two types of sentences – saturated and unsaturated sentences. Saturated sentences are those that are not further combined with temporal modifiers, while unsaturated sentences are expected to combine with a temporal modifier. We, therefore, have the following general form for sentences in our language.

\[(402) \text{sen}([\text{sem}: \Phi, \text{num}: \delta, \text{tense}: \gamma, \text{class}: \phi, \text{type}: \psi])\]

where \(\delta\) can be assigned \textit{sgl} (singular) or \textit{pl} (plural) values, \(\phi\) ranges over the four aspectual class of verbs, \(\gamma\) ranges over the simple and complex tenses in English and \(\psi\) is either \textit{sat} (saturated) or \textit{unsat} (unsaturated).

As required by production rule \((403)\), the simplest forms of sentences in our language are generated from the combination of noun phrases and untensed verb phrases.

\[(403) \text{sen}([\text{sem}: \Phi, \text{num}: X, \text{tense}: \text{untensed}, \text{class}: R, \text{type}: \text{sat}]) \rightarrow \text{np}([\text{sem}: \Psi, \text{num}: X, \text{type}: \delta]),\]

\[\text{vp}([\text{sem}: \Gamma, \text{num}: X, \text{class}: R, \text{tense}: \text{untensed}])\]

where \(\delta\) is assigned \textit{count} or \textit{delta}. In our language, untensed sentences do not combine with temporal modifiers and hence have their type feature assigned the saturated value.

Given a tensed verb phrase, we generate sentences as required by the following rule.

\[(404) \text{sen}([\text{sem}: \Phi, \text{num}: X, \text{tense}: Y, \text{class}: R, \text{type}: \gamma]) \rightarrow \text{np}([\text{sem}: \Psi, \text{num}: X, \text{type}: \delta]),\]

\[\text{vp}([\text{sem}: \Gamma, \text{num}: X, \text{class}: R, \text{tense}: Y])\]

where \(Y\) ranges over simple and complex English tenses, \(\gamma\) can be assigned either \textit{sat} (saturated) or \textit{unsat} (unsaturated) values and \(X\) is assigned the same value as its constituents, hence enforcing number agreement. Given the sentences \((405)\), its corresponding production rule is given in \((406)\).

\[(405) \text{The boy wrote a book}\]

\[(406) \text{sen}([\text{sem}: \Phi, \text{num}: \text{sgl}, \text{tense}: \text{past}, \text{type}: \text{sat}]) \rightarrow \text{np}([\text{sem}: \Psi, \text{num}: \text{sgl}, \text{type}: \text{count}]),\]

\[\text{vp}([\text{sem}: \Gamma, \text{num}: \text{sgl}, \text{tense}: \text{past}])\]
6.3. SEMANTICS

We can also generate sentences from the combination of unsaturated sentences and temporal prepositional phrases. Consider the following production rules.

(407) \( \text{sen}([\text{sem}: \Phi, \text{num}: X, \text{tense}: Y, \text{type}: \gamma]) \rightarrow \)
\( \text{sen}([\text{sem}: \Psi, \text{num}: X, \text{tense}: Y, \text{type}: \text{unsat}]), \)
\( \text{tempAdv}([\text{sem}: \Gamma, \text{num}: \_]) \)

(408) \( \text{sen}([\text{sem}: \Phi, \text{num}: X, \text{tense}: Y, \text{type}: \gamma]) \rightarrow \)
\( \text{sen}([\text{sem}: \Psi, \text{num}: X, \text{tense}: Y, \text{type}: \text{unsat}]), \)
\( \text{tpp}([\text{sem}: \Gamma, \text{type}: \psi]) \)

(409) \( \text{sen}([\text{sem}: \Phi, \text{num}: X, \text{tense}: Y, \text{type}: \gamma]) \rightarrow \)
\( \text{sen}([\text{sem}: \Psi, \text{num}: X, \text{tense}: Y, \text{type}: \text{unsat}]), \)
\( \text{tpp}([\text{sem}: \Gamma, \text{tense}: Y, \text{type}: \psi]) \),

where production rule (407) generates a sentence modified by a temporal adver-bial, production rule (408) generates a sentence modified by a temporal prepositional phrase and (409) generates a sentence modified by a temporal conjunction such that tense agreement is required between the main and subordinate clauses.

6.3 Semantics

Section 6.2 presents the syntactic representations of the various structures in our language. Given that our grammar is a semantically annotated definite clause grammar, this section provides the corresponding semantic interpretation of the syntactic structures considered, that is, the value of their sem feature.

In order to show the generation of the semantic interpretation of the structures in our language, we introduce the following conventions.

1. Given a production rule with a single symbol in its body (terminal or non-terminal), the semantic interpretation of the head symbol of the rule is equal to the semantic interpretation of the symbol in the body of the rule.

2. Given a production rule with two non-terminal symbols in its body, the semantic interpretation of the head of the rule is generated by applying the semantic interpretation of the first symbol, to the semantic interpretation of the second symbol in the body of the rule.

We indicate cases when there are deviations from these conventions.
6.3.1 Determiners

The `det` category in our language can be divided into two sub-categories – temporal determiners and non-temporal determiners. Non-temporal determiners in our language are assigned the semantic type `((e,t),((e,t),t))` – that is, they are functions from functions from objects to Boolean to functions from objects to Boolean to Boolean. Temporal determiners on the other hand are assigned that semantic type `((i,t),(i,t),t))` – that is, functions from functions from intervals to Boolean to functions from functions from intervals to Boolean to Boolean. In our language, we define four types of determiners – nominal negation, definite, indefinite and universal determiners. The semantic interpretations assigned to these types of determiner are respectively given thus,

\[(410)\] \(\lambda P\lambda Q[\exists x(\neg P(x) \land Q(x))]\]

\[(411)\] \(\lambda P\lambda Q[\exists x(P(x) \land \forall y(P(y) \rightarrow (x = y)) \land Q(x))]\]

\[(412)\] \(\lambda P\lambda Q[\exists x(P(x) \land Q(x))]\]

\[(413)\] \(\lambda P\lambda Q[\forall x(P(x) \rightarrow Q(x))]\).

Temporal determiners are assigned similar interpretations to non-temporal determiners. The major difference however is that variables \(P\) and \(Q\) in interpretations \[(410)\)-\[(413)\] range over lambda terms of semantic types \((e,t)\) as opposed to the interpretations of temporal determiners in \[(414)\]-\[(417)\] which range over lambda terms of semantic type \((i,t)\). However, in order to distinguish the semantic interpretations of temporal determiners from non-temporal determiners, we have intervals represented by specific variables – \(i,j\) while objects can be represented by other letters of the alphabet, typically \(x, y\) and \(z\).

\[(414)\] \(\lambda P\lambda Q[\exists i(\neg P(i) \land (i \subseteq I) \land Q(i))]\]

\[(415)\] \(\lambda P\lambda Q[\exists i(P(i) \land (i \subseteq I) \land \forall j(P(j) \rightarrow (i = j)) \land Q(i))]\]

\[(416)\] \(\lambda P\lambda Q[\exists i(P(i) \land (i \subseteq I) \land Q(i))]\]

\[(417)\] \(\lambda P\lambda Q[\forall i(P(i) \land (i \subseteq I) \rightarrow Q(i))]\].
6.3. SEMANTICS

6.3.2 Noun and Noun phrases

In Section 6.2.2 we presented several types of nouns, namely, proper, countable, event, interval, time, day, month, year. In this section we provide interpretations for these noun types.

We noted that the event, interval, time, day, month and year types are temporal nouns while proper and countable nouns are non-temporal. First let us consider non-temporal nouns – proper and countable nouns. The semantic interpretation of proper nouns in our language is assigned the semantic type \((e,t)\) – that is, they are functions from functions from objects to Boolean to Boolean. Given therefore the lexical item *John*, the corresponding n category is assigned the following semantic interpretation.

\[
\llbracket n \text{ john} \rrbracket = \lambda P[P(\text{john})]
\]

The semantic interpretations of countable nouns are assigned the semantic type \((e,t)\) – that is, they are functions from objects to Boolean. Given therefore the lexical item *boy*, the corresponding n category is assigned the following semantic interpretation.

\[
\llbracket n \text{ boy} \rrbracket = \lambda y[\text{boy}(y)]
\]

Temporal nouns have similar interpretations to non-temporal nouns. Event and interval nouns behave like countable nouns and hence have similar interpretations. For example, the event noun *meeting* has its corresponding n assigned the semantic interpretation (420). Nouns of the type time, day, month and year (these are generally referred to as temporal proper nouns) behave like proper nouns with the inclusion of a temporal context. Given for example the temporal proper noun *January* of the type month, its corresponding semantic interpretation is given in (421).

\[
\llbracket n \text{ meeting} \rrbracket = \lambda i[\text{meeting}(i)]
\]

\[
\llbracket n \text{ january} \rrbracket = \lambda P[P(\text{january} \land (\text{january} \subseteq I))]
\]

The semantic interpretation of event and interval nouns (e.g., semantic interpretation (420)) are assigned the lambda type \((i,t)\) – that is, they are functions from intervals to Boolean. The semantic interpretation of temporal proper nouns on the other hand are assigned the semantic type \(((i,t),t)\) – that is, they are functions from functions from intervals to Boolean to Boolean.
The np (noun phrases) category is generated in two ways – proper nouns directly generate noun phrases as required by rule (368) and determiners combine with nouns to form noun phrases as required by rule (367). The semantic interpretation of the np category when generated from a proper noun is assigned the same semantic interpretation as the proper noun in the body of the rule as seen in (422) and (423).

(422) \[ \text{np john} = [n \text{ John}] = \lambda P[P(\text{john})] \]

(423) \[ \text{np january} = [n \text{ January}] = \lambda P[P(\text{january} \land (\text{january} \subseteq I))] \]

Given for example, the noun phrase the boy, such that its constituent determiner is assigned the semantic interpretation in (411), and the constituent noun is assigned the semantic interpretation in (419), the semantic interpretation noun phrase is generated by applying the semantic interpretation of the determiner to the semantic interpretation of the noun as seen in (424). Similarly, noun phrases generated from the combination of temporal determiners and event or interval nouns are assigned semantic interpretations such as (425).

(424) \[ \text{np the boy} = [\text{det the}](\text{np boy}) = \\
\lambda P \lambda Q[\exists x (P(x) \land \forall y (P(y) \rightarrow (x = y)) \land Q(x))] (\lambda x_1[\text{boy}(x_1)]) = \\
\lambda Q[\exists x (\text{boy}(x) \land \forall y (\text{boy}(y) \rightarrow (x = y)) \land Q(x))] \]

(425) \[ \text{np every meeting} = [\text{det every}](\text{np meeting}) = \\
\lambda P \lambda Q[\forall i (P(i) \land (i \subseteq I) \rightarrow Q(i))] (\lambda i_1[\text{meeting}(i_1)]) = \\
\lambda Q[\forall i (\text{meeting}(i) \land (i \subseteq I) \rightarrow Q(i))] \]

6.3.3 Tense

We provide semantic interpretations for two tenses in our language – past and future tenses, which are both assigned the semantic type \(((i, (e, t)), (e, t))) – that is, they are functions from functions from intervals to functions from objects to Boolean to functions from objects to Boolean. In chapter 5 we defined three interpretations for tenses, these interpretation are given in (426)-(428).

(426) \[ \text{tense past} = \lambda H \lambda x[\exists i (H(i)(x) \land (i < \text{now}) \land (i \subseteq I))] \]

(427) \[ \text{tense past} = \lambda T \lambda x[\exists i_1 \forall j ((i_1 \subseteq I) \land (j \subset i_1) \rightarrow T(j)(x) \land (i < \text{now})]) \]
(428) \[ [\text{tense past}] = \lambda T \lambda x_{1} \left[ \forall i ((i \subset I) \rightarrow T(i)(x_{1}) \land (i < \text{now})) \right] \]

where the operator \(<\) provides temporal order and the constant \(\text{now}\) represents the time of utterance. The above interpretations therefore locate the event time before the time of utterance. The future tense is assigned similar interpretations, the temporal order reversed (i.e., the speech time is located before the event time). We show in section 6.3.4 the interaction of tense with different verbal categories.

### 6.3.4 Verb and Verb Phrases

The semantic interpretations of verbs vary depending on whether they are transitive or intransitive as well as whether they are tense or untensed. The semantic interpretations of untensed intransitive verbs are assigned the semantic type \((e, t)\) – that is, they are functions from objects to Boolean. For example, given an untensed intransitive verb such as \(\text{write}\), the semantic feature of its corresponding \(v\) category is assigned the following semantic interpretation.

(429) \[ [v \text{ write}] = \lambda u [\text{write}(u)] \]

The semantic interpretations of untensed transitive verbs are assigned the semantic type \((((e, t), t), (e, t))\) – that is, they are functions from functions from objects to Boolean to functions from object to Boolean. Given a transitive untensed verb such as \(\text{write}\), the semantic feature of its corresponding \(v\) is assigned the following semantic interpretation.

(430) \[ [v \text{ write}] = \lambda u \lambda v [u(\lambda w [\text{write}(v, w)])] \]

Tensed verbs require a temporal variable to account for the event time. Tensed intransitive verbs are assigned the semantic type \((i, (e, t))\) – that is, they are functions from intervals to functions from objects to Boolean. Given therefore a past tensed intransitive verb such as \(\text{wrote}\), the semantic feature of its corresponding \(v\) category is assigned the following interpretation.

(431) \[ [v \text{ wrote}] = \lambda i \lambda w [\text{write}(w)(i)] \]

The semantic interpretations of tensed transitive verbs are assigned the semantic type \(((e, t), t), (i, (e, t))\) – that is, they are functions from functions from objects to Boolean to functions from intervals to functions from objects to Boolean. Given therefore a past tensed transitive verb such
as *wrote*, the semantic feature of its corresponding \( v \) category is assigned the following interpretation.

\[(432) \] 
\[ \lambda u \lambda i \lambda v \left[ u \left( \lambda w \lbrack \text{write}(v, w)(i) \rbrack \right) \right] \]

Just as with verbs, the semantic interpretations of verb phrases are varied depending on whether or not the verb head is tensed or untensed. Untensed verb phrases are assigned the semantic type \((e, t)\). There are two rules for generating untensed verb phrases in our language. First, an untensed verb phrase is assigned the same semantic interpretation as its constituent intransitive verb as seen in (433). Untensed transitive verbs, however, combine with noun phrases to generate verb phrases as given in production rule (381). The semantic interpretation of the verb phrase is generated by applying the interpretation of the constituent verb phrase to the semantic interpretation of the noun phrase as seen in (434).

\[(433) \] 
\[ \lambda u \left[ \text{write}(u) \right] \]

\[(434) \] 
\[ \lambda u \lambda i \lambda v \left[ u \left( \lambda w \lbrack \text{write}(v, w)(i) \rbrack \right) \right] \left( \lambda Q \left[ \exists x \left( \text{book}(x) \land Q(x) \right) \right] \right) = \lambda i \lambda v \left[ \exists x \left( \text{book}(x) \land \text{write}(v, x) \right) \right] \]

The semantic interpretations of tensed verb phrases are assigned the semantic type \((i, (e, t))\). From syntactic rules (384) and (385), we know that the category \( v \bar{b} \) is generated directly from tensed intransitive verbs or from the combination of tensed transitive verbs and noun phrases. The semantic interpretation of the \( v \bar{b} \) is assigned the same semantic interpretation as the semantic interpretation of its constituent intransitive verb \( v \) as seen in (435), or by applying the semantic interpretation of the constituent \( v \) category to the semantic interpretation of the constituent \( \text{np} \) category as seen in (436).

\[(435) \] 
\[ \lambda i \lambda w \left[ \text{write}(w)(i) \right] \]

\[(436) \] 
\[ \lambda i \lambda w \left[ u \left( \lambda w \lbrack \text{write}(v, w)(i) \rbrack \right) \right] \left( \lambda Q \left[ \exists x \left( \text{book}(x) \land Q(x) \right) \right] \right) = \lambda i \lambda w \left[ \exists x \left( \text{book}(x) \land \text{write}(v, x)(i) \right) \right] \]

Finally, let us consider the generation of the semantic interpretation of verb phrases from the combination of tense and the \( v \bar{b} \) category. After combination with tense, verb phrases are assigned the semantic type \((e, t)\). From production
6.3. SEMANTICS

We know that a verb phrase is generated from the combination tense and the \( \text{vbar} \) category. We therefore generated the semantic interpretation of the generated \( \text{vp} \) by applying the semantic interpretation of tense to the semantic interpretation of the \( \text{vbar} \) as seen in (437).

\[
\begin{align*}
(437) \quad & [\text{vp} \text{ wrote a book}] = [\text{tense past}]( [\text{vb} \text{ wrote a book}]) = \\
& \lambda T \lambda x_2[\exists i_1(T(i_1)(x_2) \land (i_1 < \text{now}) \land (i_1 \subseteq I))] \\
& (\lambda i \lambda v[\exists x(\text{book}(x) \land \text{write}(v, x)(i))]) = \\
& \lambda x_2[\exists i_1 \exists x(\text{book}(x) \land (x_2, x)(i_1) \land (i_1 < \text{now}) \land (i_1 \subseteq I))]
\end{align*}
\]

Syntactically, verb phrases sometimes include the auxiliary verbs. We, however, do not need to consider auxiliary verbs given that they are not of any semantic significance but rather to retain grammatical correctness.

### 6.3.5 Temporal Modifiers

In this section, we present the semantic interpretation of temporal adverbials, temporal prepositions/conjunctions and temporal prepositional phrases. Production rule (392) generates temporal adverbs directly from temporal noun phrases, the semantic feature of the temporal adverb is assigned the same value as the semantic interpretation of its constituent temporal noun phrase. For example given the temporal noun \textit{every day}, we have the following semantic interpretation assigned to the temporal adverb it generates.

\[
(438) \quad [\text{tempAdv every day}] = [\text{tp every day}] = \lambda Q[\forall i(\text{day}(i) \land (i \subseteq I) \rightarrow Q(i))]
\]

Temporal prepositions/ conjunctions are assigned the semantic type \(((i, t), t), ((i, t), t))\) – that is, they are functions from functions from functions from intervals to Boolean to Boolean, to functions from functions from intervals to Boolean to Boolean. Given the temporal preposition/conjunction \textit{before}, we have the following semantic interpretation assigned to its semantic feature.

\[
(439) \quad [\text{tp before}] = \lambda \Psi \lambda \Phi[\lambda i[\Phi(\text{init}(i, I))])]
\]

Temporal prepositional phrases as required by production rules (396) and (401) are generated from the combination of temporal prepositions and temporal noun phrases or the combination of temporal conjunctions and sentences respectively. The semantic interpretation of a given temporal prepositional phrase is
therefore generated by applying the semantic interpretation of its constituent temporal preposition to the semantic interpretation of its constituent noun phrase as seen in (440) or complement sentence as seen in (441).

\[(440) \text{[tpp before every meeting]} = \text{[tp before][np every meeting]} = \lambda \Psi \lambda \Phi [\Psi (\lambda i [\Phi (\text{init}(i, I))])(\lambda Q [\forall i_1 (\text{meeting}(i_1) \land (i_1 \subseteq I) \rightarrow Q(i_1))])] = \lambda \Phi [\forall i_1 (\text{meeting}(i_1) \land (i_1 \subseteq I) \rightarrow \Phi (\text{init}(i_1, I)))]\]

\[(441) \text{[tpp before John arrived]} = \lambda \Psi \lambda \Phi [\Psi (\lambda i [\Phi (\text{init}(i, I))])(\lambda Q [\exists i_1 (\text{arrive}(\text{john})(i_1) \land (i_1 \subseteq I) \land Q(i_1))])] = \lambda \Phi [\exists i_1 (\text{arrive}(\text{john})(i_1) \land (i_1 \subseteq I) \land \Phi (\text{init}(i_1, I)))]\]

The semantic interpretations of temporal prepositional phrases are assigned the semantic type \((i, t, t)\) – that is, they are functions from functions from intervals to Boolean.

6.3.6 Sentences

In section 6.2.6 we presented the rules for generating sentences in our language. In this section we provide the semantic interpretation for the \textit{sen} syntactic category.

The \textit{sen} category in our language has two types – unsaturated and saturated. We noted that the simplest form of sentences in our language are untensed sentences. As observed from production rule (403) they are generated from the combination of noun phrases and untensed verb phrases. The semantic interpretation of an untensed sentence is therefore generated by applying the semantic interpretation of the constituent noun phrase to the semantic interpretation of the constituent verb phrase and lambda abstracting the free variable. Given for example the noun phrase \textit{the boy} with the interpretation given in (442), and an untensed verb phrase \textit{writes a book} as interpreted in (443), we generate the semantic interpretation of a sentence as seen in (444).

\[(442) \text{[np the boy]} = \lambda Q [\exists x (\text{boy}(x) \land \forall y (\text{boy}(y) \rightarrow (x = y)) \land Q(x))]\]

\[(443) \text{[vp writes a book]} = \lambda v [\exists z (\text{book}(z) \land \text{write}(v, z))]\]

\[(444) \text{[sen the boy writes a book]} = \lambda I [\text{[np the boy]}(\text{[vp writes a book]})] = \lambda Q [\exists x (\text{boy}(x) \land \forall y (\text{boy}(y) \rightarrow (x = y)) \land Q(x))]\]

\[(\lambda v [\exists z (\text{book}(z) \land \text{write}(v, z))]) = \exists x (\text{boy}(x) \land \forall y (\text{boy}(y) \rightarrow (x = y)) \land \exists z (\text{book}(z) \land \text{write}(x, z)))\]
The production rule (404) shows the generation of a tensed sentence from the combination of a noun phrase and a tensed verb phrase. Given therefore the noun phrase the boy with the semantic interpretation given in (442) and a past-tensed verb phrase wrote a book with semantic interpretation given in (445), we generate the semantic interpretation of a tensed sentence by applying the semantic interpretation of the noun phrase to the semantic interpretation of the tense verb phrase and lambda-abstract the free variable as seen in (446).

\[(445) \text{vp wrote a book} = \lambda v [\exists i \exists z (\text{book}(z) \land (v,z)(i) \land (i < \text{now}) \land (i \subseteq I))]\]

\[(446) \text{sen the boy wrote a book} = \lambda I [\lambda \Phi [\exists i_1 (\text{arrive(john)}(i_1) \land (i_1 \subseteq I) \land \Phi(\text{init}(i_1, I)))]
\quad (\lambda I_1 [\exists x (\text{boy}(x) \land \forall y (\text{boy}(y) \rightarrow (x = y)) \land
\quad \exists i \exists z (\text{book}(z) \land \text{write}(x,z)(i) \land (i < \text{now}) \land (i \subseteq I_1))]]]
\quad \lambda I [\exists i \exists j (\text{book}(z) \land \text{write}(x,z)(i) \land (i < \text{now}) \land (i \subseteq \text{init}(i_1 \subseteq I)))]\]

We generate the semantic interpretation of a modified sentence as seen in (447) by applying the semantic interpretation of the temporal prepositional phrase given in (440) to the semantic interpretation of the unmodified sentence given in (446) and lambda-abstract the free variable I.

\[(447) \text{sen the boy wrote a book before every meeting} = \lambda I [\lambda \Phi [\exists i_1 (\text{arrive(john)}(i_1) \land (i_1 \subseteq I) \land \Phi(\text{init}(i_1, I)))]
\quad (\lambda I_1 [\exists x (\text{boy}(x) \land \forall y (\text{boy}(y) \rightarrow (x = y)) \land
\quad \exists i \exists z (\text{book}(z) \land \text{write}(x,z)(i) \land (i < \text{now}) \land (i \subseteq I_1))]]]
\quad \lambda I [\exists i \exists j (\text{book}(z) \land \text{write}(x,z)(i) \land (i < \text{now}) \land (i \subseteq \text{init}(i_1 \subseteq I)))]\]

6.4 Conclusion

In this chapter, we provided a description of our language specifically the lexicon, grammar rules and associated semantic interpretations. We defined our grammar using definite clause grammar, which enables us to implement syntactic and semantic restrictions with the aid of feature-value pairs assigned to each syntactic category.

In chapter 4 we provided a survey of computer processable controlled natural languages. We observe the subsets defined by most of the considered languages
do not include temporal expressions. Our language attempts to provide meaning to temporal expressions as intuitively as possible, taking into consideration the effects temporal expressions have on the semantic interpretation of a given sentence within which they co-occur.
Chapter 7

Conclusions and Future Work

We have thus far shown the syntactic behaviours of temporal expressions, their semantic interpretations and inclusion within member sentences of a controlled natural language. In section 7.1 we conclude our findings by providing responses to the research questions given in chapter 1. In section 7.2 we consider possible future work on controlled natural languages with temporal features.

7.1 Conclusions

We now consider the three research questions raised in the introduction and provide answers based on results presented in this thesis.

1. How do we determine what combinations of temporal expressions are grammatical?
   
   We primarily considered three temporal expressions in this thesis, namely, tense, aspect, and temporal modifiers. Intuitively as English speakers, we know that there are specific configurations of these temporal expressions that produce grammatical sentences. In order to determine what combinations of these temporal expressions are grammatical, we generated a set of sentences for each of the temporal modifiers considered in chapter 5 with various tense, aspect and aspectual class configurations. Furthermore, we extract and analyse the syntactic structure of sentences containing the temporal modifiers of interest from a corpus. We are, therefore, able to determine the configuration of tense, aspect and temporal modifiers from the analysis of our manually generated set of sentences as well as from sentences extracted from the Brown corpus as discussed in section 5.1. It is
worth mentioning that other grammatical configurations of tense, aspect and temporal modifiers can be encountered if our analysis is carried out on a larger corpus.

2. **What effect does the co-occurrence of temporal expressions have on the semantic interpretation of a given natural language sentence?**

Montague semantics introduced in chapter 2 provides a mechanism for generating the semantic interpretation of a natural language sentence from the semantic interpretation of its constituent categories. The semantic interpretations of temporal expressions however often vary depending on the presence of other temporal expressions within a given sentence, for example, the semantic interpretation of the perfective is often dependent on the temporal modifier complementing the said sentence. As a result, along with the syntactic analysis of temporal modifiers presented in chapter 5, we provide the semantic interpretations of the considered temporal modifiers. We also consider the effect the various grammatical tense/aspect constructions of verbs in the main and subordinate clauses can have on the semantic interpretations assigned to the considered temporal modifiers.

3. **How can the syntactic behaviour and semantic interpretations of the combination of the various temporal expressions be defined in a controlled natural language?**

In chapter 2 we defined a language as a set of sentences over an alphabet or vocabulary. We also defined grammar as a set of rules responsible for the production of sentences in a given language. Defining a controlled natural language, therefore, requires the definition of a set of grammar rules responsible for producing member sentences of the said CNL. From our survey of different grammar formalisms, we find that definite clause grammar is most suitable for defining the syntax of a controlled natural language with temporal features given that it permits sub-categorization and it is easy to implement in Prolog. Section 6.1 and 6.2 describes our language’s lexicon and grammar respectively. The grammar particularly takes into consideration linguistic rules such as tense and number agreement. The different grammatical configurations of tense, aspects and temporal modifiers as discussed in chapter 5 are also applied in defining the grammar rules for sentences in our language. The semantic interpretations of member sentences
in our language are generated with the aid of Montague semantics. We are able to assign appropriate semantic interpretations to the various syntactic categories contained in our grammar with the aid of the sub-categorization facilitated by DCG arguments. As a result, sentences containing the temporal expressions – tense, aspect and temporal modifiers are translatable into first-order interval logic sentences.

Finally, we return to the hypothesis upon which the research questions are based.

- **A controlled natural language grammar with temporal features can sufficiently represent temporal information and relationships expressed by temporal expressions – tense, aspect, aspectual class and temporal modifiers.**

Given our ability to determine the syntactic behaviours of temporal expressions as required by the first research question; assign semantic interpretations for the different configurations of temporal expressions as required by the second research question; and define grammar rules and corresponding semantic rules based on our analyses as required by the third research question, our language in comparison to the controlled natural languages discussed in chapter 4 contains member sentences with temporal expressions such that the semantic interpretation of said member sentences unambiguously express the interactions between the contained temporal expressions. We, therefore, claim that a controlled natural language grammar with temporal features sufficiently represents temporal information and relationships as expressed by temporal expressions.

### 7.2 Future Work

Having proposed a controlled natural language with temporal features, we consider in this section the future of this research area.

An obvious future work for work for this research will be improving the expressiveness of our language. We can easily imagine a controlled natural language with temporal features that permits relative pronouns, handles anaphora, and queries a discourse of sentences, etc.

Expanding the focus of our language from just parsing sentences with temporal expressions will allow for application in many interesting domains. For
example, application to question answering systems is an interesting area of future research as we can extend our language to answer questions of temporal significance. Given that CNLs are popularly applied to software and hardware specification authoring, we can provide temporal relations between events and states contained in such documents.
Appendix A

Syntactic Analysis of Temporal Modifiers

<table>
<thead>
<tr>
<th></th>
<th>Perf</th>
<th>Prog</th>
<th>PerfProg</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Past</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity</td>
<td>7</td>
<td>3</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>Achievement</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Accomplishment</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>State</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Present</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Achievement</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Accomplishment</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>State</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Future</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Achievement</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Accomplishment</td>
<td>1</td>
<td>7</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>State</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

Table A.1: Syntactic Structure of Main clause of *in* Sentences

Table [A.1] presents the distribution of the syntactic structure of the main clause of a hundred temporal *in* sentences.
APPENDIX A.  SYNTACTIC ANALYSIS OF TEMPORAL MODIFIERS

Table A.2: Syntactic Structure of Main clause of on Sentences
Table A.2 presents the distribution of the syntactic structure of the main clause of in a hundred temporal on sentences.

<table>
<thead>
<tr>
<th></th>
<th>Perf</th>
<th>Prog</th>
<th>PerfProg</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>Achievement</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Accomplishment</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>State</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Perf</th>
<th>Prog</th>
<th>PerfProg</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>Achievement</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Accomplishment</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>State</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Table A.3: Syntactic Structure of Main clause of at Sentences
Table A.3 below presents the distribution of the syntactic structure of a hundred temporal at sentences.

<table>
<thead>
<tr>
<th></th>
<th>Perf</th>
<th>Prog</th>
<th>PerfProg</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
<td>4</td>
<td>2</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>Achievement</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>Accomplishment</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>State</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Perf</th>
<th>Prog</th>
<th>PerfProg</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Achievement</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Accomplishment</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>State</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Perf</th>
<th>Prog</th>
<th>PerfProg</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>7</td>
</tr>
<tr>
<td>Achievement</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Accomplishment</td>
<td>2</td>
<td>5</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>State</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>
Table A.4: Syntactic Structure of Main clause of by Sentences
Table A.4 presents the syntactic structures of fifty sentences modified by the temporal prepositions by.

<table>
<thead>
<tr>
<th></th>
<th>Perf</th>
<th>Prog</th>
<th>PerfProg</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Achievement</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Accomplishment</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>State</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Perf</th>
<th>Prog</th>
<th>PerfProg</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Achievement</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Accomplishment</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>State</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Perf</th>
<th>Prog</th>
<th>PerfProg</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Achievement</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Accomplishment</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>State</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Table A.5: Syntactic Structure of Main clause of before Sentences
Table A.5 presents the distribution of syntactic structure of the main clause of the 158 instances of temporal before.
### Table A.6: Syntactic Structure of Main clause of *after* Sentences

Table A.6 presents the distribution of syntactic structure main clause of the two hundred instances of temporal *after*.

<table>
<thead>
<tr>
<th></th>
<th>Past</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Perf</td>
<td>Prog</td>
<td>PerfProg</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Activity</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Achievement</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>Accomplishment</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Present</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Achievement</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>Accomplishment</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Future</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity</td>
<td>1</td>
<td>13</td>
<td>0</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>Achievement</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Accomplishment</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>

### Table A.7: Syntactic Structure of Main clause of *until* Sentences

Table A.7 presents the syntactic structure of the main clause of 167 temporal instances of *until* phrases.

<table>
<thead>
<tr>
<th></th>
<th>Past</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Perf</td>
<td>Prog</td>
<td>PerfProg</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Activity</td>
<td>0</td>
<td>15</td>
<td>0</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Achievement</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Accomplishment</td>
<td>0</td>
<td>18</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>26</td>
<td></td>
</tr>
<tr>
<td>Present</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity</td>
<td>2</td>
<td>8</td>
<td>0</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Achievement</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Accomplishment</td>
<td>3</td>
<td>14</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>Future</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activity</td>
<td>7</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Achievement</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Accomplishment</td>
<td>0</td>
<td>13</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>State</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>15</td>
<td></td>
</tr>
</tbody>
</table>
Table A.8: Syntactic Structure of Main clause of *since* Sentences
Table A.8 presents the syntactic structure of the main clause of 93 temporal instances of *until* phrases.
Appendix B

Controlled Natural Language with Temporal Features

In this section we present in part the code for a controlled natural language with temporal features.

B.1 Lexicon

We show in this section how the lexical items in our lexicon are defined.

B.1.1 Determiners

\[
\text{lex(det, [syntax:[a], num:X, class:Y, type:indef]):-}
\]
\[
X = \text{sgl},
\]
\[
(Y = \text{nontemp}; Y = \text{temp}).
\]

\[
\text{lex(det, [syntax:[an], num:X, class:Y, type:indef]):-}
\]
\[
X = \text{sgl},
\]
\[
(Y = \text{temp}; Y = \text{nontemp}).
\]

\[
\text{lex(det, [syntax:[the], num:X, class:Y, type:def]):-}
\]
\[
(X = \text{sgl}; X = \text{pl}),
\]
\[
(Y = \text{temp}; Y = \text{nontemp}).
\]

\[
\text{lex(det, [syntax:[some], num:X, class:Y, type:indef]):-}
\]
B.1. LEXICON

B.1.2 Nouns

\[
\text{lex(noun,[nlpr:boy, syntax:[X], num:Y, type:count]):} \]
\[
(X = \text{boy}, \hspace{1cm} Y = \text{sgl}); \hspace{1cm} (X = \text{boys}, \hspace{1cm} Y = \text{pl}).
\]

\[
\text{lex(noun,[nlpr:book, syntax:[X], num:Y, type:count]):} \]
\[
(X = \text{book}, \hspace{1cm} Y = \text{sgl}); \hspace{1cm} (X = \text{books}, \hspace{1cm} Y = \text{pl}).
\]

\[
\text{lex(noun,[nlpr:student, syntax:[X], num:Y, type:count]):} \]
\[
(X = \text{student}, \hspace{1cm} Y = \text{sgl}); \hspace{1cm} (X = \text{students}, \hspace{1cm} Y = \text{pl}).
\]

\[
\text{lex(noun,[nlpr:Y, syntax:[Y], num:sgl, type:proper]):} \]
\[
Y = \text{john}; \hspace{1cm} Y = \text{mary}; \hspace{1cm} Y = \text{jane}.
\]

\[
\text{lex(noun,[nlpr:day, syntax:[X], num:Y, type:interval]):} \]
\[
(X = \text{day}, \hspace{1cm} Y = \text{sgl}); \hspace{1cm} (X = \text{days}, \hspace{1cm} Y = \text{pl}).
\]

\[
\text{lex(noun,[nlpr:lecture, syntax:[X], num:Y, type:event]):} \]

(X = lecture, 
  Y = sgl);
(X = lectures, 
  Y = pl).

lex(noun, [nlpr:meeting, syntax:[X], num:Y, type:event]):-
  (X = meeting, 
   Y = sgl);
  (X = meetings, 
   Y = pl).

lex(noun, [nlpr:Y, syntax:[Y], num:sgl, type:month]):-
  Y = january; Y = july;
  Y = february; Y = august;
  Y = march; Y = september;
  Y = april; Y = october;
  Y = may; Y = november;
  Y = june; Y = december.

lex(noun, [nlpr:Y, syntax:[Y], num:sgl, type:day]):-
  Y = monday; Y = tuesday;
  Y = wednesday; Y = thursday;
  Y = friday; Y = saturday;
  Y = sunday.

B.1.3 Verbs

lex(verb, [nlpr:write, syntax:[Syn], num:Num, tense:Tense, 
  class:Class, type:Type]):-
  (Class = acc; Class = act),
  (Type = tran; Type = intran),
  ((Num = sgl; Num = pl),
    ((Tense = past, Syn = wrote);
       (Tense = participle, Syn = written);
       (Tense = progressive, Syn = writing);
       (Tense = future, Syn = write));
(Tense = untensed, Num = pl, Syn = write);
(Tense = untensed, Num = sgl, Syn = writes)).

lex(verb, [nlpr:arrive, syntax:[Syn], num:Num, tense:Tense, class:Class, type:Type]) :-
  (Class = ach),
  Type = intran,
  ((Num = sgl; Num = pl),
   ((Tense = past, Syn = arrived);
    (Tense = participle, Syn = arrived);
    (Tense = progressive, Syn = arriving);
    (Tense = future, Syn = arrive));
  (Tense = untensed, Num = pl, Syn = arrive);
  (Tense = untensed, Num = sgl, Syn = arrives)).

B.1.4 Auxiliary Verbs

lex(aux, [syntax:[has], num:Num, tense:Tense, pol:pos, type:have]) :-
  Num=sgl,
  (Tense = present; Tense = future).

lex(aux, [syntax:[has, not], num:Num, tense:Tense, pol:neg, type:have]) :-
  Num=sgl,
  (Tense = present; Tense = future).

lex(aux, [syntax:[have], num:Num, tense:Tense, pol:pos, type:have]) :-
  Num=pl,
  (Tense = present; Tense = future).

lex(aux, [syntax:[will, have], num:Num, tense:Tense, pol:pos, type:have]) :-
  (Num=pl; Num=sgl),
  (Tense = present; Tense = future).

lex(aux, [syntax:[will, not, have], num:Num, tense:Tense, pol:neg,
APPENDIX B. CONTROLLED NATURAL LANGUAGE WITH TEMPORAL FEATURES

```
lex(aux,[syntax:[have, not],num:Num,tense:Tense, pol:neg, type:have]):-
   Num=pl, (Tense = present; Tense = future).
lex(aux,[syntax:[had],num:Num,tense:past, pol:pos, type:have]):-
   Num = sgl; Num=pl.
lex(aux,[syntax:[had, not],num:Num,tense:past, pol:neg, type:have]):-
   Num = sgl; Num=pl.
lex(aux,[syntax:[will], num:Num, tense:future, pol:pos, type:will]):-
   Num = sgl; Num=pl.
lex(aux,[syntax:[will, not], num:Num, tense:future, pol:neg, type:will]):-
   Num = sgl; Num=pl.
lex(aux,[syntax:[is], num:sgl, tense:present, pol:pos, type:be]).
lex(aux,[syntax:[is, not], num:sgl, tense:present, pol:neg, type:be]).
lex(aux,[syntax:[was], num:sgl, tense:past, pol:pos, type:be]).
lex(aux,[syntax:[was, not], num:sgl, tense:past, pol:neg, type:be]).
lex(aux,[syntax:[are], num:pl, tense:present, pol:pos, type:be]).
```
B.2. SEMANTICS

In this section, we present the semantic interpretation of lexical items in our language.

```
lex(aux,[syntax:[are, not], num:pl, tense:present, pol:neg, type:be]).

lex(aux,[syntax:[were], num:pl, tense:past, pol:pos, type:be]).

lex(aux,[syntax:[were, not], num:pl, tense:past, pol:pos, type:be]).

lex(aux,[syntax:[be],num:pl, tense:Tense, pol:pos, type:be]):-
Tense = untensed; Tense = future.

lex(aux,[syntax:[be],num:sgl, tense:nil, pol:pos, type:be]).

lex(aux,[syntax:[do, not], num:pl, tense:present, pol:neg, type:do]).

lex(aux,[syntax:[does, not], num:pl, tense:present, pol:neg, type:do]).

lex(aux,[syntax:[did, not], num:Num, tense:present, pol:neg, type:do]):-
Num = sgl; Num = pl.
```

B.1.5 Temporal Prepositions/Conjunctions

```
lex(tempPrep, [syntax:[Y], type:Y]):-
Y = after; Y = before;
Y = in; Y = on;
Y = at; Y = for;
Y = by; Y = until;
Y = while; Y = since;
Y = during.
```

B.2 Semantics

In this section, we present the semantic interpretation of lexical items in our language.
B.2.1 Determiners

`semLex(det, M):- M = [type:uni, sem:lbd(p,lbd(q,forall(x,(p@x -> q@x)))).]

`semLex(det, M):- M = [type:indef, class:nontemp, sem:lbd(p,lbd(q,exists(x,(p@x & q@x)))).]

`semLex(det, M):- M = [type:indef, class:temp, sem:lbd(p,lbd(q,exists(i,(p@i&(i subeq j)& q@i)))).]

`semLex(det, M):- M = [type:def, class:nontemp, sem:lbd(p, lbd(q, exists(x, p@x & forall(y, p@y->y = x)&q@x)))).]

`semLex(det, M):- M = [type:def, class:temp, sem:lbd(p, lbd(q, exists(i, p@i & forall(i1, p@i1->i1 = i)& (i subeq j)&q@i)))).]

`semLex(det, M):- M = [type:uni, class:temp, sem:lbd(p,lbd(q,forall(i,(p@i &(i subeq j) -> q@i)))).]

`semLex(det, M):- M = [type:uni, class:nontemp, sem:lbd(p,lbd(q,forall(i,(p@x -> q@x)))).]

B.2.2 Noun

`semLex(noun, M):- M = [nlpr:Sym, sem:lbd(x,Fla),type:count], compose(Fla, Sym, [x]).
B.2. SEMANTICS

semLex(noun, M):-
    M = [nlpr: Sym, sem: lbd(p, p@Sym), type: proper].

semLex(noun, M):-
    M = [nlpr: Sym, sem: lbd(p, p@Sym & (Sym subeq j)), type: month].

semLex(noun, M):-
    M = [nlpr: Sym, sem: lbd(p, p@Sym & (Sym subeq j)), type: time].

semLex(noun, M):-
    M = [nlpr: Sym, sem: lbd(p, p@Sym & (Sym subeq j)), type: day].

semLex(noun, M):-
    M = [nlpr: Sym, sem: lbd(i, Fla), type: event],
    compose(Fla, Sym, [i]).

semLex(noun, M):-
    M = [nlpr: Sym, sem: lbd(i, Fla), type: interval],
    compose(Fla, Sym, [i]).

compose(Term, Sym, ArgList):- Term =..[Sym — ArgList].

B.2.3 Verbs

semLex(verb, M):-
    M = [nlpr: Sym, inf: untensed, sem: lbd(x2, lbd(x, x2@lbd(x1, Fla))), type: tran],
    compose(Fla, Sym, [x, x1]).

semLex(verb, M):-
    M = [nlpr: Sym, inf: untensed, sem: lbd(x, Fla), type: intran],
    Fla=..[Sym, x].

semLex(verb, M):-
    M = [nlpr: Sym, inf: tensed,
B.2.4 Tense

semLex(tense, M):-
Fla = t⊗i,
M = [sem:lbdt, lbd(x, exists(i,forall(i, (i subeq j) &
Fla@x & (i < now))))], inf:future].

semLex(tense, M):-
Fla = t⊗i,
M = [sem:lbdt, lbd(x,forall(i, (i subeq j) &
Fla@x & (now < i) &
(i subeq j))))], inf:future].

semLex(tense, M):-
Fla = t⊗i,
M = [sem:lbdt, lbd(x, exists(i1,forall(i, (i1 subeq j) &
(Fla@x & (i < now)))))], inf:future].

semLex(tense, M):-
Fla = t⊗i,
M = [sem:lbdt, lbd(x,forall(i, (i subeq j) &
(Fla@x & (now < i) &
(i subeq j)))))), inf:future].

semLex(tense, M):-
Fla = t⊗i,
M = [sem:lbdt, lbd(x, exists(i1,forall(i, (i1 subeq j) &
(Fla@x & (now < i) &
(i subeq i1) & (i subeq j))))), inf:future].

semLex(tense, M):-
Fla = t⊗i,
M = [sem:lbdt, lbd(x, exists(i,forall(i, (i subeq j) &
(Fla@x & (now < i) &
(i subeq j))))), inf:future].

semLex(tense, M):-
Fla = t⊗i,
M = [sem:lbdt, lbd(x, exists(i,forall(i, (i subeq j) &
(Fla@x & (now < i) &
(i subeq j))))), inf:future].

semLex(tense, M):-
Fla = t⊗i,
M = [sem:lbdt, lbd(x, exists(i,forall(i, (i subeq j) &
(Fla@x & (now < i) &
(i subeq j))))), inf:future].

semLex(tense, M):-
Fla = t⊗i,
M = [sem:lbdt, lbd(x, exists(i,forall(i, (i subeq j) &
(Fla@x & (now < i) &
(i subeq j))))), inf:future].

semLex(tense, M):-
Fla = t⊗i,
M = [sem:lbdt, lbd(x, exists(i,forall(i, (i subeq j) &
(Fla@x & (now < i) &
(i subeq j))))), inf:future].
B.2. SEMANTICS

semLex(tense, M):-
   Fla = t@i,
   M = [sem:lbd(t, lbd(x,forall(i, (i subeq j)->
        Fla@x&(now < i)))), inf:past].

B.2.5 Temporal Prepositions

semLex(tempPrep,M):-
   M = [type:in, sem:lbd(z, lbd(p, z@ lbd(i, p@i)))]).

semLex(tempPrep, M):-
   M = [type:at, sem:lbd(z, lbd(p, z@ lbd(i, p@i)))]).

semLex(tempPrep, M):-
   M = [type:on, sem:lbd(z, lbd(p, z@ lbd(i, p@i)))]).

semLex(tempPrep, M):-
   M = [type:during, sem:lbd(z, lbd(p, z@ lbd(i, p@i)))]).

semLex(tempPrep, M):-
   M = [type:while, sem:lbd(z, lbd(p, z@ lbd(i, p@i)))]).

semLex(tempPrep,M):-
   Fla =..[init,i,j],
   M = [type:before, sem:lbd(p, z@lbd(i, p@Fla))].

semLex(tempPrep,M):-
   Fla =..[init,i,j],
   M = [type:by, sem:lbd(p, z@lbd(i, p@Fla))].

semLex(tempPrep,M):-
   Fla =..[init,i,j],
   M = [type:until, sem:lbd(p, z@lbd(i, p@Fla))].

semLex(tempPrep,M):-
B.3 Grammar

This section provides grammar rules that govern the generation of various syntactic categories.

Clauses of the form (448) are implementations of α-reduction – that is it simply renames bounded variables. Clauses of the form (449) are implementations of lambda calculus β-reduction, such that the lambda term A is applied to the lambda term B to generate the C.

(448) var_replace(A, B)

(449) beta(A@B, C)

The following rules therefore show the generation of the semantic interpretation of a given category from the semantic interpretation of its constituents.

B.3.1 Determiners

det([sem:Sem, num:X, class:Y, type:Z])-->
{
    lex(det,[syntax:Word, num:X, class:Y, type:Z])
},
Word,
{
    semLex(det, [type:Z, class:Y, sem:Sem])
}.

B.3.2 Nouns

n([sem:Sem, num:X, type:Y])-->

Fla =..[comp,i,j],
M = [type:after, sem:1bd(p, z@1bd(i, p@Fla))].

semLex(tempPrep,M):-
    Fla =..[comp,i,j],
    M = [type:since, sem:1bd(p, z@1bd(i, p@Fla))].
B.3. GRAMMAR

\{
    \text{lex}(\text{noun}, [\text{nlpr}: \text{Sym}, \text{syntax}: \text{Word}, \text{num}: X, \text{type}: Y])
\},
\text{Word},
\{
    \text{semLex}(\text{noun}, [\text{nlpr}: \text{Sym}, \text{sem}: \text{Sem}, \text{type}: Y])
\}.

B.3.3 Noun Phrases

np([\text{sem}: \text{Sem}, \text{num}: X, \text{type}: Y]) \rightarrow
n([\text{sem}: \text{Sem}, \text{num}: X, \text{type}: Y]),
\{
    Y = \text{proper};
    Y = \text{month};
    Y = \text{year};
    Y = \text{time};
    Y = \text{day}
\}.

np([\text{sem}: \text{Sem}, \text{num}: \text{Num}, \text{type}: \text{Type}]) \rightarrow
det([\text{sem}: \text{DetSem}, \text{num}: \text{Num}, \text{class}: \text{Class}, \text{type}: \text{DetType}]),
n([\text{sem}: \text{NSem}, \text{num}: \text{Num}, \text{type}: \text{Type}]),
\{
    ((\text{Class} = \text{nontemp}, \text{Type} = \text{count});
    (\text{Class} = \text{temp}, (\text{Type} = \text{event}; \text{Type} = \text{interval})))),
    (\text{DetType} = \text{def}; \text{DetType} = \text{indef}; \text{DetType} = \text{uni}),
    \text{var_replace}(\text{DetSem}, \text{DetSem1}),
    \beta(\text{DetSem1}@$\text{NSem}$, \text{Sem})
\}.

np([\text{sem}: \text{Sem}, \text{num}: \text{Num}, \text{type}: \text{Type}]) \rightarrow
n([\text{sem}: \text{NSem}, \text{num}: \text{Num}, \text{type}: \text{Type}]),
\{
    \text{Num} = \text{pl},
    \text{Type} = \text{count},
\}.
APPENDIX B. CONTROLLED NATURAL LANGUAGE WITH TEMPORAL FEATURES

\[
\text{var\_replace(lbd(p,lbd(q,\text{exists}(x,(p@x & q@x)))), \text{DetSem1}),} \\
\beta(\text{DetSem1@NSem, Sem})
\]

\]

B.3.4 Auxiliary Verbs

\[
\text{aux([num:Num, tense:Tense, pol:Pol, type:Type])}\rightarrow \\
\begin{cases} \\
\text{lex(aux, [syntax:Word, num:Num, tense:Tense, pol:Pol, type:Type])} \\
\end{cases}
\]

B.3.5 Verbs

\[
\text{v([sem:Sem, num:Num, tense:untensed, class:Class, type:Type])}\rightarrow \\
\begin{cases} \\
\text{lex(verb, [nlpr:Sym, syntax:Word, num:Num, tense:untensed,} \\
\text{class:Class, type:Type])} \\
\end{cases}
\]

\[
\text{v([sem:Sem, num:Num, tense:Tense, class:Class, type:Type])}\rightarrow \\
\begin{cases} \\
\text{lex(verb, [nlpr:Sym, syntax:Word, num:Num, tense:Tense,} \\
\text{class:Class, type:Type]),} \\
(Tense = \text{past}; \\
Tense = \text{future}; \\
Tense = \text{progressive}; \\
Tense = \text{participle}) \\
\end{cases}
\]


semLex(verb, [nlpr:Sym, inf:tensed, sem:Sem, type:Type])
]
}

B.3.6 Tense
tense([sem:Sem, inf:past]) --> []
{
    semLex(tense,[sem:Sem, inf:past])
}.

tense([sem:Sem, inf:future]) --> []
{
    semLex(tense,[sem:Sem, inf:future])
}.

tense([sem:Sem, inf:future]) --> [will],
{
    semLex(tense,[sem:Sem, inf:future])
}.

B.3.7 Verb Phrases

vb([sem:Sem, num:Num, class:Class, tense:Tense]) -->
    v([sem:Sem, num:Num, tense:Tense, class:Class, type:intran]),
    {
        (Tense = participle; Tense = progressive)
    }.

vb([sem:Sem, num:Num, class:Class, tense:Tense]) -->
    v([sem:VSem, num:Num, tense:Tense, class:VClass, type:tran]),np([sem:NPSem, num:NPNum, type:Type]),
    {
        (Type == proper; Type==count),
        (Tense == participle; Tense == progressive),
        aspectClassConverter(VClass, NPNum, Class),
        var_replace(VSem, VSem1),
    }.
APPENDIX B. CONTROLLED NATURAL LANGUAGE WITH TEMPORAL FEATURES

\[
\text{beta}(\text{VSem1@NPSem}, \text{Sem})
\]

\[
vbar([\text{sem}: \text{Sem}, \text{num}: \text{Num}, \text{class}: \text{Class}, \text{pol}: \text{Pol}, \text{tense}: \text{Tense}]) \rightarrow
\begin{align*}
\text{aux}(\text{[num}: \text{Num}, \text{tense}: \text{AuxTense}, \text{pol}: \text{Pol}, \text{type}: \text{AuxType}]), \\
\text{vb}(\text{[sem}: \text{Sem}, \text{num}: _, \text{class}: \text{Class}, \text{tense}: \text{VBTense}]), \\
\end{align*}
\]
\[
\begin{align*}
\{ \\
\text{AuxTense} &= \text{untensed}, \\
\text{VBTense} &= \text{participle}, \\
\text{AuxType} &= \text{have}, \\
\text{atom_concat}(\text{AuxTense}, \text{VBTense}, \text{Tense}); \\
\end{align*}
\]

\[
\text{vbar}(\text{[sem}: \text{Sem}, \text{num}: \text{Num}, \text{class}: \text{Class}, \text{pol}: \text{pos}, \text{tense}: \text{Tense}]) \rightarrow
\begin{align*}
\text{v}(\text{[sem}: \text{Sem}, \text{num}: \text{Num}, \text{tense}: \text{Tense}, \text{class}: \text{Class}, \text{type}: \text{intran}]); \\
\end{align*}
\]
\[
\begin{align*}
\{ \\
\text{Tense} &= \text{past} \\
\end{align*}
\]

\[
vbar([\text{sem}: \text{Sem}, \text{num}: \text{Num}, \text{class}: \text{Class}, \text{pol}: \text{pos}, \text{tense}: \text{Tense}]) \rightarrow
\begin{align*}
\text{v}(\text{[sem}: \text{VSem}, \text{num}: \text{Num}, \text{tense}: \text{Tense}, \text{class}: \text{VClass}, \text{type}: \text{tran}]), \\
\text{np}(\text{[sem}: \text{NPSem}, \text{num}: \text{NPNum}, \text{type}: \text{NPType}]), \\
\end{align*}
\]
\[
\begin{align*}
\{ \\
\text{NPType} &= \text{proper}; \text{NPType} = \text{count}, \\
\text{aspectClassConverter}(\text{VClass}, \text{NPNum}, \text{Class}), \\
\text{(Tense} \neq \text{participle}; \text{Tense} \neq \text{progressive}, \\
\text{var_replace}(\text{VSem}, \text{VSem1}), \\
\text{beta}(\text{VSem1@NPSem}, \text{Sem}) \\
\}
\]

\[
\text{vp([sem}: \text{Sem}, \text{num}: \text{Num}, \text{class}: \text{Class}, \text{pol}: \text{pos}, \text{tense}: \text{Tense}]) \rightarrow
\begin{align*}
\text{v}[\text{[sem}: \text{Sem}, \text{num}: \text{Num}, \text{tense}: \text{Tense}, \text{class}: \text{Class}, \text{type}: \text{Type}]), \\
\end{align*}
\]
B.3. GRAMMAR

{ 
  Tense = untensed,
  Type = intran

}.

vp([sem:Sem, num:Num, class:Class, pol:pos, tense:Tense])--> 
v([sem:VSem, num:Num, tense:Tense, class:VClass, type:Type]),
  np([sem:NPSem, num:NPNum, type:NPType]),
  
  (NPType = count; NPType = proper),
  Type = tran,
  aspectClassConverter(VClass, NPNum, Class),
  var_replace(VSem, VSem1),
  beta(VSem1@NPSem, Sem)
  
}.

vp([sem:Sem, num:Num, class:Class, pol:pos, tense:Tense])--> 
  
  (((Tense = past; Tense = future),
    var_replace(TSem, TSem1),
    beta(TSem1@VSem, Sem))
  
}.

vp([sem:Sem, num:Num, class:Class, pol:pos, tense:Tense])--> 
  
  (((VTense = pastparticiple; VTense = pastProgressive),
    Tense = past);
  ((VTense = futureparticiple; VTense = futureProgressive),


Tense = future;
(VTense = presentprogressive, Tense = future),
var_replace(TSem, TSem1),
beta(TSem1@VSem, Sem)
}

senTemp([sem:lbd(t, lbd(x, lbd(q, exists(i, (t@i)x&
  (i subeq j)&q@i))))]) --> [].

compVerb([sem:Sem, num:Num, class:Class, pol:pos, tense:Tense])--> 
  senTemp([sem:STSem]),
  vbar([sem:VSem, num:Num, class:Class, pol:pos, tense:Tense]),
  {
    var_replace(STSem, STSem1),
    beta(STSem1@VSem, Sem)
  }.
aspectClassConverter(acc, sgl, acc).
aspectClassConverter(acc, pl, act).
aspectClassConverter(state, sgl, state).

B.3.8 Temporal Prepositions

tp([sem:Sem, type:Type])--> 
  {
    lex(tempPrep, [syntax:Word, type:Type])
  },
Word,
  {
    semLex(tempPrep, [type:Type, sem:Sem])
  }.

B.3.9 Temporal Prepositional Phrases

tpp([sem:Sem, type:Type])--> 
  tp([sem:TPSem, type:Type]), np([sem:NPSem, num:_, type:NPType]),
B.3. GRAMMAR

{ 
  (Type = during,
  (NPType = interval; NPType = event));
  (Type = in,
  (NPType = month; NPType = year));
  (Type = on,
  NPType = day);
  (Type = at,
  (NPType = time));
  (Type = before,
  (NPType = month; NPType = year; NPType = day;
  NPType = time; NPType = interval; NPType = event));
  (Type = by,
  (NPType = month; NPType = year; NPType = day; NPType = time));
  (Type = until,
  (NPType = month; NPType = year; NPType = day; NPType = time));
  (Type = since,
  (NPType = month; NPType = year; NPType = day; NPType = time)),
  var_replace(TPSem, TPSem1),
  beta(TPSem1@NPSem, Sem)
}. 

B.3.10 Sentences

sen([sem:Sem, num:Num, tense:untensed, class:Class, pol:pos,
  type:unmodified])-->
np([sem:NPSem, num:Num, type:Type]),
vp([sem:VPSem, num:Num, class:Class, pol:pos, tense:untensed]),
{ 
  (Type=count; Type=proper),
  var_replace(NPSem,NPSem1),
  beta(NPSem1@VPSem, Sem)
}. 

sen([sem:Sem, num:Num, tense:Tense, class:Class, pol:pos,
type:unmodified])-->
np([sem:NPSem, num:Num, type:Type]),
vp([sem:VPSem, num:Num, class:Class, pol:pos, tense:Tense]),
{
    (Type=count; Type=proper),
    var_replace(NPSem,NPSem1),
    beta(NPSem1@VPSem, Sem)
}.

sen([sem:Sem, num:Num, tense:Tense, class:Class, pol:pos,
    type:modified]) -->
    sen([sem:SSem, num:Num, tense:Tense, class:Class, pol:pos,
        type:unmodified]),
tpp([sem:TPPSem, type:Type]),
{
    (Type = during; Type = before; Type = after;
    Type = in; Type = on; Type = at),
    var_replace(TPPSem,TPPSem1),
    beta(TPPSem1@lbd(j,SSem), Sem)
}.

sen([sem:Sem, num:Num, tense:Tense, class:Class, pol:pos,
    type:modified]) -->
    sen([sem:SSem, num:Num, tense:Tense, class:Class, pol:pos,
        type:unmodified]),
tpp([sem:TPPSem, tense:Tense, type:Type]),
{
    ((Type = until, (Class = state; Class = act))
    (Type = until, (Class = state; Class = act, Class = ach));
    (Type = before, (Class = state; Class = act; Class = ach;
    Class = acc));
    (Type = after, (Class = state; Class = act; Class = ach;
    Class = acc));
    (Type = while, (Class = state; Class = act; Class = acc)))
    var_replace(TPPSem,TPPSem1),
B.3. GRAMMAR

\[
\begin{align*}
\text{beta}(\text{TPPSem}1@\text{lbd}(j, SSem), \text{Sem}) \\
\text{}. \\
\end{align*}
\]

\[
\text{compSen}([\text{sem}: \text{Sem}, \text{num}: \text{Num}, \text{tense}: \text{Tense}, \text{class}: \text{Class}, \text{pol}: \text{pos}]) \rightarrow \\
\text{np}([\text{sem}: \text{NPSem}, \text{num}: \text{Num}, \text{type}: \text{Type}]), \\
\text{compVerb}([\text{sem}: \text{VPSem}, \text{num}: \text{Num}, \text{class}: \text{Class}, \text{pol}: \text{pos}, \text{tense}: \text{Tense}]), \\
\{ \\
\text{(Type=count; Type=proper)}, \\
\text{var_replace}(	ext{NPSem}, \text{NPSem}1), \\
\text{beta}(\text{NPSem}1@\text{VPSem}, \text{Sem}) \\
\}.
\]
References


Byamugisha J., Keet C. M., DeRenzi B. Bootstrapping a Runyankore CNL from an isiZulu CNL // Controlled Natural Language. 2016. 25–36.


Kaljurand K., Alumäe T. Controlled Natural Language in Speech Recognition Based User Interfaces // Controlled Natural Language: Third International Workshop. 2012. 79–94.


Murray W. R., Singliar T. Spatiotemporal Extensions to a Controlled Natural Language // Controlled Natural Language. 2012. 61–78.


REFERENCES


Tesnière L. Elements de syntaxe structurale. 1959.

