Logics for Natural Language Inference
(expanded version of lecture notes from a course
at ESSLLI 2010)

Lawrence S. Moss
November 2010
Contents

Chapter 1. Introduction 1
  1.1. Logic for Natural Language, Logic in Natural Language 1

Chapter 2. Basic Syllogistic Proof Systems 5
  2.1. The Simplest Fragment “of All” 5
  2.2. All and Some 10
  2.3. All and No 14
  2.4. A Logic for $S$ 16
  2.5. Additional Exercises 17

Chapter 3. Syllogistic Logic with Complements: $S^\dagger$ 21
  3.1. Background: Completeness of Basic Sentential Logics 21
  3.2. Adding Sentential Boolean Operations to $L(\text{all, some})$ 27
  3.3. Syntax and Semantics of $S^\dagger$ 30
  3.4. Completeness via Representation of Orthoposets 33
  3.5. Adding Boolean Connectives on Sentences: $S^\dagger_{bc}$ 38

Chapter 4. Cardinality Comparisons 41
  4.1. Adding $\exists \geq$ to the Boolean Syllogistic Fragment 43
  4.2. Digression: Most 45
  4.3. On the Numerical Syllogistic 47

Chapter 5. Verbs: $R$ 49
  5.1. A Logic for $R$ 52
  5.2. There is no Finite Syllogistic Logic for $R$ 59
  5.3. An Alternative for the Positive Fragment 61
  5.4. Incorporating Background Facts, I 64

Chapter 6. Relative Clauses: $RC$ 67
  6.1. $RC$: an Indirect System 67
  6.2. Complexity Results for $RC$, $R^\dagger$, and $RC^\dagger$ 70
  6.3. Complete Subsystems 74
  6.4. Incorporating Background Facts, II 75

Chapter 7. Comparative Adjectives 77
  7.1. Requiring Transitivity in $R$ 78
  7.2. Requiring Transitivity and Irreflexivity in $RC$ 80

Chapter 8. Logic Beyond the Aristotle Border 89
  8.1. Fitch-style Proof System for $RC^\dagger$ 89
  8.2. Adding Transitivity: $L(adj)$ 100
<table>
<thead>
<tr>
<th>Chapter 9. Monotonicity and Polarity</th>
<th>107</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.1. Introduction</td>
<td>107</td>
</tr>
<tr>
<td>9.2. Background: Categorial Grammar</td>
<td>109</td>
</tr>
<tr>
<td>9.3. Background: Preorders and their Opposites</td>
<td>111</td>
</tr>
<tr>
<td>9.4. Higher-order Terms over Preorders</td>
<td>114</td>
</tr>
<tr>
<td>9.5. Examples and Discussion</td>
<td>117</td>
</tr>
</tbody>
</table>

Bibliography  
Index
CHAPTER 1

Introduction

1.1. Logic for Natural Language, Logic in Natural Language

Logic is one of the oldest subjects in the western intellectual tradition. It was initiated by Aristotle and it served alongside grammar and rhetoric as a component of the medieval trivium. It has an illustrious contemporary connections to topics as diverse as rhetoric, theoretical computer science, and infinity. It is today pursued by philosophers, mathematicians, and computer scientists. In more detail, philosophers have always been interested in the central themes of logic: reasoning, representation, meaning, and proof. Around 1900, the field of logic became heavily mathematical, and indeed today there are whole fields of mathematics which are offshoots of logic. Part of the mathematical interest of logic came about by presentations of logic as a foundation of mathematics, and the same is happening today with computer science. Indeed, almost every area of computer science makes use of logic, just as most areas of physics end up using branches of mathematics such as the calculus. And so computer science is a vital source of work in logic.

The specific focus of these notes is the development of areas of logic with an eye towards natural language. Although logic often is presented as a study of reasoning in language, its main contemporary thrusts are to applications in computer science, mathematics, or philosophy. For example, students of logic might learn that All cats are animals might be symbolized ($\forall x(\text{cat}(x) \rightarrow \text{animal}(x))$) and that All who love all animals love all cats might be symbolized as

$$(\forall y)((\forall x(\text{animal}(x) \rightarrow \text{love}(y, x))) \rightarrow ((\forall x)(\text{cat}(x) \rightarrow \text{love}(y, x))))$$.

In a good first course, the same students might even learn that the second sentence follows logically from the first. But this notion of “follows logically” is defined entirely on the logical representations, not on the English sentences with which we started. The interaction of logic and language is a neglected area, and so our topic area is outside the mainstream of work in logic. But our view is that the topic is of vital interest in several fields. In a sense, the project goes back to Aristotle's syllogisms. However, we have in mind a much broader and deeper set of research questions: whereas syllogistic reasoning only applies to arguments in very restricted forms, we would like to study reasoning as close to “surface forms” (real sentences) as possible. Instead of limiting attention to categorical (yes/no) reasoning, we eventually hope to develop connections to default reasoning and reasoning under uncertainty. However, these are not treated in these notes.

These notes present logical systems that attempt to model aspects of natural language inference. Most of these are listed on the map on the next page, but to keep the chart readable I have left off several of them. Indeed, to keep the notes themselves readable, I have tried to keep the chapters fairly short by concentrating
on the main ideas in the logical systems instead of presenting all the details, and also by omitting some proofs entirely. The chart mostly consists of logical languages, each with an intended semantics. For example, FOL at the top is first-order logic. (Very short explanations of the systems appear to the right of the vertical line.) In the chart, the lines going down mean “subsystem,” and sometimes the intention is that we allow a translation preserving the intended semantics. So all of the systems in the chart may be regarded as subsystems of FOL with the exception of $S^\geq$.

The smallest system in the chart is $S$, a system even smaller than the classical syllogistic. The syntax of $S$ is quite impoverished and contains only sentences of the kinds All $p$ are $q$ and Some $p$ are $q$, where $p$ and $q$ are variables. It is our intent that these variables be interpreted as plural common nouns of English, and that a model $M$ be an arbitrary set $M$ together with interpretations of the variables by subsets of $M$. 
Moving up, $S_{\geq}$ adds additional sentences of the form \textit{there are at least as many} $p$ \textit{as} $q$. The additions are not expressible in FOL, and we indicate this by setting $S_{\geq}$ on the “outside” of the “Peano-Frege Boundary.”

The language $S^1$ adds full negation on nouns to $S$. For example, one can say \textit{all} $p$ \textit{are} $\neg q$ with the intended reading “no $p$ are $q$.” One can also say \textit{All} $\neg p$ \textit{are} $\neg q$, and this goes beyond what is usually done in the syllogistic logic. The use of the $\dag$ notation will be maintained throughout these notes as a mark of systems which contain complete noun-level negation.

Moving up the chart, let us discuss the systems $R$, $RC$, $R^\dag$, and $RC^\dag$. The system $R$ extends $S$ by adding verbs, interpreted as arbitrary relations. (The ‘$R$’ stands for ‘relation.’) This system and the others in this paragraph originate in Pratt-Hartmann and Moss [29]. So $R$ would contain \textit{Some} dogs chase no cats, and the yet larger system $RC$ would contain relative clauses as exemplified in \textit{All who love all animals love all cats}. The languages with the dagger such as $S^1$ and $R^\dag$ are further enrichments which allow subject nouns to be negated. This is rather unnatural in standard speech, but it would be exemplified in sentences like Every \textit{non-dog runs}. The point: the dagger fragments are beyond the Aristotle boundary in the sense that they cannot be treated by the relatively simpler syllogistic logics. The only known logical systems for them use variables in a key way. The line marked “Aristotle” separates the logical systems above the line, systems which can be profitably studied on their own terms without devices like variables over individuals, from those which cannot.

Continuing, the chart continues with the additions of comparative adjective phrases with the systems $RC(tr)$ and $RC^\dag(tr)$.

In addition to the logical systems on the charts, the notes go further by considering logical systems that directly attempt to

Previous work. There are two main predecessor projects to the proposed project. The first is a line of work begun by Johan van Benthem on a calculus of monotonicity. It would take us too far afield to present the basic ideas here, but perhaps an example would suffice. From \textit{All cats are animals}, we might infer by monotonicity that \textit{Someone who owns a cat owns an animal}. To inference of \textit{All who love all cats love all animals} is by monotonicity in the downward direction. The original idea is to propose a systematic account of monotonicity inferences in language, an account which works on something closer to real sentences than to logic representations. This is a goal which we share in these notes, but the bulk of our work is not concentrated on the monotonicity calculus.

The second predecessor project studies logics for fragments of language. Many of the results The chart above is a summary of results in several papers in the area, written by the proposer and also in a key paper jointly with Pratt-Hartmann. Listed on the chart are a number of tiny pieces of English with names like $S$, $R$, etc.

Origin of these notes. These notes are an expanded and modified version of the notes for my ESSLLI 2010 course. My intention was, and still is, to present much of what I know about logical systems for linguistic reasoning and to outline what I think are important research topics and directions. The notes are based on papers on logical systems for natural language inference that were written in the past years, put together with a uniform notation, point of view, and tone.

For slides of my lectures, please see
http://logicforlanguage.blogspot.com/

My lectures generally did not cover everything in the slides, and the slides also cover only a part of what is in these notes. I hope the notes will be useful for two different kinds of readers: First, people who want it should be able to get the big picture without being lost in a sea of details. Second, those who really want to dive into that sea should be able to do so without drowning. For this, I have augmented the proofs with more motivation and examples than one would find in a journal publication, and at the same time I have relegated parts of long proofs to exercises.

Much of these notes is concerned with logical completeness results. We'll have more to say about what these are and why we are interested in them in Chapter 2. However, we are also interested in issues of computational complexity pertaining to the logical systems themselves. And here is a version of our map that presents the results.

As you can see, this complexity chart mentions several logical languages that are not present on the first chart.
CHAPTER 2

Basic Syllogistic Proof Systems

2.1. The Simplest Fragment “of All”

We begin our notes with the simplest logical fragment whatsoever. The sentences are all of the form All \( p \) are \( q \). So we have a very impoverished language, with only one kind of sentence. But we’ll have a precise semantics, a proof system, and a soundness/completeness theorem which relates the two.

2.1.1. Syntax and semantics. For the syntax, we start with a collection \( P \) of unary atoms (for nouns). We write these as \( p, q, \ldots \). Then the sentences of this first fragment are the expressions

\[
\text{All } p \text{ are } q,
\]

where \( p \) and \( q \) are any atoms in \( P \). We call this language \( L(\text{all})^1\).

The semantics is based on models.

**Definition 2.1** A model \( M \) for this fragment \( L(\text{all}) \) is a structure

\[
M = (M, \llbracket \ ] \rrbracket)
\]

consisting of a set \( M \), together with an interpretation \( \llbracket p \rrbracket \subseteq M \) for each noun \( p \in P \). The main semantic definition is truth in a model:

\[
M \models \text{All } p \text{ are } q \iff \llbracket p \rrbracket \subseteq \llbracket q \rrbracket
\]

We read this in various ways, such as \( M \) satisfies All \( p \) are \( q \), or All \( p \) are \( q \) is true in \( M \).

From this definition, we get two further notions: If \( \Gamma = \varnothing \) is a set of sentences, we say that \( M \models \Gamma \) iff \( M \models \varphi \) for every \( \varphi \in \Gamma \). Finally, we say that \( \Gamma \models \varphi \) iff whenever \( M \models \Gamma \), also \( M \models \varphi \). We read this as \( \Gamma \) logically implies \( \varphi \), or \( \Gamma \) semantically implies \( \varphi \), or that \( \varphi \) is a semantic consequence of \( \Gamma \).

**Example 2.2** Here is an example of the semantics. Let \( M = \{1, 2, 3, 4, 5\} \). Let \( \llbracket n \rrbracket = \emptyset \), \( \llbracket p \rrbracket = \{1, 3, 4\} \), and \( \llbracket q \rrbracket = \{1, 3\} \). This is all we need to specify a model \( M \). Then the following sentences are true in \( M \): All \( n \) are \( n \), All \( n \) are \( p \), All \( n \) are \( q \), All \( q \) are \( p \), and All \( q \) are \( q \). (In the first two of these example sentences, we use the fact that the empty set is a subset of every set.) The other four sentences using \( n \), \( p \), and \( q \) are false in \( M \).

---

1Note that \( L(\text{all}) \) is really a family of languages, one for each set \( P \) of nouns at the outset. This dependence on primitives is true for practically all logical systems. For most of the work in these notes, we suppress mention of the primitive syntactic items because it makes the notation lighter and because we rarely need to call attention to the primitives in the first place.
Example 2.3 Here is an example of a semantic consequence which can be expressed in \( L(\text{all}) \): We claim that

\[
\{ \text{All } p \text{ are } q, \text{All } n \text{ are } p \} \models \text{All } n \text{ are } q.
\]

To see this, we give a straightforward mathematical proof in natural language. Let \( M \) be any model for \( L(\text{all}) \), assuming that the underlying set \( \mathcal{P} \) contains \( n, p, \) and \( q \). Assume that \( M \) satisfies \( \text{All } p \text{ are } q \) and \( \text{All } n \text{ are } p \). We must prove that \( M \) also satisfies \( \text{All } n \text{ are } q \). From our first assumption, \( [p] \subseteq [q] \). From our second, \( [n] \subseteq [p] \). It is a general fact about sets that the inclusion relation (written as \( \subseteq \) here) is transitive, and so we conclude that \( [n] \subseteq [q] \). This verifies that indeed \( M \) satisfies \( \text{All } n \text{ are } q \). And since \( M \) was arbitrary, we are done.

Example 2.4 Finally, we have an example of a failure of semantic consequence:

\[
\{ \text{All } p \text{ are } q \} \not\models \text{All } q \text{ are } p.
\]

To show that a given set \( \Gamma \) does not logically entail another sentence \( \varphi \), we need to build a model \( M \) of \( \Gamma \) which is not a model of \( \varphi \). In this example, \( \Gamma \) is \( \{ \text{All } p \text{ are } q \} \), and \( \varphi \) is \( \text{All } q \text{ are } p \). We can get a model \( M \) that does the trick by setting \( M = \{1, 2\} \), \( [p] = \{1\} \), and \( [q] = \{1, 2\} \). For that matter, we could also use a different model, say \( N \), defined by \( N = \{61\} \), \( [p] = \emptyset \), and \( [q] = \{61\} \).

2.1.2. Proof system. We construct a proof system for this language based on the rules shown in Figure 2.1 and the definition of a proof tree.

Definition 2.5 A proof tree over \( \Gamma \) is a finite tree \( T \) whose nodes are labeled with sentences, and each node is either an element of \( \Gamma \), or comes from its parent(s) by an application of one of the rules.

\( \Gamma \vdash \varphi \) means that there is a proof tree \( T \) over \( \Gamma \) whose root is labeled \( \varphi \). We read this as \( \Gamma \) proves \( \varphi \), or \( \Gamma \) derives \( \varphi \), or that \( \varphi \) follows in our proof system from \( \Gamma \).

Example 2.6 Here is an example, chosen to make some several points: Let \( \Gamma \) be

\[
\{ \text{All } l \text{ are } m, \text{All } q \text{ are } l, \text{All } m \text{ are } p, \text{All } n \text{ are } p, \text{All } l \text{ are } q \}
\]

Let \( \varphi \) be \( \text{All } q \text{ are } p \). Here is a proof tree showing that \( \Gamma \vdash \varphi \):

\[
\begin{align*}
\text{All } q \text{ are } l & \quad \text{All } l \text{ are } m & \quad \text{All } m \text{ are } p \\
\text{All } q \text{ are } p & \quad \text{All } l \text{ are } m & \quad \text{All } m \text{ are } p \\
\end{align*}
\]

Note that all of the leaves belong to \( \Gamma \) except for one: that is \( \text{All } m \text{ are } m \). Note also that some elements of \( \Gamma \) are not used as leaves. This is permitted according to our definition. The proof tree above shows that \( \Gamma \vdash \varphi \). Also, there is a smaller proof.
2.1. THE SIMPLEST FRAGMENT “OF ALL”

A tree that does this, since the use of All \( m \) are \( m \) is not really needed. (The reason why we allow leaves to be labeled like this is so that we can have one-element trees labeled with sentences of the form All \( l \) are \( l \).)

The main theoretical question for this chapter is: what is the relation the semantic notion \( \Gamma \models S \) with the proof-theoretic notion \( \Gamma \vdash S \)? This kind of question will present itself for all of the logical systems in this course. Probably the first piece of work for you is to be sure you understand the question.

Lemma 2.7 (Soundness) If \( \Gamma \vdash \varphi \), then \( \Gamma \models \varphi \).

Proof By strong induction on the number of nodes of proof trees \( T \) over \( \Gamma \). If \( T \) is a tree with one node, let \( \varphi \) be the label. Either \( \varphi \) belongs to \( \Gamma \), or else \( \varphi \) is of the form All \( p \) are \( p \). In the first case, every model satisfying every sentence in \( \Gamma \) clearly satisfies \( \varphi \), as \( \varphi \) belongs to \( \Gamma \). And in the second case, every model whatsoever satisfies \( \varphi \).

Let’s suppose that we know our result for all proof trees over \( \Gamma \) with less than \( n \) nodes, and let \( T \) be a proof tree over \( \Gamma \) with \( n \) nodes. The argument breaks into cases depending on which rule is used at the root. Suppose the root and its parents are labeled All \( p \) are \( q \) All \( q \) are \( n \) All \( p \) are \( q \).

Let \( T_1 \) and \( T_2 \) be the subtrees ending at All \( p \) are \( q \) and All \( q \) are \( n \). Then \( T_1 \) and \( T_2 \) are proof trees over \( \Gamma \) themselves. For some unary atom \( q \), the root of \( T_1 \) is labeled All \( p \) are \( q \), and the root of \( T_2 \) is labeled All \( q \) are \( n \). Now \( T_1 \) and \( T_2 \) both have fewer nodes than \( T \). By our induction hypothesis, \( \Gamma \models All \( p \) are \( q \) \), and also \( \Gamma \models All \( q \) are \( n \) \). We claim that \( \Gamma \models All \( p \) are \( n \) \). To see this, take any model \( M \) in which all sentences in \( \Gamma \) are true. Then \( \{p\} \subseteq \{q\} \) by our first point above. And \( \{q\} \subseteq \{n\} \) by second. So \( \{p\} \subseteq \{n\} \) by transitivity of the inclusion relation on sets. Since the model \( M \) here is arbitrary, we conclude that \( \Gamma \models All \( p \) are \( n \) \). □

So at this point, we know that our logic is sound: If we have a tree showing that \( \Gamma \vdash \varphi \), then \( \varphi \) follows semantically from \( \Gamma \). This means that the formal logical system is not going to give us any bad results. Now this is a fairly weak point. If we dropped some of the rules, it would still hold. Even if we decided to be conservative and say that \( \Gamma \vdash \varphi \) never holds, the soundness fact would still be true. So the more interesting question to ask is whether the logical system is strong enough to prove everything it conceivably could prove. We want to know if \( \Gamma \models \varphi \) implies that \( \Gamma \vdash \varphi \). If this implication does hold for all \( \Gamma \) and \( \varphi \), then we say that our system is complete!logical system. Here is a summary of our definitions.

Definition 2.8 A proof system is sound if \( \Gamma \vdash \varphi \) implies that \( \Gamma \models \varphi \).

The same proof system is complete if the converse holds: if \( \Gamma \models \varphi \) implies that \( \Gamma \vdash \varphi \).

Before we turn to the completeness proof itself, we make a general definition that will reappear throughout these notes.

Definition 2.9 Let \( \Gamma \) be a set of sentences in any fragment containing All. Define \(^{2}\) \( u \leq v \) to mean that

\[
(2.1) \quad \Gamma \vdash All \ u \ are \ v.
\]

\(^{2}\)It is important to note that this relation \( \leq \) is defined in terms of \( \Gamma \), but that the notation \( \leq \) hides this dependence. We could write it as \( \leq_{\Gamma} \), and this might make it easier on readers the
Proposition 2.10 \( \leq \) is a preorder: it is reflexive and transitive relation on the set \( P \) of unary atoms.

**Proof** The reflexivity comes from the fact that we permit single-node trees to be proofs of statements \( \text{All} \ u \ \text{are} \ u \). For the transitivity, assume that \( u \leq v \leq w \). Then we have proof trees for \( \text{All} \ u \ \text{are} \ v \) and for \( \text{All} \ v \ \text{are} \ w \). All of whose leaves are in \( \Gamma \). Putting the trees together and using (Barbara) at the root gives a proof tree showing that \( \Gamma \vdash \text{All} \ u \ \text{are} \ w \). This means that \( u \leq w \), as desired. \( \square \)

We are going to use Proposition 2.10 frequently in these notes, and often without explicitly mentioning it.

**Example 2.11** Let

\[
\Gamma = \left\{ \begin{array}{c}
\text{All } j \text{ are } k, \\
\text{All } j \text{ are } l, \\
\text{All } k \text{ are } l, \\
\text{All } l \text{ are } m, \\
\text{All } k \text{ are } n, \\
\text{All } m \text{ are } q, \\
\text{All } p \text{ are } q, \\
\text{All } q \text{ are } p
\end{array} \right\}
\]

Then the associated preorder \( \leq \) is shown below:

```
   p, q
  /   \
 m   k, l
 /     \
j
```

For example, since \( j \leq k \), we draw \( j \) below \( k \). Note also that \( k \leq l \) and \( l \leq k \). We indicate this in the picture by situating \( k \) and \( l \) together.

**Theorem 2.12** The logic of Figure 2.1 is complete for \( L(\text{all}) \).

**Proof** Suppose that \( \Gamma \models \text{All } p \text{ are } q \). Build a model \( M \), taking \( M \) to be the set of unary atoms. The rest of the structure is given by down-sets: \( [u] = \downarrow u \). That is,

\[
[u] = \{ v : u \leq v \}.
\]

**Claim** \( M \models \Gamma \).

**Proof** Suppose that \( \text{All } u \text{ are } v \) belongs to \( \Gamma \); that is \( u \leq v \). We need to show that \( [u] \subseteq [v] \). But this is just a general fact about down-sets in preorders: if \( x \in [u] \), then \( x \leq u \). So by transitivity, \( x \leq v \). \( \square \)
Recall that we are assuming that $\Gamma \models \text{All } p \text{ are } q$, and that we are proving that $\Gamma \vdash \text{All } p \text{ are } q$. Hence for the $p$ and $q$ in our statement, $[p] \subseteq [q]$. But we have a one-point tree showing that $\Gamma \vdash \text{All } p \text{ are } p$. This goes to show that $p \in [p]$. And so $p \in [q]$; this means that $p \leq q$. But this is exactly what we want: $\Gamma \vdash \text{All } p \text{ are } q$.

This completes the proof. □

**Example 2.13** We continue a discussion begun in Example 2.11. The model $M$ obtained following the proof of Theorem 2.12 has $M = \{j, k, l, m, n, p, q\}$ and also

- $[j] = \{j\}$
- $[k] = \{j, k, l\}$
- $[l] = \{j, k, l\}$
- $[m] = \{j, k, l, m\}$
- $[n] = \{j, k, l, n\}$
- $[p] = \{j, k, l, m, p, q\}$
- $[q] = \{j, k, l, m, p, q\}$

One can check that $M \models \Gamma$. But in contrast, consider a sentence which is not a semantic consequence of $\Gamma$, such as $\text{All } n \text{ are } k$. We can check that $[n] \not\subseteq [k]$ in our model $M$. The point is that our proof gives something stronger than completeness: it gives a single model $M \models \Gamma$ such that for all $\varphi$ such that $\Gamma \not\vdash \varphi$, $M \not\models \varphi$. That is, the same model falsifies all sentences which do not follow from $\Gamma$.

**2.1.3. Comments.** There are a few comments to be made at this point. These are based on questions which I have received in teaching this material, or similar material to bright students.

1. *Doesn’t this proof confuse syntax and semantics?* This is a good question, since we are building a model out of the “material from the syntax” (in this case, the unary atoms). But when one thinks about it, there is nothing wrong with building a model out of chairs, numbers, abstract objects, or even the same objects that we used in the syntax. It is more interesting that the interpretation function $[\ ]$ in the model was defined in terms of a syntactic notion, the proof system. Again, we are free to define the semantics of a model any way we like. It is interesting that we prove completeness in this way. But in a sense it should not be such a surprise. For completeness is about a relation between syntax and semantics, and so it makes sense that it involves a single structure that has aspects of both.

2. *I thought that the semantic assertion $\Gamma \models \varphi$ meant that all models of $\Gamma$ are again models of $\varphi$. How is it that we only argue completeness on one particular model rather than many models?* It’s true that our semantic assertion $\Gamma \models \varphi$ is a statement about all models. But this does not mean that in proving something we need to use all the models. In a sense, the question can be turned around to make an observation: the model $M$ we built from a set $\Gamma$ is as “bad” as any model could possibly be! It makes true any sentence which does not follow from $\Gamma$. So once we have built a single model that covers for all the models, we can use that one model to prove completeness.

**2.1.4. A Stronger Result.** Theorem 2.12 proves the completeness of the logical system. But it doesn’t give us all the information we would need to have an efficient procedure to decide whether or not $\Gamma \vdash \varphi$ in this fragment.

---

3The reason is that we still have to examine all possible models on a one element set in order to check whether $\Gamma \models \varphi$ or not. It might seem at first glance that there are very few such models. But if the sequent $\Gamma \models \varphi$ contains $k$ atoms, then there are $2^k$ models to consider. The question of efficient decidability is for us the question of whether a polynomial-time algorithm exists. For that, we need to do further work.
2. BASIC SYLLOGISTIC PROOF SYSTEMS

Some $p$ are $q$  Some $p$ are $q$

Some $q$ are $p$  Some $p$ are $p$

All $q$ are $n$  Some $p$ are $q$

Some $p$ are $n$

Figure 2.2. The logic of Some and All, in addition to the logic of All.

**Theorem 2.14** Let $\Gamma$ be any set of sentences in $L(\text{all})$, let $\preceq^*$ be defined from $\Gamma$ as above. Let $p$ and $q$ be any atoms. Then the following are equivalent:

1. $\Gamma \vdash \text{All } p \text{ are } q$.
2. $\Gamma \models \text{All } p \text{ are } q$.
3. $p \preceq^* q$.

**Proof** (1)$\implies$(2) is by soundness, and (3)$\implies$(1) is by induction on $\preceq^*$. The most significant part (2)$\implies$(3). Consider the model $M$ whose universe is a singleton $\{\ast\}$, and with $[[z]] = M$ iff $p \preceq^* z$. We claim that all sentences in $\Gamma$ are true in $M$. Consider All $v$ are $w$. We may assume that $[[v]] = M$, or else our claim is trivial. Then $p \preceq^* v$. But $v \preceq w$, so we have $p \preceq^* w$, as desired. This verifies that $M \models \Gamma$. And since $[[p]] = M$, we have $[[q]] = M$ as well. Thus $p \preceq^* q$, as needed for (3). \(\square\)

The original definition of the entailment relation $\Gamma \models \varphi$ involves looking at all models of the language. Theorem 2.14 is important because part (3) gives a criterion the entailment relation that is algorithmically sensible. To see whether $\Gamma \models \text{All } p \text{ are } q$ or not, we only need to construct $\preceq^*$. This is the reflexive-transitive closure of a syntactically defined relation, so it is computationally very manageable.

2.2. All and Some

We move from the language $L(\text{all})$ to a bigger language $L(\text{all}, \text{some})$ by adding sentences of the form Some $p$ are $q$. We call these new sentences existentials, since formalizing them in first-order logic would use the existential quantifier $\exists$.

The semantics for the existential sentences is the obvious one:

$M \models \text{Some } p \text{ are } q$ iff $[[p]] \cap [[q]] \neq \emptyset$.

We already have a proof system for All, given in the last section. Since the logic with All and Some is a bigger system, we keep the rules for All. We merely extend the logical system by adding the rules in Figure 2.2.

**Example 2.15** The first important derivation in the logic:

\[
\begin{array}{c}
\text{All } n \text{ are } p & \text{Some } n \text{ are } n \\
\hline
\text{Some } p \text{ are } n
\end{array}
\]

\[
\begin{array}{c}
\text{All } n \text{ are } q & \text{Some } p \text{ are } n \\
\hline
\text{Some } p \text{ are } q
\end{array}
\]

That is, if there is a $n$, and if all $ns$ are $ps$ and also $qs$, then some $p$ is a $q$. 
DEFINITION 2.16 If Γ is a set of sentences in \(\mathcal{L}(all, some)\), we write \(\Gamma_{all}\) for the sentences in \(\Gamma\) of the form \(All\ p\ are\ q\), and \(\Gamma_{some}\) for the sentences of the form \(Some\ p\ are\ q\).

**Lemma 2.17** Let \(\Gamma \subseteq \mathcal{L}(all, some)\). Then there is a model \(M\) with the following properties:

1. \(M \models \Gamma\).
2. If \(\varphi\) is any sentence in \(\mathcal{L}(some)\) and \(M \models \varphi\), then \(\Gamma \vdash \varphi\).

**Proof** For the universe \(M\) of the model, we take \(\Gamma_{some}\). For an atom \(p\), set
\[
[u] = \{\varphi : \text{some } v \leq u \text{ occurs in } \varphi\}.
\]
(Again, \(\leq\) is defined in (2.1).) In other words,
\[
All\ x\ are\ y\ belongs\ to\ [u]\ iff\ \text{either } x \leq u\ or\ y \leq u.
\]
This defines the model \(M\).

We check that \(M \models \Gamma\). Consider a sentence \(All\ x\ are\ y\ in \Gamma\). Then \(x \leq y\). The transitivity of \(\leq\) implies that \([x] \subseteq [y]\). Second, consider an existential sentence \(\varphi\) and assume that \([p] \cap [q] \neq \emptyset\). Let \(Some\ x\ are\ y\ belong\ to\ this\ set\). Since \(Some\ x\ are\ y\ belongs\ to\ [p]\), either \(x \leq p\) or \(y \leq p\).

Similarly, either \(x \leq q\) or \(y \leq q\). At this point, our proof splits into four cases. One case is when \(x \leq p\) and \(y \leq q\). Recalling that \(Some\ x\ are\ y\ belongs\ to\ \Gamma\), we have a proof tree as follows:
\[
\begin{align*}
\vdots & \quad Some\ x\ are\ y \\
\vdots & \quad Some\ y\ are\ x \\
\begin{array}{c}
\vdots \\
\vdots \\
All\ x\ are\ p & Some\ y\ are\ x \\
\end{array} & \begin{array}{c}
\vdots \\
\vdots \\
Some\ y\ are\ p & Some\ p\ are\ y \\
\end{array} \\
\begin{array}{c}
\vdots \\
\vdots \\
All\ y\ are\ q \\
\end{array} & \begin{array}{c}
\vdots \\
\vdots \\
Some\ p\ are\ q \\
\end{array} \\
\end{align*}
\]
(2.3)
The other cases are similar. You might like to check the details to see where the second rule of \(\mathcal{L}(some)\) gets used.

**Theorem 2.18** The system in Figures 2.1 and 2.2 is complete for \(\mathcal{L}(all, some)\).

**Proof** Suppose that \(\Gamma \models \varphi\). There are two cases, depending on whether \(\varphi\) is of the form \(All\ p\ are\ q\) or of the form \(Some\ p\ are\ q\). The cases are handled differently. We leave the first to you as Exercise 1. The second follows immediately from Lemma 2.17.

**Example 2.19** Let us check that
\[
\{\text{Some } p\ are\ q,\ Some\ q\ are\ n,\ All\ q\ are\ m\} \not\models \text{Some } p\ are\ n
\]
by building a model in which the hypotheses hold and the conclusion fails. Let us do this following the proof of Lemma 2.17.

Let \(\varphi\) be \(Some\ p\ are\ q\), and let \(\psi\) be \(Some\ q\ are\ n\). We take for a model \(M = \{\varphi, \psi\}\) with \([p] = \{\varphi\}, [q] = \{\varphi, \psi\}, [n] = \{\psi\}, [m] = \{\varphi, \psi\}\). These come from (2.2). It is clear that \(M\) has the desired properties. Although one could build such a model by hand, our general definitions show how to do this in a perfectly general way.
Exercise 1  Complete the proof of Theorem 2.18 by showing that if $\Gamma$ is a set of sentences in All and Some, and if $\Gamma \models All\ p\ are\ q$, then also $\Gamma \vdash All\ p\ are\ q$. [It will be easier to modify the proof of Theorem 2.12 to account for existential sentences than to use Lemma 2.17.]

Exercise 2  Let $\Gamma$ be a set of sentences in All and Some, and let $\varphi$ be a sentence in Some. As we know from Lemma 2.17, if $\Gamma \not\models \varphi$, there is a $M \models \Gamma$ which makes $\varphi$ false. The proof gets a model $M$ whose size of the $M$ will be the number of existential sentences in $\Gamma$. Can we do better?

1. Show that there is a model as desired whose size is at most 2.
2. Show that 2 is the smallest we can get by showing that if we only look at one-element models, then

$$\{\text{Some}\ p\ are\ q, \text{Some}\ q\ are\ n}\models \text{Some}\ p\ are\ q$$

Exercise 3  Let $\Gamma$ be the set $\{All\ p\ are\ q\}$. Prove that there is no model $M$ such that for all sentences $\varphi$ in the fragment of this section, $M \models \varphi$ iff $\Gamma \vdash \varphi$. The point of this problem is that there is some $M'$ with the property that for all $\varphi$ of the form Some $u$ are $v$, $M \models \varphi$ iff $\Gamma \vdash \varphi$. But it is not possible to extend this result to sentences with All. So we cannot hope to avoid the split in the proof of Theorem 2.18 due to the syntax of $\varphi$.

Exercise 4  Give an algorithm which takes finite sets $\Gamma$ in the fragment of this section and also single sentences $\varphi$ and tells whether $\Gamma \models \varphi$ or not. [You may be sketchy, as we were in our discussion of this matter at the end of Section 2.1.]

Exercise 5  Suppose that one wants to say that All $p$ are $q$ is true when $[p] \subseteq [p]$ and also $[p] \neq \emptyset$. Then the following rule becomes sound:

$$\begin{align*}
\text{All}\ p\ are\ q \\
\text{Some}\ p\ are\ q
\end{align*}$$

Show that if we add this rule to the proof system for this section, then we get a complete system for the modified semantics. [Hint: Given $\Gamma$, let $\Gamma'$ be $\Gamma$ together with all sentences Some $p$ are $q$ such that All $p$ are $q$ belongs to $\Gamma$. Show that $\Gamma \vdash \varphi$ in the modified system iff $\Gamma \vdash \varphi$ in the old system.]

Exercise 6  Check that

$$\{\text{Some}\ p\ are\ q, \text{Some}\ q\ are\ n, All\ q\ are\ m\} \not\models \text{Some}\ p\ are\ n$$

by examining proofs. [We have seen in Exercise 2.19 that the hypotheses do not semantically imply the conclusion, and since the proof system is sound, we know that the hypotheses cannot derive the conclusion in the proof system. So this exercise is unnecessary. Still, it is a good exercise in working with a proof system.]

Exercise 7  This exercise asks you to come up with definitions and to check their properties.

1. Define the appropriate notions of submodel and homomorphism of models.
2. Which sentences $\varphi$ in our language have the property that if $M$ is a submodel of $M'$ and $M' \models \varphi$, then also $M \models \varphi$?
3. Which sentences $\varphi$ in our language have the property that if $M$ is a surjective homomorphic image of $M'$ and $M \models \varphi$, then also $M' \models \varphi$?
2.2. **ALL AND SOME**

(4) Would anything change if we changed “if” to “iff”?

**Exercise 8** What would you do to the system to add sentences of the form *Some p exists?*

### 2.2.1. Alternative notation: a first look.

Recall that our syntax begins with a set \( P \) of *unary atoms* that we think of as simple nouns in English such as *cat, animal*, etc. A *unary literal* is an expression of either of the forms \( p \) or \( \overline{p} \), where \( p \) is a unary atom. A unary literal is called *positive* if it is a unary atom; otherwise, *negative*. We continue to use letters like \( p, q, x, y \), etc., to range over unary atoms, and \( l, m, n \) to range over unary literals. With these conventions, an *S-formula* is an expression of any of the forms

\[
\exists(p,l), \quad \exists(l,p), \quad \forall(p,l), \quad \forall(l,\overline{p}).
\]

We provide English glosses for S-formulas as follows:

\[
\begin{align*}
\forall(p,q) & \quad \text{Every } p \text{ is a } q \quad \forall(p,\overline{q}) & \quad \text{No } p \text{ is a } q \\
\exists(p,q) & \quad \text{Some } p \text{ is a } q \quad \exists(p,\overline{q}) & \quad \text{Some } p \text{ is not a } q.
\end{align*}
\]

Note that \( S \) contains the formulas \( \exists(\overline{p},q) \) and \( \forall(\overline{p},\overline{q}) \), which are not glossed in (2.5); however, according to the semantics below, these formulas are logically equivalent to \( \exists(q,\overline{p}) \) and \( \forall(q,p) \), respectively. We may regard \( S \) as the language of the traditional syllogistic.

### 2.2.2. Adding Names.

It is natural to add *names* for individuals to the logic; we want to proceed with larger and larger fragments, and names would seem to be one of the first things to add. We are going to add names in this section, but for the most part in these notes, we shall set names aside. The reason for this is that it is usually routine to add names to our fragments, but it makes our work easier to avoid them.

To our formal language we add names \( J, M, \ldots \), (for *John* and *Mary*) \( J_1, \ldots, J_n, \ldots \), etc. The sentences we add to the fragment are *\( J \) is a \( p \) and \( J \) is \( M \)*, where \( J \) and \( M \) are names and \( p \) is an atom. The semantics is that names denote elements of models \( M \). That is, we require that \([J] \in M\) for all names \( J \).

Now we turn to the proof system. Since we are going to add names to \( \mathcal{L}(\text{all, some}) \), the rules of the logic for that language are still sound. So we use the rules in Figures 2.1 and 2.2, and for the names we add the rules in Figure 2.3.

![Figure 2.3. The logic of names, on top of the rules in Figures 2.1 and 2.2.](image-url)
Fix a set $\Gamma$ of sentences in this fragment. Let $\equiv$ be the relation on names defined by
\begin{equation}
J \equiv M \text{ iff } \Gamma \vdash J \text{ is } M.
\end{equation}

**Lemma 2.20** $\equiv$ is an equivalence relation.

**Lemma 2.21** Let $\Gamma$ be any set of sentences in Some, All, and names. Then there is a model $M$ with the following properties:
1. $M \models \Gamma$.
2. If $\varphi$ is any sentence in Some or names and $M \models \varphi$, then $\Gamma \vdash \varphi$.

**Proof** As before, we define $\leq$ to be from (2.1). We also have the equivalence relation $\equiv$ from (2.6). Let the set of equivalence classes of $\equiv$ be $[J_1], \ldots, [J_m]$. We take $M$ to be $\Gamma_{\text{some}} \cup \{[J_1], \ldots, [J_m]\}$. We assume these sets are disjoint.

We define
\begin{equation}
[Z] = \{i : \text{either } V_i \leq Z \text{ or } W_i \leq Z\} \\
\cup \{[J] : \text{for some } M \in [J], \Gamma \vdash M \text{ is a } Z\}
\end{equation}
To finish defining our model, we take $[J] = [J]$. That is, the semantics of $J$ is the equivalence class $[J]$. It is easy to see that the semantics is monotone in the sense that if $V \leq W$, then $[V] \subseteq [W]$. This implies that all of the universal assertions of $\Gamma$ are true in our model $M$. The existential assertions in $\Gamma$ are Some $V_i$ is $W_i$ for $i \leq n$, and for each $i$, the number $i$ belongs to $[V_i] \cap [W_i]$. The identity sentences $J$ is $M$ from $\Gamma$ are clearly true in $M$. Finally, consider a sentence $J$ is a $n$ in $\Gamma$. Then $\Gamma \vdash J$ is a $n$. So $[J] = [J] \in [Z]$. This means that our sentence $J$ is a $n$ is true in $M$.

Let $M \models \text{Some } p \text{ are } q$. If there is some number $i$ in $[X] \cap [Y]$, then the proof of Theorem 2.18 shows that $\Gamma \vdash \text{Some } p \text{ are } q$. The only alternative is when for some $J$, $[J] \in [X] \cap [Y]$. By the definition in (2.7), there are $M \in [J]$ and $N \in [J]$ such that $\Gamma \vdash M \text{ is a } p$ and $\Gamma \vdash N \text{ is a } q$. We thus have a proof tree over $\Gamma$:

\[
\frac{\vdots \quad \vdots \quad M \text{ is a } p \quad J \text{ is } J \quad M \text{ is } N \quad \vdots}{M \text{ is a } p \quad \text{Some } p \text{ are } q}
\]

So $\Gamma \vdash \text{Some } p \text{ are } q$, as desired.

Continuing, let $M \models J$ is $M$. Then $[J] = [M]$. So by Lemma 2.20, $\Gamma \vdash J$ is $M$.

Finally, suppose $M \models J$ is a $p$. Then for some $M$, $M \in [J]$ and $\Gamma \vdash M$ is a $p$. So we see that $\Gamma \vdash J$ is a $p$ using the last rule in Figure 2.3. 

**Exercise 9** Prove that the logic of Figures 2.1, 2.2, and 2.3 is complete for All, Some, and names.

**2.3. All and No**

In this section, we consider the fragment $L(\text{all, no})$ which contains sentences of the form All $p$ are $q$ and No $p$ are $q$. We use the same kinds of models that we have already seen, and interpret the new sentences by

\[
M \models \text{No } p \text{ are } q \quad \text{iff} \quad [p] \cap [q] = \emptyset.
\]
To make a proof system for this language, we use the rules of Figure 2.1 and also the rules in Figure 2.4.

**Example 2.22** We check that

\[ \{ \text{All } p \text{ are } v, \text{All } q \text{ are } w, \text{No } v \text{ are } w \} \vdash \text{No } p \text{ are } q. \]

\[
\begin{array}{c}
\text{All } p \text{ are } v & \text{All } q \text{ are } w \\
\hline
\text{No } p \text{ are } w \\
\text{No } w \text{ are } p \\
\hline
\text{No } p \text{ are } q
\end{array}
\]

(2.8)

**Lemma 2.23** Let \( \Gamma \) be any set of sentences in \( \mathcal{L}(\text{all, no}) \). Then there is a model \( M \) with the following properties:

1. \( M \models \Gamma \).
2. If \( \varphi \) is any sentence in \( \mathcal{L}(\text{all, no}) \), and \( M \models \varphi \), then \( \Gamma \vdash \varphi \).

**Proof** Recall that all our syntactic notions are built on a set \( P \) of atoms (for nouns). We take for \( M \) the set of sets \( A \subseteq P \) satisfying the following conditions:

1. \( A \) is up-closed: If \( v \in A \) and \( v \leq w \), then \( w \in A \).
2. \( A \) is \( \text{ALL} \): If \( v, w \in A \), then \( \Gamma \not\vdash \text{No } v \text{ are } w \).

(Note as a special case of (2.10) that if \( v \in A \), then \( \Gamma \not\vdash \text{No } v \text{ are } v \).) We set

\[
[v] = \{ A \in M : v \in A \}.
\]

We claim that \( M \models \Gamma \). Condition (2.9) implies that if \( \text{All } v \text{ are } w \) belongs to \( \Gamma \), then \( [v] \subseteq [w] \). Suppose that \( \text{No } v \text{ are } w \) belongs to \( \Gamma \). Let \( A \in [v] \), so that \( v \in A \). By condition (b), \( w \notin A \). So \( A \notin [w] \). This argument shows that \( [v] \cap [w] = \emptyset \).

With (1) proved, we turn to (2). Let \( M \models \varphi \). We first deal with the case that \( \varphi \) is the of the form \( \text{All } p \text{ are } q \). Let

\[
A = \{ z : p \leq z \}.
\]

Case I: \( A \notin M \). Then there are \( v, w \in A \) such that \( \Gamma \vdash \text{No } v \text{ are } w \). In this case, we use the result of Example 2.22 (with \( q = p \)) to see that \( \Gamma \vdash \text{No } p \text{ are } p \).

As a result, we have \( \Gamma \vdash \varphi \).

Case II: \( A \in M \). Then since \( A \in [p] \), we have \( A \in [q] \). The semantics in (2.11) tells us that \( A \in [q] \). Thus \( \Gamma \vdash \text{All } p \text{ are } q \), as desired.

This concludes our work when \( \varphi \) is \( \text{All } p \text{ are } q \). Suppose that \( \varphi \) is \( \text{No } p \text{ are } q \). Let

\[
A = \{ z : p \leq z \text{ or } q \leq z \}.
\]

Note that \( p, q \in A \). We claim that \( A \notin M \). For if \( A \in M \), we would have \( A \in [p] \cap [q] \). And then \( [p] \cap [q] \neq \emptyset \). But this contradicts \( \Gamma \models \varphi \). So indeed, \( A \notin M \). Hence there are \( v, w \in A \) such that \( \Gamma \vdash \text{No } v \text{ are } w \). There are four cases, depending on whether \( \Gamma \vdash \text{All } p \text{ are } v \) or \( \Gamma \vdash \text{All } q \text{ are } v \), and similarly for \( w \).

Case I: \( \Gamma \vdash \text{All } p \text{ are } v \), and \( \Gamma \vdash \text{All } q \text{ are } w \). It follows from Example 2.22 that \( \Gamma \vdash \varphi \).

Case II: \( \Gamma \vdash \text{All } p \text{ are } v \), and \( \Gamma \vdash \text{All } p \text{ are } w \).

From Example 2.22 (with \( q = p \)), we have from \( \Gamma \) that \( \text{No } p \text{ are } p \). So we also have \( \text{No } p \text{ are } q \).

The remaining two cases are similar. \[\square\]
16 2. BASIC SYLLOGISTIC PROOF SYSTEMS

![Syllogistic Proof System Table]

Figure 2.4. The logic of No p are q on top of All p are q.

**Theorem 2.24** The logic of Figures 2.1 and 2.4 is complete for $L(\text{all, no})$.

**2.4. A Logic for $S$**

The language $S$ contains the syllogistic sentences in All, Some, No which we have seen. We could also call the language $L(\text{all, some, no})$, following what we have been doing earlier. But since we think of this language (and its relative $S^1$) in connection with syllogisms, this is why we use the letter $S$.

We know the semantics of $S$, and it remains to join the proof systems which we have already seen in Figures 2.1, 2.2, 2.3, and 2.4. We also must add a principle relating Some and No. For the first time, we face the problem of potential inconsistency: to say Some $p$ are $q$ is to deny that No $p$ are $q$. There are no models of Some $p$ are $q$ and No $p$ are $q$. We see that according to our semantics, any sentence $\varphi$ whatsoever follows from these two. We thus add the rule of ex falso quodlibet (also called ex contradictione quodlibet) to our system $4$. We call this rule (X).

\[
(2.12) \quad \frac{\text{Some } p \text{ are } q \quad \text{No } p \text{ are } q}{\varphi} \quad X \tag{2.12}
\]

The full set of rules of $S$ is listed in Figure 2.6.

**Definition 2.25** A set $\Gamma$ is inconsistent if $\Gamma \vdash \varphi$ for all $\varphi$. Otherwise, $\Gamma$ is consistent.

**Theorem 2.26** The logic in Figure 2.6 is complete for $S$.

**Proof** Let $\Gamma$ be a set of sentences. Suppose that $\Gamma \models \varphi$. We show that $\Gamma \vdash \varphi$.

We may assume that $\Gamma$ is consistent, or else our result is trivial.

Divide $\Gamma$ into three parts in the obvious way:

\[
\Gamma = \Gamma_{\text{all}} \cup \Gamma_{\text{some}} \cup \Gamma_{\text{no}}.
\]

The proof now splits into two cases.

**Case I** $\varphi$ is a Some-sentence. Let $M$ be from Lemma 2.21 for $\Gamma_{\text{all}} \cup \Gamma_{\text{some}}$.

Subcase 1 $M \models \Gamma_{\text{no}}$. Then by hypothesis, $M \models \varphi$. Then Lemma 2.17 shows that $\Gamma \models S$, as desired.

Subcase 2 There is some No $p$ are $q$ in $\Gamma_{\text{no}}$ such that $[p] \cap [q] \neq \emptyset$. And again, Lemma 2.17 shows that $\Gamma_{\text{all}} \cup \Gamma_{\text{some}} \vdash \text{Some } p \text{ are } q$. Using (X), $\Gamma$ is inconsistent.

**Case II** $\varphi$ is a sentence in All or No. Let $M$ be from Lemma 2.23 for $\Gamma_{\text{all}} \cup \Gamma_{\text{no}}$.

---

4Please do not confuse this with reductio ad absurdum. See page 53 for more on this.
Subcase 1 \( M \models \Gamma_{\text{some}} \). Then by hypothesis \( M \models \varphi \). By Lemma 2.23, \( \Gamma \vdash \varphi \).

Subcase 2 there is some sentence \( \text{Some } p \text{ are } q \) in \( \Gamma_{\text{some}} \) such that \( M \not\models \text{Some } p \text{ are } q \). But then \( M \models \text{No } p \text{ are } q \). By Lemma 2.23, \( \Gamma \vdash \text{No } p \text{ are } q \). So again using (X), we see that \( \Gamma \) is inconsistent.

2.5. Additional Exercises

In these exercises, we are concerned with the full proof system.

EXERCISE 10 Prove that if \( \Gamma \vdash \varphi \), then there are infinitely many proof trees which establish that \( \Gamma \vdash \varphi \).

EXERCISE 11 Let \( T \) be a proof tree over \( \Gamma \) with more than one node. Prove that either \( T \) is a proof tree over \( \emptyset \), or else every atom and name which occurs in \( T \) also occurs in some sentence in \( \Gamma \).

EXERCISE 12 Show that \( \Gamma \) has a model iff \( \Gamma \) is consistent.

EXERCISE 13 Let \( \Gamma \cup \{ \text{Some } p \text{ are } q \} \) be inconsistent. Show that \( \Gamma \vdash \text{No } p \text{ are } q \). Similarly, let \( \Gamma \cup \{ \text{No } p \text{ are } q \} \) be inconsistent. Show that \( \Gamma \vdash \text{Some } p \text{ are } q \). [Here it is important to give a proof-theoretic argument. Prove the first part by induction on the length of the shortest path in a proof tree from a leaf labeled \( \text{Some } p \text{ are } q \) to a node labeled with an instance of the rule in (2.12).]

EXERCISE 14 Let \( \Gamma \) be a consistent set. Using Exercise 13, prove that there is a consistent \( \Gamma' \supseteq \Gamma \) such that for all \( p \) and \( q \), \( \Gamma' \) contains either \( \text{Some } p \text{ are } q \) or \( \text{No } p \text{ are } q \), but \( \Gamma' \) does not contain more sentences than \( \Gamma \) in \( \text{All or names} \). We call such a \( \Gamma' \) a strong extension of \( \Gamma \). [You may work with the case of \( \Gamma \) a finite set (even though the result holds in general), since the details in the finite case contain all the real work of the general case.]

The rest of the exercises outline a different proof of Theorem 2.26. It is important not to use completeness in working those exercises.

EXERCISE 15 Let \( \Gamma \) be a consistent set of sentences, and write \( \Gamma = \Gamma_{\text{All}} \cup \Gamma_{\text{Some}} \cup \Gamma_{\text{No}} \cup \Gamma_{\text{names}} \). Let \( \varphi \) be a sentence in \( \text{All or No} \). Assume that \( \Gamma \models \varphi \), and prove semantically that \( \Gamma_{\text{All}} \cup \Gamma_{\text{No}} \models \varphi \) as well. [That is, take \( M \models \Gamma_{\text{All}} \cup \Gamma_{\text{No}} \). Find a model \( M^+ \) so that \( M \) is a submodel of \( M^+ \), and \( M^+ \models \Gamma \). Our assumption on \( \Gamma \) tells us that \( M^+ \models \varphi \). And the submodel condition implies that \( M \models \varphi \).]

EXERCISE 16 Let \( \Gamma \) be consistent. Let \( M \) be the model from the proof of Lemma 2.21 for \( \Gamma_{\text{All}} \cup \Gamma_{\text{Some}} \cup \Gamma_{\text{names}} \). Show that \( M \models \Gamma \).

It follows from this fact that every consistent set \( \Gamma \) has a model.

EXERCISE 17 Use Exercises 13, 15, and 16 to give a different proof of the Completeness Theorem 2.26. [You’ll need to show that if \( \Gamma \) is consistent and \( \Gamma \models \varphi \), then \( \Gamma \vdash \varphi \). For this, we again need a split into cases. The cases of \( \text{All and No} \) use Theorem 2.24.]

EXERCISE 18 The classical syllogisms also considered sentences \( \text{Some } p \text{ is not } a \text{ q} \). In our setting, it makes sense also to add other sentences with negative verb phrases: \( J \text{ is not } a \text{ p} \), and \( J \text{ is not } M \). Give some sound proof rules for these sentences (on top of the system we already have).
2. BASIC SYLLOGISTIC PROOF SYSTEMS

Exercise 19 Adding your rules to those in Figure 2.6, prove the completeness of your system. [It will probably be easiest to use the method of Exercise 17. Having the extra sentences around adds a balance to the system and often makes it easier to prove theoretical properties like completeness, despite the additional cases that come from a bigger syntax.]

Exercise 20 Consider the language \( \mathcal{L} \) with \( \text{All } p \text{ are } q \), \( \text{Some } p \text{ are } q \), \( \text{No } p \text{ are } q \), and \( \text{Some } p \text{ are not } q \). But

\[ \varphi = \text{All } p \text{ which are } q \text{ are } n \]

is not part of the language \( \mathcal{L} \). The problem here is to show that \( \varphi \) cannot be expressed \( \mathcal{L} \), not even by a set of sentences. That is, there is no set \( \Gamma \) of sentences in \( \mathcal{L} \) such that for all \( M \), \( M \models \Gamma \) iff \( M \models R \). Here is an outline of the proof.

1. Consider the model \( M \) with universe \( \{x, y, a\} \) with \([p] = \{x, a\}\), \([q] = \{y, a\}\), \([n] = \{a\}\), and also \([u] = \emptyset\) for other atoms \( u \). Consider also a model \( N \) with universe \( \{x, y, a, b\} \) with \([p] = \{x, a, b\}\), \([q] = \{y, a, b\}\), \([n] = \{a\}\), and the rest of the structure the same as in \( M \). Show that for all sentences \( \psi \) in \( \mathcal{L} \), \( M \models \psi \) iff \( N \models \psi \).

2. Suppose towards a contradiction that we could express \( \varphi \), say by the set \( \Gamma \). Then since \( M \) and \( N \) agree on all sentences of \( \mathcal{L} \), they agree on \( \Gamma \). But \( M \models R \) and \( N \not\models R \), a contradiction.

Exercise 21 As a continuation of Exercise 20, show that in \( \mathcal{L} \) we cannot express \( \text{No } p \text{ which are } q \text{ are } n \).

Exercise 22 For any sentence \( S \), let \( S[J/M] \) be the same as \( S \) except that all \( J \)'s are replaced by \( M \)'s. Suppose that the set \( \Gamma \) has the property that if \( S \in \Gamma \), then \( S[J/M] \in \Gamma \). Show that for all \( S \), if \( \Gamma \models S \), then \( \Gamma \models S[J/M] \).

2.5.1. Exercises on a new construct: \( \text{All } x \text{ which are } y \text{ are } z \). We conclude this chapter with a few exercises on a logic for sentences of the form

\[ \text{All } x \text{ which are } y \text{ are } z. \]

To save space, we abbreviate this by \((x, y, z)\). We take this sentence to be true in a given model \( M \) if \([x] \cap [y] \subseteq [z] \). Note that \( \text{All } x \text{ are } x \) is semantically equivalent to \((x, x, y)\).

Exercise 23 Prove that the logic of \( \text{All } x \text{ which are } y \text{ are } z \) in Figure 2.5 is complete. [Hint: you can do this with a one-point model, just as in the completeness result for \( \mathcal{L}(\text{all}) \), Theorem 2.12.]

Exercise 24 Let \( \Gamma \) be a set of \((x, y, z)\) sentences.
(1) Show that if \( \Gamma \models (x, y, z) \), then \( \Gamma \models (y, x, z) \).
(2) Show that if \( \Gamma \vdash (x, y, z) \), then \( \Gamma \vdash (y, x, z) \).
(3) Suppose that we remove the axiom \((x, y, y)\), and in its place take the symmetry rule
\[
\begin{array}{c}
(y, x, z) \\
(x, y, z)
\end{array}
\]
Show that the new system is complete.

**Exercise 25** For each sentence \( \varphi = \text{All } p \text{ are } q \), let \( \varphi^* = (x, x, y) \). If \( \Gamma \) is a set of sentences of the first fragment, let \( \Gamma^* = \{ \varphi^* : \varphi \in \Gamma \} \). It is easy to check that if \( \Gamma \vdash \varphi \), then \( \Gamma^* \vdash \varphi^* \). Prove the converse. We say that the system of this section is a **conservative extension** of the system for All.

Sources for this chapter. The material on syllogistic fragments originates with Moss [18].
<table>
<thead>
<tr>
<th>All ( p ) are ( p )</th>
<th>( \text{All } p \text{ are } q )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{All } p \text{ are } q )</td>
<td>( \text{All } n \text{ are } q )</td>
</tr>
<tr>
<td>( \text{Some } p \text{ are } q )</td>
<td>( \text{Some } q \text{ are } p )</td>
</tr>
<tr>
<td>( \text{Some } q \text{ are } p )</td>
<td>( \text{Some } p \text{ are } q )</td>
</tr>
<tr>
<td>( \text{All } q \text{ are } n )</td>
<td>( \text{All } p \text{ are } q )</td>
</tr>
<tr>
<td>( \text{Some } p \text{ are } q )</td>
<td>( \text{Some } p \text{ are } q )</td>
</tr>
<tr>
<td>( \text{All } q \text{ are } n )</td>
<td>( \text{All } p \text{ are } n )</td>
</tr>
<tr>
<td>( \text{No } p \text{ are } q )</td>
<td>( \text{No } q \text{ are } p )</td>
</tr>
<tr>
<td>( \text{No } q \text{ are } p )</td>
<td>( \text{No } q \text{ are } p )</td>
</tr>
<tr>
<td>( \text{No } p \text{ are } p )</td>
<td>( \text{No } q \text{ are } p )</td>
</tr>
<tr>
<td>( \text{No } p \text{ are } q )</td>
<td>( \text{No } p \text{ are } q )</td>
</tr>
<tr>
<td>( \text{All } p \text{ are } q )</td>
<td>( \varphi )</td>
</tr>
</tbody>
</table>

**Figure 2.6.** The full set of rules of 5.
This chapter presents a logic for statements of the form *All* $p$ *are* $q$ and *Some* $p$ *are* $q$. As in Chapter 2, $p$ and $q$ are intended as (plural) nouns or other expressions whose natural denotation is as subsets of an underlying universe. The novelty here is to add an *explicit complement operator* to the syntax. So we now can say, for example, *All* $p'$ *are* $q$, or *Some* $p'$ *are* $q$. The point of the chapter is to present a sound and complete proof system for the associated entailment relation.

The methods that we have seen in Chapter 2 do not work, and so this chapter goes in a different direction, building models using a representation theorem coming from quantum logic. Before we turn to our logic, we take a detour in Section 3.1 to remind you about the completeness of classical sentential logic. That section is a very quick presentation, and so it is not a terribly good way to learn about sentential logic. But its aim is to expose the connection between *representation theorems* in algebra and *completeness theorems* in logic. So to showcase this connection, we eliminate everything that is not related to it.

### 3.1. Background: Completeness of Basic Sentential Logics

This section presents a short review of sentential logic\(^1\), aiming mainly at presenting in outline form some arguments for completeness. We do this partly because we need the results later on, and partly because we develop variations of those arguments.

There are many formulations of sentential logic, each with advantages and disadvantages. To settle matters, we fix a presentation. We start with a collection $\text{AtSen}$ of *atomic sentences* $p$, $q$, ...; sometimes we use subscripts for these\(^2\). This collection $\text{AtSen}$ may be *any set*, and it will be important in the future to take various sets as $\text{AtSen}$. We form *sentences over* $\text{AtSen}$ using the symbols $\neg$, $\land$, and $\lor$. We use Greek letters such as $\varphi$, $\psi$, and $\chi$ to denote sentences. So our syntax generates sentences such as $p \land \neg(p \lor q)$, $\neg
eg p$, etc. The set of sentences generated from $\text{AtSen}$ will be called $\text{Sen}$.

The semantics of sentential logic is given in terms of *valuations*. These are functions $v : \text{AtSen} \to \{T, F\}$. Each $v$ extends to a function $\tau : \text{Sen} \to \{T, F\}$ using

---

\(^1\)Sentential logic is also called *propositional logic*.

\(^2\)Note the difference in font between atomic sentences of sentential logic $p$, $q$, ... and unary atoms of syllogistic languages $p$, $q$, ...
3. SYLLOGISTIC LOGIC WITH COMPLEMENTS: \( s^t \)

the truth tables below.

<table>
<thead>
<tr>
<th>( \varphi )</th>
<th>( \neg \varphi )</th>
<th>( \psi )</th>
<th>( \varphi \land \psi )</th>
<th>( \varphi \lor \psi )</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

We use \( \varphi \rightarrow \psi \) as an abbreviation for \((\neg \varphi) \lor \psi\), and \( \varphi \leftrightarrow \psi \) for \((\varphi \rightarrow \psi) \land (\psi \rightarrow \varphi)\).

Suppose that \( v(p) = T \) and \( v(q) = F \). As an example of the calculations, we verify in detail that \( v((p \land q) \rightarrow q) = T \):

\[
\begin{align*}
\pi((p \land q) \rightarrow q) &= \pi(\neg(p \land q)) \lor \pi(q) \\
&= \neg(\pi(p) \land \pi(q)) \rightarrow F \\
&= \neg(T \land F) \lor F \\
&= \neg F \lor F \\
&= T
\end{align*}
\]

We say that \( v \) satisfies \( \varphi \) if \( v(\varphi) = T \). We also say that \( v \) makes \( \varphi \) true. If \( \Gamma \subseteq \text{Sen} \) and \( \varphi \in \text{Sen} \), we write \( \Gamma \models \varphi \) to mean that every model \( v \) which satisfies all sentences in \( \Gamma \) also satisfies \( \varphi \). If \( \Gamma \) is the empty set, then we drop it from the notation. And to say that \( \models \varphi \) means that every valuation makes \( \varphi \) true; in this case we call \( \varphi \) a tautology or a valid sentence (of sentential logic). For example, \( p \lor (\neg p) \) is a tautology.

**Lemma 3.1** The following sentences are tautologies of sentential logic.

1. \( p \rightarrow p \).
2. \( q \rightarrow (p \rightarrow q) \).
3. \( (p \rightarrow r) \rightarrow ((p \rightarrow (r \rightarrow q)) \rightarrow (p \rightarrow q)) \).
4. \( (p \rightarrow q) \rightarrow (\neg p \rightarrow q) \rightarrow q) \).

Moreover, every substitution instance of a tautology is again a tautology.

We should explain the point at the end about substitution instances. It means that if we take one of our tautologies, say \( q \rightarrow (p \rightarrow q) \), and replace \( p \) by any sentence (say \( \neg q \), just for an example) and also replace \( q \) by any sentence (say \( p \land q \)), then we get another tautology:

\[
(p \land q) \rightarrow (\neg q \rightarrow (p \land q))
\]

We are going to mention a proof system shortly, and this will provide a notion \( \Gamma \vdash \varphi \) that is intended to match the semantic notion \( \Gamma \models \varphi \) which we have just seen. But before we do this, it will be good to make an important distinction.

**Definition 3.2** Let \( \text{AtSen} \) be a set of atomic sentences, Let \( \mathcal{C} \) be a class of valuations, and let \( \vdash \) be a proof system for \( \text{Sen} \).

1. \( \Gamma \models \varphi \) on \( \mathcal{C} \) means that if \( v \in \mathcal{C} \) and \( \pi(\varphi) = T \) for all \( \psi \in \mathcal{C} \), then also \( \pi(\varphi) = T \).
2. \( \vdash \) is weakly complete for \( \mathcal{C} \) iff whenever \( \models \varphi \) on \( \mathcal{C} \), then also \( \vdash \varphi \). In other words, every sentence which is valid on \( \mathcal{C} \) is provable.
3. A proof system \( \vdash \) is (strongly) complete for \( \mathcal{C} \) iff whenever \( \Gamma \models \varphi \) on \( \mathcal{C} \), then also \( \Gamma \vdash \varphi \).

Unless otherwise mentioned, complete means strongly complete.
3.1. BACKGROUND: COMPLETENESS OF BASIC SENTENTIAL LOGICS

Figure 3.1. The smallest sentential logical system $\mathcal{SL}$. It turns out that this system $\mathcal{SL}$ is strongly complete for sentential logic. We shall also be interested in logical systems which extend this system by adding additional axioms which are sound for various classes $\mathcal{C}$ of models.

**Definition 3.3** A sentential logical system is any proof system $\vdash$ for $\text{Sen}$ extending the system in Figure 3.1. That is, the axioms of $\vdash$ must include the tautologies of sentential logic, and the only rule of the system is modus ponens.

**Theorem 3.4** (Deduction Theorem) Let $\vdash$ be a sentential logical system. Suppose that $\Gamma \cup \{\varphi\} \vdash \psi$. Then $\Gamma \vdash \varphi \rightarrow \psi$.

**Proof** By induction on the height of the derivation showing $\Gamma \cup \{\varphi\} \vdash \psi$. If the height is 1, then either $\psi$ is an axiom of the system, $\psi \in \Gamma$, or $\psi$ is $\varphi$. If $\varphi$ is an axiom or if $\psi \in \Gamma$, then we use the tautology $\psi \rightarrow (\varphi \rightarrow \psi)$ in Lemma 3.1 part 2. That is, we use the proof tree

$$
\begin{align*}
\Gamma & \vdash \psi \\
\Gamma & \vdash \psi \rightarrow (\varphi \rightarrow \psi) \\
\Gamma & \vdash \varphi \rightarrow \psi
\end{align*}
$$

If $\psi$ is $\varphi$, then note that $\varphi \rightarrow \varphi$ is a tautology.

Assume that $\Gamma \cup \{\varphi\} \vdash \psi$ with a derivation of height $n + 1$, and suppose that the last step concludes $\psi$ using modus ponens from $\chi$ and $\chi \rightarrow \psi$. Then $\Gamma \cup \{\varphi\} \vdash \chi$ with a proof tree of height $\leq n$, and also $\Gamma \cup \{\varphi\} \vdash \chi \rightarrow \psi$ with a proof tree of height $\leq n$. By induction hypothesis,

$$
\Gamma \vdash \varphi \rightarrow \chi \quad \text{and} \quad \Gamma \vdash \varphi \rightarrow (\chi \rightarrow \psi).
$$

Then we use Lemma 3.1, part 3. □

**Definition 3.5** In our statement of Lemma 3.6, we need some general definitions. For any set $\Gamma$ in any sentential logical system $\text{Sen}$, let

| $\Gamma_{at}$ | $\Gamma \cap \text{AtSen}$ |
| $\neg \Gamma_{at}$ | $\{\neg \varphi : \varphi \in \Gamma_{at}\}$ |
| $\Gamma_{+at}$ | $\Gamma_{at} \cup \neg \Gamma_{at}$ |

**Lemma 3.6** Let $\text{AtSen}$ be a set of atomic sentences, Let $\vdash$ be a sentential logical system for $\text{Sen}$, and let $\Gamma$ be maximal $\vdash$-consistent. Let $v : \text{AtSen} \rightarrow \{T, F\}$ be any valuation. If $v(\Gamma_{+at}) = T$, then $v(\Gamma) = T$ also.

**Proof** We show that for all $\varphi$, $v(\varphi) = T$ iff $\varphi \in \Gamma$. By induction on $\varphi$. The base of the induction is when $\varphi$ is an atomic sentence. Then $\varphi \in \Gamma_{+at}$, so $v(\varphi) = T$.

And if $v(\varphi) = T$, we claim that $\varphi \in \Gamma_{+at}$. For if not, then $\varphi$ is a negation, say $\neg \psi$, and $\psi \in \Gamma_{-at}$. But then $v(\psi) = T$, so $v(\varphi) = F$. This is a contradiction.
24 3. SYLLOGISTIC LOGIC WITH COMPLEMENTS: $S^t$

Now assume our statement for $\varphi$; we prove it for $\neg \varphi$. The key point is that $\Gamma$ is maximal $\vdash$-consistent. So we have

\[
\neg \varphi \in \Gamma \quad \text{iff} \quad \varphi \notin \Gamma \quad \text{by maximality}
\]

\[
\text{iff} \quad \pi(\varphi) \neq T \quad \text{by induction hypothesis}
\]

\[
\text{iff} \quad \pi(\varphi) = F
\]

\[
\text{iff} \quad \pi(\neg \varphi) = T
\]

The induction steps for $\land$ and $\lor$ are similar, and they again use maximal $\vdash$-consistency and the induction hypothesis. \hfill $\Box$

### 3.1.1. Boolean algebras.

**Definition 3.7** A boolean algebra is a tuple

\[ B = (B, 1, 0, \neg, \land, \lor) \]

such that $B$ is a set; $1, 0 \in B$: $\neg : B \to B$; and $\land, \lor : B \times B \to B$ such the laws in Figure 3.2 hold.

There are a number of abbreviations that are adopted in the study of boolean algebras. We write $x \leq y$ for $x \land y = x$. (This is equivalent to writing $x \lor y = y$.) We also write $x \rightarrow y$ for $\neg x \lor y$, and we write $x \leftrightarrow y$ for $(x \rightarrow y) \land (y \rightarrow x)$.

**Example 3.8** For us, there are three main examples of boolean algebras. Two will be presented here, and the last in Example 3.9.

First, $\{T, F\}$ with the truth table operations is a boolean algebra. We take $1$ to be $T$ and $0$ to be $F$.

Second, for every set $X$, the power set of $X$ is a boolean algebra:

\[ \mathcal{P}(X) = (\mathcal{P}(X), X, 0, \land, \lor') \]

Here $\mathcal{P}(X)$ is the set of all subsets of $X$. Two examples of subsets of $X$ are $X$ itself and $\emptyset$, and these play the roles of $1$ and $0$. The $\land$ and $\lor$ of $\mathcal{P}(X)$ are intersection and union, respectively, and the complement operation is given by $A' = X \setminus A$. (This is the complement of $A$ relative to $X$.)

**Example 3.9** We shall be interested in a class of boolean algebras derived from sentential logics and sets of sentences. Let $\vdash$ be a sentential logic, and let

| $x \land x$ | $x$ | $x \lor x$ | $x$ |
| $x \land y$ | $y \land x$ | $x \lor y$ | $y \lor x$ |
| $x \land (y \land z)$ | $(x \land y) \land z$ | $x \lor (y \lor z)$ | $(x \lor y) \lor z$ |
| $x \land (x \lor y)$ | $x \lor (x \land y)$ | $x$ |
| $x \land (y \lor z)$ | $(x \land y) \lor (x \land z)$ | $x \lor (y \land z)$ | $(x \lor y) \land (x \lor z)$ |
| $\neg (x \land y)$ | $(\neg x) \land (\neg y)$ | $\neg (x \lor y)$ | $(\neg x) \land (\neg y)$ |
| $1 \land x$ | $x$ | $1 \lor x$ | $1$ |
| $0 \land x$ | $0$ | $0 \lor x$ | $x$ |
| $x \land \neg x$ | $0$ | $x \lor \neg x$ | $1$ |

Figure 3.2. The laws of boolean algebra.
Γ ⊆ Sen. We define the syntactic boolean algebra \( B_\Gamma \) as follows: let \( \equiv \) be the relation on \( Sen \) defined by

\[ \varphi \equiv \psi \quad \text{iff} \quad \Gamma \vdash \varphi \leftrightarrow \psi. \]

Then \( \equiv \) is an equivalence relation, and indeed it is a congruence. This means that if \( \varphi \equiv \varphi' \) and \( \psi \equiv \psi' \), then \( \varphi \wedge \psi \equiv \varphi' \wedge \psi' \); and similarly for \( \neg \) and \( \vee \). The quotient set \( Sen/\equiv \) is then a boolean algebra, where we define the operations in the natural way: for example, \( \neg [\varphi] = [\neg \varphi] \). We also take 1 to be \([p \lor \neg p]\) and 0 to be \([p \land \neg p]\). This boolean algebra is denoted \( B_\Gamma \). It is sometimes called the Lindenbaum-Tarski algebra of \( \Gamma \).

**Definition 3.10** Let \( B \) be a boolean algebra. A set \( F \subseteq B \) with (1)–(3) is called a filter on \( B \).

1. If \( x \in U \) and \( x \leq y \), then \( y \in U \).
2. If \( x, y \in U \), then \( x \land y \in U \).
3. 1 ∈ U, and 0 /∈ U.

An ultrafilter on \( B \) is a filter with the additional property that

4. For all \( x \in B \), either \( x \in U \) or \( \neg x \in U \).

**Example 3.11** Let \( B \) be \( \mathcal{P}(\{1, 2, 3\}) \), the power set of a three-element set, considered as a boolean algebra (see Example 3.8). One example of a filter would be

\[ \{\{1, 2\}, \{1, 2, 3\}\} \]

This is not an ultrafilter because, for example, it contains neither \( \{1\} \) nor its complement \( \{2, 3\} \). An example of an ultrafilter, consider \( \{\{2\}, \{1, 2\}, \{1, 2, 3\}\} \). (This is the set of supersets of \( \{2\} \). Let us write this as \( 2^\uparrow \). In general, if \( X \) is any set the principal ultrafilters on the boolean algebra \( \mathcal{P}(X) \) are the sets of the form \( x^\uparrow \) for some \( x \in X \).)

**Proposition 3.12** Let \( B \) be a boolean algebra, and let \( F \subseteq B \) be a filter. Then there is some ultrafilter \( U \) of \( B \) such that \( F \subseteq U \).

**Proof** We use Zorn’s Lemma. Let

\[ \mathbb{P}_F = \{ E \subseteq B : F \subseteq E \text{ and } E \text{ is a filter} \}. \]

\( \mathbb{P}_F \) is a partially ordered set under the inclusion relation, and it is also closed under unions of chains. (That is, the union of any set of filters is a filter.) Thus Zorn’s Lemma applies, and we obtain an element of \( \mathbb{P}_F \) which is maximal with respect to inclusion. Let \( U \) be such an element. Then \( U \) is a filter and \( F \subseteq U \). We claim that \( U \) is indeed an ultrafilter. For suppose not. Then there is some \( x \in B \) such that neither \( x \) nor \( \neg x \) belongs to \( U \). It follows that \( U \cup \{x\} \) is not a subset of any filter, and similarly for \( U \cup \{\neg x\} \). This point about \( U \cup \{x\} \) implies that there is a finite set \( y_1, \ldots, y_m \in U \) such that

\[ y_1 \land \cdots \land y_m \land x = 0. \]

(For if not, then \( \{b \in B : \text{ for some } y_1, \ldots, y_m \in U, \land y_i \land x \leq b\} \) would be a filter strictly larger than \( U \), contradicting maximality.) Similarly, there is a finite \( z_1, \ldots, z_n \in U \) such that

\[ z_1 \land \cdots \land z_n \land (\neg x) = 0. \]
And now, note that
\[
\begin{align*}
y_1 \land \cdots \land y_m \land z_1 \land \cdots \land z_n \\
= (\land y_1 \land \land z_j) \land (x \lor \neg x) \\
= (\land y_1 \land \land z_j \land x) \lor (\land y_i \land \land z_j \land \neg x) \\
= 0 \land 0 \\
= 0
\end{align*}
\]
But \(\land y_i \land \land z_j\) belongs to \(U\), so by the filter property, \(0 \in U\) also. And this contradicts \(U\) being a filter.

\[\square\]

### 3.1.2. On the completeness of sentential logics.

**Lemma 3.13** Let \(\vdash\) be a sentential logic, let \(\Gamma \subseteq \text{Sen}\), and let
\[
\Gamma = \{[\varphi] : \Gamma \vdash \varphi\}.
\]

1. \(\Gamma\) is \(\vdash\)-consistent iff \(\Gamma\) is a filter on \(B\Gamma\).
2. \(\Gamma\) is maximal \(\vdash\)-consistent in the logic iff \(\Gamma\) is an ultrafilter on \(B\Gamma\).

**Proof** Suppose first that \(\Gamma\) is \(\vdash\)-consistent. If \([\varphi]\) and \([\psi]\) belong to \(\Gamma\), then \(\Gamma \vdash \varphi\) and \(\Gamma \vdash \psi\). Using the tautology \(\varphi \rightarrow (\psi \rightarrow (\varphi \land \psi))\), we see that \(\Gamma \vdash \varphi \land \psi\).

And if \(\Gamma \vdash \varphi\) and \(\models \varphi \rightarrow \psi\), then \(\Gamma \vdash \psi\). This time, the tautology is \(\varphi \rightarrow ((\varphi \rightarrow \psi) \rightarrow \psi)\). It is immediate that \([F] \notin \Gamma\). Therefore \(\Gamma\) is a filter.

In the other direction, assume that \(\Gamma\) is a filter. Since \([F] \notin \Gamma\), \(\Gamma\) is \(\vdash\)-consistent.

If \(\Gamma\) is maximal \(\vdash\)-consistent, then for every sentence \(\varphi\), either \(\varphi \in \Gamma\) or \(\neg \varphi \in \Gamma\).

If follows that for all \(\varphi\), either \([\varphi] \in \Gamma\) or \([\neg \varphi] \in \Gamma\). So \(\Gamma\) is an ultrafilter. The converse is similar.

\[\square\]

**Proposition 3.14** Let \(\vdash\) be a sentential logic, and let \(\Gamma\) be a set of sentences in \(\text{Sen}\). If \(\Gamma\) is \(\vdash\)-consistent, then there is some maximal \(\vdash\)-consistent \(\Delta \supseteq \Gamma\).

**Proof** By Lemma 3.13 and Proposition 3.12.

**Lemma 3.15** Let \(\vdash\) be a sentential logic, let \(\Gamma\) be a set of sentences in \(\text{Sen}\), and let \(\mathcal{C}\) be a class of valuations. The following are equivalent assertions:

1. \(\vdash\) is complete for \(\mathcal{C}\): If \(\Gamma \models \varphi\) on \(\mathcal{C}\), then \(\Gamma \vdash \varphi\).
2. Every \(\vdash\)-consistent set \(\Gamma\) has a model in \(\mathcal{C}\).
3. Every maximal \(\vdash\)-consistent set \(\Gamma\) has a model in \(\mathcal{C}\).

**Proof** (1)\(\Rightarrow\)(2): Let \(\Gamma\) be \(\vdash\)-consistent. If \(\Gamma\) has no models in \(\mathcal{C}\), then \(\Gamma \models \varphi\) on \(\mathcal{C}\) and \(\Gamma \models \neg \varphi\) on \(\mathcal{C}\), for any \(\varphi\). By completeness, \(\Gamma \vdash \varphi\), and \(\Gamma \vdash \neg \varphi\). So \(\Gamma\) is \(\vdash\)-inconsistent, a contradiction.

(2)\(\Rightarrow\)(3) is trivial.

The important direction for us is (3)\(\Rightarrow\)(1). Suppose that every on \(\vdash\)-consistent set has a model in \(\mathcal{C}\), and suppose that \(\Gamma \models \varphi\). If \(\Gamma \nvdash \varphi\), then \(\Gamma \cup \{\neg \varphi\}\) must be \(\vdash\)-consistent. (For if \(\Gamma \cup \{\neg \varphi\} \vdash \bot\), then by the Deduction Theorem, \(\Gamma \vdash \neg \varphi \rightarrow \bot\).)

Now we have a tautology \((\neg \varphi \rightarrow \bot) \rightarrow \varphi\). And so \(\Gamma \vdash \varphi\) after all. Let \(\Delta \supseteq \Gamma\) be maximal \(\vdash\)-consistent; \(\Delta\) exists by Proposition 3.14. By hypothesis, we have some \(M \models \Delta\) with \(M \in \mathcal{C}\). A fortiori, \(M \models \Gamma\).

\[\square\]

**Theorem 3.16** Let \(\text{AtSen}\) be any set of atomic sentences. Let \(\mathcal{C}\) be the class of all valuations \(v : \text{AtSen} \rightarrow \{\top, \bot\}\). The proof system in Figure 3.1 is sound and complete for \(\mathcal{C}\): \(\Gamma \models \varphi\) iff \(\Gamma \vdash \varphi\).
3.2. ADDING SENTENTIAL BOOLEAN OPERATIONS TO $\mathcal{L}(\text{all, some})$

**Proof** The soundness is an easy induction. Lemma 3.15 reduces the completeness to the verification that every $\Gamma$ which is maximal consistent in the logic is satisfied by some valuation $v$. Let $v(\varphi)$ be defined for atomic $\varphi$ by

$$v(\varphi) = T \quad \text{iff} \quad \varphi \in \Gamma.$$ 

Recall the set $\Gamma_{+\text{at}}$ from Definition 3.5. We claim that for all $\varphi \in \Gamma_{+\text{at}}$, $v(\varphi) = T$. For $\varphi \in \text{AtSen}$, this is immediate. And if $\varphi = \neg \psi$ for some $\psi \in \Gamma_{+\text{at}}$, then we know that $\psi \not\in \Gamma$ (lest $\Gamma$ be inconsistent). So $v(\psi) = F$; hence $v(\varphi) = \neg v(\psi) = \neg F = T$.

By this claim and Lemma 3.6, $v(\Gamma) = T$. This completes the proof. □

3.1.3. Why bother with representation theorems? After going through this exercise, you might wonder why anyone would bother proving representation theorems if all they are interested in is the completeness theorem that comes as a corollary. The reason is that formulating logical matters in algebraic terms makes it easier to see solution to the mathematical problems underlying completeness.

3.2. Adding Sentential Boolean Operations to $\mathcal{L}(\text{all, some})$

At this point, we combine what we did in Chapter 2 with our work in Section 3.1 just above. We let

$$(3.1) \quad \mathcal{L}(\text{all, some})_{bc} = \text{SentLogic}(\mathcal{L}(\text{all, some}))$$

That is, we study sentential logic where the atomic sentences are the sentences of $\mathcal{L}(\text{all, some})$. In addition to the sentences in $\mathcal{L}(\text{all, some})$, we also have sentential connectives $\neg$, $\land$, and $\lor$. But everything is built out of sentences in $\mathcal{L}(\text{all, some})$. We call this language $\mathcal{L}(\text{all, some})_{bc}$, with “bc” standing for “boolean connectives.”

We must settle on a class $C$ of valuations in order to state and prove a completeness result. The semantics of $\mathcal{L}(\text{all, some})$ is based on models $M = (M, \models)$. Every $M$ for $\mathcal{L}(\text{all, some})$ gives a valuation

$$v_M : \mathcal{L}(\text{all, some}) \to \{T, F\},$$

given by

$$v_M(\varphi) = \begin{cases} T & \text{if } M \models \varphi \\ F & \text{if } M \not\models \varphi \end{cases}$$

In the rest of this section, let $C = \{v_M : M \text{ is a model for the language } \mathcal{L}(\text{all, some})\}$.

Our goal is to craft a sentential logical system $\vdash$ and then to show that $\Gamma \models \varphi$ on $C$ if $\Gamma \vdash \varphi$.

**Remark** When we defined models and semantics in Chapter 2, we defined what it meant to say that $M \models \varphi$ for $\varphi$ in $\mathcal{L}(\text{all, some, no})$. We can easily extend this definition to cover the case when $\varphi$ is in the larger set $\mathcal{L}(\text{all, some, no})_{bc}$. To say that $\Gamma \models \varphi$ on the class $C$ is exactly the same thing as saying that for all models $M$, if $M \models \psi$ for all $\psi \in \Gamma$, then also $M \models \varphi$. In other words, one may reformulate the notion $\Gamma \models \varphi$ in a way that does not mention valuations. The two notions are equivalent.
We present our sentential logical system for $L(all, some)_{bc}$ in Figure 3.3. As with any sentential logical system, the system also includes the substitution instances of the tautologies as axioms, and the only rule is modus ponens. The soundness of this system is routine.

It should be mentioned that although we are going to prove that our logic is complete, this doesn’t mean that it is a very convenient system to work in. To be sure, constructing formal proofs in Hilbert-style systems with no shortcuts is a tedious matter, as the next example shows.

**Example 3.17** Let $\alpha, \ldots, \varepsilon$ be the sentences listed below.

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$\gamma$</th>
<th>$\delta$</th>
<th>$\varepsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>All $p$ are $u$</td>
<td>All $p$ are $v$</td>
<td>Some $u$ are $v$</td>
<td>Some $p$ are $p$</td>
<td>Some $p$ are $u$</td>
</tr>
</tbody>
</table>

We wish to show that $\{\alpha, \beta, \neg \gamma\} \vdash \neg \delta$. We’ll exhibit a proof tree for this, using some abbreviations. First, note that $(\alpha \land \delta) \rightarrow \varepsilon$ and $(\varepsilon \land \beta) \rightarrow \gamma$ are axioms of our logic. Second, define some additional sentences as follows:

<table>
<thead>
<tr>
<th>$\varphi_1$</th>
<th>$\varphi_2$</th>
<th>$\varphi_3$</th>
<th>$\varphi_4$</th>
<th>$\varphi_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\neg \gamma \rightarrow \neg \delta$</td>
<td>$\beta \rightarrow \alpha$</td>
<td>$\alpha \rightarrow \varphi_1$</td>
<td>$((\varepsilon \land \beta) \rightarrow \gamma) \rightarrow \varphi_3$</td>
<td>$((\alpha \land \delta) \rightarrow \varepsilon) \rightarrow \varphi_4$</td>
</tr>
</tbody>
</table>

One can check that $\varphi_5$ is a substitution instance of a tautology of sentential logic. Now we have a proof tree over $\Gamma$:

$$
\frac{\varphi_5}{\varphi_4} \quad \frac{\varphi_4}{\varphi_3} \quad \frac{\varphi_3}{\varphi_2} \quad \frac{\varphi_2}{\varphi_1} \quad \frac{\alpha \land \beta}{\beta \rightarrow \gamma} \quad \frac{\gamma}{\varepsilon} \quad \frac{\varepsilon \land \beta}{\gamma} \quad \frac{\gamma}{\delta} \quad \frac{\delta}{\neg \delta}$$

This shows that indeed $\{\alpha, \beta, \neg \gamma\} \vdash \neg \delta$.

Of course, in constructing this example one takes an informal proof and derives $\varphi_1, \ldots, \varphi_5$ from it. A friendlier system would allow one to directly use proof strategies such as hypothetical proofs and proofs by contradiction, without the detour through long sentential tautologies.
Theorem 3.18  Let $\vdash$ be the sentential logical system for $L(all, some)_{bc}$ defined in Figure 3.3, and let $\mathcal{C}$ be the class of valuations derived from models. Then the following completeness theorem holds: $\Gamma \models \varphi$ on $\mathcal{C}$ iff $\Gamma \vdash \varphi$.

Proof By Lemma 3.15, we only need to show that that every $\Gamma$ which is maximal $\vdash$-consistent has a model of the form $v_M$. This maximality implies that for every $\varphi$, either $\varphi \in \Gamma$ or else $\neg \varphi \in \Gamma$. And $\Gamma$ is closed under deduction in $L(all, some)_{bc}$.

We have seen the notation $\Gamma_{some}$ for the set of sentences in $\Gamma$ of the form Some $p$ are $q$. Let $\Gamma_{\neg all}$ be the set of $\neg (All$ $p$ $are$ $q)$ sentences in $\Gamma$.

Let $M$ be given by:

$$M = \Gamma_{some} \cup \Gamma_{\neg all}$$

$$[u] = \{ \varphi \in \Gamma_{some} : \varphi \text{ is Some } p \text{ are } q, \text{ and either } p \leq u \text{ or } q \leq u \}$$

$$= \bigcup \{ \varphi \in \Gamma_{\neg all} : \varphi \text{ is } \neg (All \ p \ are \ q) \text{ and } p \leq u \}$$

Notice that $M$ is the union of $\Gamma_{some}$ and $\Gamma_{\neg all}$. We assume that this union is disjoint; i.e., no sentence belongs to both sets.

We have also seen the notation $\Gamma_{\pm at}$ in Definition 3.5.

Claim For all $\varphi \in \Gamma_{\pm at}$, $M \models \varphi$.

Proof First, suppose that $\Gamma$ contains the sentence All $u$ are $v$. So $u \leq v$, and it is easy to check that $[u] \subseteq [v]$. So All $u$ are $v$ in $\Gamma$ is true in $M$.

Second, consider a sentence Some $u$ are $v$ in $\Gamma$. Call this sentence $\varphi$. Then $\varphi \in \Gamma_{some}$, and indeed $\varphi \in [u] \cap [v]$. Hence $[u] \cap [v] \neq \emptyset$, and so $\varphi$ is true in $M$.

Continuing, suppose that $\neg (All \ u \ are \ v)$ in $\Gamma$. Again, call this $\varphi$. This time, $\varphi \in \Gamma_{\neg all}$. Note that $\varphi \in [u]$. We claim that $\varphi \notin [v]$, and from this it follows that $\varphi$ is true in $M$. For if $\varphi \in [v]$, it must be the case that $u \leq v$. That is, $\Gamma \vdash All \ u \ are \ v$. Hence $\Gamma$ is $\vdash$-inconsistent, and this is a contradiction.

Finally, suppose that $\Gamma$ contains $\neg (Some \ u \ are \ v)$. Suppose towards a contradiction that this sentence were true in $M$. Thus we have some $\varphi \in [u] \cap [v]$. We have two cases depending on whether $\varphi \in \Gamma_{some}$ or $\varphi \in \Gamma_{\neg all}$. Suppose first that $\varphi \in \Gamma_{some}$, say $\varphi$ is Some $p$ are $q$. Note that $\Gamma \vdash \varphi$, since $\varphi \in \Gamma_{some}$. Then we have four subcases, and two of these are typical: first, $p \leq u$ and $q \leq v$; second, $p \leq u$ and $p \leq v$. In the first subcase, we easily get Some $u$ are $v$ from $\Gamma$; see the proof displayed in line (2.3) on page 11. In the second subcase, we again get Some $u$ are $v$ from $\Gamma$, but with a slightly different derivation. Either way, we have a contradiction in this case. The second case is when $\varphi \in \Gamma_{\neg all}$. In this case, write $\varphi$ as $\neg (All \ p \ are \ q)$. Then $p \leq u$ and $p \leq v$. Using what we did in Example 3.17, we get $\neg Some \ p \ are \ p$ from $\Gamma$. And so we also have All $p$ are $q$. So $\Gamma$ is $\vdash$-inconsistent, and this is a contradiction.

At this point, our claim is shown. By Lemma 3.6, $M \models \Gamma$.

This concludes our proof of Theorem 3.18.

Exercise 26 Where in the proof of Theorem 3.18 were axioms (2)–(6) in Figure 3.3 used? Our proof did not make this clear.
3. Syntax and Semantics of $S^\dagger$

We introduce a language $S^\dagger$ which is strictly bigger than $S$ in that it has noun-level negation rather than sentence-level negation.

In the syntax, we again begin with a set $\mathbf{P}$ of (unary) atoms. We use $p, q, \ldots$, for atoms. The idea once again is that unary atoms represent plural common nouns. Let $\text{Lit} = \mathbf{P} + \mathbf{P}$. (The $+$ here is the disjoint union, so we have two copies of $\mathbf{P}$.) We call the elements of this set literals following uses in areas of logic. We write the elements of $\text{Lit}$ as either atoms $p, q, \ldots$, or as complemented atoms $p', q', \ldots$.

Moreover, we extend this idea of complementation to a function complementation operation $\lnot : \text{Lit} \rightarrow \text{Lit}$ on the literals such that $p'' = p$ for all literals $p$. (Yes, we use the same letters $p, q, \ldots$ to range over literals in this chapter.) This involutive property implies that complementation is a bijection on $\text{Lit}$. Then we consider sentences $\text{All } p \text{ are } q$ and $\text{Some } p \text{ are } q$. Here $p$ and $q$ are any literals, including the case when they are the same. We call this language $S^\dagger$. We shall use letters like $\varphi$ to denote sentences.

Semantics. One starts with a set $M$ and a subset $[[p]] \subseteq M$ for each literal $p$, subject to the requirement that $[[p']] = M \setminus [[p]]$ for all $p$. This gives a model $M = (M, [\ [ ] ] )$. We then define the satisfaction relation $M |\models = \varphi$ just as in Chapter 2, and also derived notions such as $\Gamma |\models = \varphi$.

Example 3.19 We claim that $\Gamma |\models = \text{All } x \text{ are } z$, where

$\Gamma = \{ \text{All } y' \text{ are } p, \text{All } p \text{ are } q, \text{All } q \text{ are } y, \text{All } y \text{ are } p, \text{All } q \text{ are } z \}$.

Here is an informal explanation. Since all $y$ and all $y'$ are $p$, everything whatsoever is a $p$. And since all $p$ are $q$, and all $q$ are $y$, we see that everything is a $y$. In particular, all $x$ are $y$. But the last two premises and the fact that all $p$ are $q$ also imply that all $y$ are $z$. So all $x$ are $z$.

Exercise 27 Here is an example which we mention mostly as a challenge. Let $\Gamma$ be the set of the sentences below:

- $\text{All } y \text{ are } x$, $\text{All } y' \text{ are } x$, $\text{All } z' \text{ are } y$, $\text{All } z \text{ are } y'$, $\text{All } z \text{ are } w$
- $\text{Some } x \text{ are } x$, $\text{Some } x' \text{ are } x'$, $\text{Some } y \text{ are } y$, $\text{Some } y' \text{ are } y'$,
- $\text{Some } z \text{ are } z$, $\text{Some } z' \text{ are } z'$

It is not true that

$\Gamma |\models = \text{All } y \text{ are } w$.

Find a model $M |\models = \Gamma$ where $[[y]] \not\subseteq [[w]]$. We shall return to this problem later (Exercise 31), after we have the tools to solve this kind of problem in a general way.

No. In previous work, we took $\text{No } p \text{ are } q$ as a basic sentence in the syntax. There is no need to do this here: we may regard $\text{No } p \text{ are } q$ as a variant notation for $\text{All } p \text{ are } q'$. So the semantics would be

$M |\models = \text{No } p \text{ are } q \iff [[p]] \cap [[q]] = \emptyset$

In other words, if one wants to add $\text{No}$ as a basic sentence forming-operation, on a par with $\text{Some}$ and $\text{All}$, it would be easy to do so.
3.3. SYNTAX AND SEMANTICS OF $S^\dagger$

All $p$ are $p$

Axiom

Some $p$ are $q$
Some $p$ are $p$

Some$_1$

Some $q$ are $p$
Some $p$ are $q$

Some$_2$

All $p$ are $q$

Barbara

All $q$ are $n$
Some $p$ are $q$

Darii

All $p$ are $n$

All $n$ are $q$

All $p$ are $q$

All $q$ are $n$

All $p$ are $q$

Zero

All $q$ are $q'$

All $p$ are $q'$

Antitone

All $p$ are $q$

Some$_1$

Some $p$ are $q$

Some $q$ are $p$

Some$_2$

All $q$ are $q'$

One

All $p$ are $q$

All $q$ are $q'$

X

Figure 3.4. $S^\dagger$: syllogistic logic with complement.

Proof trees. We have discussed the meager syntax of $S^\dagger$ and its semantics. We next turn to the proof theory. Just as in Chapter 2, a proof tree over $\Gamma$ is a finite tree $T$ whose nodes are labeled with sentences in our fragment, with the additional property that each node is either an element of $\Gamma$ or comes from its parent(s) by an application of one of the rules for the fragment listed in Figure 3.4. $\Gamma \vdash \varphi$ means that there is a proof tree $T$ for over $\Gamma$ whose root is labeled $\varphi$.

We attached names to the rules in Figure 3.4 so that we can refer to them later. We usually do not display the names of rules in our proof trees except when to emphasize some point or other. The names “Barbara” and “Darii” are traditional from Aristotelian syllogisms. But the (Antitone) rule is not part of traditional syllogistic reasoning. It is possible to drop (Some$_2$) if one changes the conclusion of (Darii) to Some $n$ are $p$. But at one point it will be convenient to have (Some$_2$), and so this guides the formulation. The rules (Zero) and (One) are concerned with what is often called vacuous universal quantification. That is, if $q' \subseteq p$, then $q$ is the whole universe and $q'$ is empty; so $q$ is a superset of every set and $q'$ a subset. We have already seen the rule (X) rule: it permits inference of any sentence $\varphi$ whatsoever from a contradiction.

Example 3.20 Returning to Example 3.19, here is a proof tree showing $\Gamma \vdash$ All $x$ are $z$:

All $y'$ are $p$

All $y$ are $p$

All $y'$ are $y$

All $y$ are $y$

All $x$ are $y$

All $p$ are $q$

All $q$ are $y$

All $p$ are $y$

All $p$ are $z$

All $y$ are $p$

All $p$ are $z$

All $y$ are $z$

All $x$ are $z$

Exercise 28 Show that

$\{\text{All } y \text{ are } p, \text{All } y' \text{ are } p\} \vdash \text{All } x \text{ are } p$. 
Exercise 29  Show that
\{All y are p, All y' are p, All q are z, Some x are z'\} \vdash Some p are q'.

Lemma 3.21  The following are derivable:

1. Some p are p \vdash \varphi  (a contradiction fact)
2. All p are n, No n are q \vdash No q are p  (Celarent)
3. No p are q \vdash No q are p  (E-conversion)
4. Some p are q, No q are n' \vdash Some p are n'  (Ferio)
5. All q are n, All q are n' \vdash No q are q  (complement inconsistency)

Proof  For the assertion on contradictions,
\[
\begin{array}{c}
\text{All p are p} \\
\text{Axiom}
\end{array}
\quad
\begin{array}{c}
\text{Some p are p}' \\
\text{S}
\end{array}
\quad
X
\]
(Celarent) in this formulation is just a re-phrasing of (Barbara), using complements:
\[
\begin{array}{c}
\text{All p are n} \\
\text{All n are q'}
\end{array}
\quad
\begin{array}{c}
\text{All q are n'}
\end{array}
\quad
\text{Barbara}
\]
(E-conversion) is similarly related to (Antitone), and (Ferio) to (Darii). For complement inconsistency, use (Antitone) and (Barbara).
\[\square\]

The logic is easily seen to be sound: if \(\Gamma \vdash \varphi\), then \(\Gamma \models \varphi\). The main contribution of this section is the completeness of this system.

Some syntactic abbreviations. The language lacks boolean connectives, but it is convenient to use an informal notation for it. It is also worthwhile specifying an operation of duals.

\[
\neg(\text{All p are q}) = \text{Some p are q}' \\
\neg(\text{Some p are q}) = \text{All p are q}'
\]

Here are some uses of this notation. We say that \(\Gamma\) is inconsistent if for some \(\varphi\), \(\Gamma \vdash \varphi\) and \(\Gamma \vdash \neg \varphi\). The first part of Lemma 3.21 tells us that if \(\Gamma \vdash \text{Some p is p}'\), then \(\Gamma\) is inconsistent. Also, we have the following result:

Exercise 30  If \(\varphi \vdash \psi\), then \(\neg \psi \vdash \neg \varphi\). What is the parallel result for the \(d\)-operation?  [This fact is not needed in this chapter, but we recommend thinking about it as a way of getting familiar with the rules.]

3.3.1. Continuing with the alternative notation. We have seen the language \(S\) in our alternative notation in Section 2.2.1. We extend this to the language \(S^\dagger\) in several steps. First, in addition to unary atoms, we have literals. Foreshadowing later notation, we use letters like \(l\) and \(m\) for literals. The literals come with a complementation operation, and we assume that \(l'' = l\) for all \(l\). Then sentences of \(S^\dagger\) are those of the form

\[
(3.2)  \quad \exists(l, m), \quad \forall(l, m).
\]

We gloss these as as for \(S\), except that we sometimes require negated subjects:

\[
\exists(\bar{p}, \bar{q})  \quad \text{Some non-p is not a q} \quad \forall(\bar{p}, q)  \quad \text{Every non-p is a q}.
\]
3.4. Completeness via Representation of Orthoposets

An important step in our work is to develop an algebraic semantics for $S^1$. There are several definitions, and then a representation theorem. As with other uses of algebra in logic, the point is that the representation theorem is also a model construction technique.

An orthoposet is a tuple $P = (P, \leq, 0, \prime)$ such that

1. $(P, \leq)$ is a partial order: $\leq$ is a reflexive, transitive, and antisymmetric relation on the set $P$.
2. 0 is a minimum element: $0 \leq p$ for all $p \in P$.
3. $x \mapsto x'$ is an antitone map in both directions: $x \leq y$ iff $y' \leq x'$.
4. $x \mapsto x'$ is involutive: $x'' = x$.
5. complement inconsistency: If $x \leq y$ and $x \leq y'$, then $x = 0$.

The notion of an orthoposet mainly appears in papers on quantum logic. (In fact, the stronger notion of an orthomodular poset appears to be more central there. However, I do not see any application of this notion to logics of the type considered in this chapter.)

Example 3.22 The example below is sometimes called the Chinese lantern, and we’ll call it $P_2$.

```
[Diagram of the Chinese lantern with nodes 0, p, p', q, q', 1, and edges between them.]
```

Here and elsewhere, we understand $(x')' = x$, $0' = 1$, $1' = 0$.

Example 3.23 For example, for all sets $p$ we have an orthoposet $(P(X), \subseteq, \emptyset, \prime)$, where $\subseteq$ is the inclusion relation, $\emptyset$ is the empty set, and $a' = X \setminus a$ for all subsets $a$ of $p$.

Definition 3.24 Let $\Gamma$ be any set of sentences in $S^1$. $\Gamma$ need not be consistent. Definition 2.9 defines the fundamental relation $\leq$ from $\Gamma$ and the logic, and Proposition 2.10 shows this relation to be a preorder. We have an induced equivalence relation $\equiv$, and we take $\text{Lit}_\Gamma$ to be the quotient set $\text{Lit}/\equiv$. That is, we define $u \equiv v$ to mean $u \leq v$ and $v \leq u$. This quotient set $\text{Lit}/\equiv$ is a poset under the induced relation: if $[u] \leq [v]$ and $[v] \leq [u]$, then $u \equiv v$ so that $[u] = [v]$. If there is some $p$ such that $p \leq p'$, then for all $q$ we have $[p] \leq [q]$ in $\text{Lit}/\equiv$. In this case, set 0 to be $[p]$ for any such $p$. (If such $p$ exists, its equivalence class is unique.) We finally define $[p]' = [p']$. If there is no $p$ such that $p \leq p'$, we add fresh elements 0 and 1 to $\text{Lit}/\equiv$. We then stipulate that $0' = 1$, and that for all $x \in P_\Gamma$, $0 \leq x \leq 1$.

It is not hard to check that we have an orthoposet $\text{Lit}/\equiv = (\text{Lit}/\equiv, \leq, 0, \prime)$. The antitone property comes from the axiom with the same name, and the complement inconsistency is verified using the similarly-named part of Lemma 3.21.

---

3. The more standard name for this seems to be $M02$, with $MO$ standing for modular ortholattice.
Example 3.25  This example pertains to Exercise 27 from before. Let $\Gamma$ be defined by:

$$\Gamma = \{ \text{All } y \text{ are } x, \text{All } y' \text{ are } x, \text{All } z' \text{ are } y, \text{All } z \text{ are } y', \text{All } z \text{ are } w \}.$$ 

Then

$$[x] = \{x\} \quad [x'] = \{x'\}$$
$$[y] = \{y, z'\} \quad [y'] = \{y', z\}$$
$$[z] = \{y', z\} \quad [z'] = \{y, z'\}$$
$$[w] = \{w\} \quad [w'] = \{w'\}$$

Here is a picture of the orthoposet $\mathbb{P}_\Gamma$:

![Orthoposet Diagram]

Definition 3.26  Let $\mathbb{P}$ and $\mathbb{Q}$ be orthoposets. A morphism of orthoposets $f : \mathbb{P} \to \mathbb{Q}$ is a map $m : P \to Q$ preserving the order (if $x \leq y$, then $mx \leq my$), the complement $m(x') = (mx)'$, and minimum elements ($m0 = 0$). We say $m$ is strict if the following extra condition holds: $x \leq y$ iff $mx \leq my$.

Definition 3.27  A point of an orthoposet $\mathbb{P} = (P, \leq, 0, ')$ is a subset $S \subseteq P$ with the following properties:

1. If $p \in S$ and $p \leq q$, then $q \in S$ (S is up-closed).
2. For all $p$, either $p \in S$ or $p' \in S$ (S is complete), but not both (S is consistent).

One should compare this definition with that of an ultrafilter (see Definition 3.10). The idea in both cases is to reflect on the concrete forms of the structures involved. For boolean algebras, this would mean power set algebras, and for orthoposets this again would be the power set orthoposets. Given one of these concrete structures, call it $\mathcal{P}(X)$, and some $x$ in the underlying set of the structure, we can ask for an algebraic characterization of the sets $s_x$ as $x$ varies over $X$:

$$\{A \in \mathcal{P}(X) : x \in A\}$$

By an “algebraic characterization” here, we mean something expressible with the language at hand: for boolean algebras, we would use $\top, \bot, \neg, \wedge,$ and $\vee$ (and anything else defined in terms of them); for orthoposets, we would use $\leq, 0$ and $'$. The idea is to come up with a definition of what it means for a subset of the structure to be one of the sets $s_x$.

Example 3.28  Let $X = \{1, 2, 3\}$, and let $\mathcal{P}(X)$ be the power set orthoposet from Example 3.23. Then $S$ is a point, where

$$S = \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}.$$
(More generally, if $p$ is any finite set, then the collection of subsets of $p$ containing more than half of the elements of $p$ is a point of $\mathcal{P}(X).$) Also, it is easy to check that the points on this $\mathcal{P}(X)$ are exactly $S$ as above and the three principal ultrafilters $1^\uparrow, 2^\uparrow,$ and $3^\uparrow$ mentioned in Example 3.11. $S$ shows that a point of a boolean algebras need not be an ultrafilter or even a filter. Also, the lemma just below shows that for $\mathcal{P}(X),$ a collection of elements is included in a point iff every pair of elements has a non-empty intersection.

**Lemma 3.29** For a subset $S_0$ of an orthoposet $\mathbb{P} = (P, \leq, \)'$, the following are equivalent:

1. $S_0$ is a subset of a point $S$ of $\mathbb{P}$.
2. For all $x, y \in S_0, x \nleq y'$.

**Proof** Clearly (1) $\Rightarrow$ (2). For the more important direction, use Zorn’s Lemma to get a $\subseteq$-maximal superset $S_1$ of $S_0$ with the consistency property. Let $S = \{q: (\exists p \in S_1) q \geq p\}$. So $S$ is up-closed. We check that consistency is not lost: suppose that $r, r' \in S$. Then there are $q_1, q_2 \in S_1$ such that $r \geq q_1$ and $r' \geq q_2$. But then $q_2' \geq r \geq q_1$. Since $q_1 \in S_1,$ so too $q_2' \in S_1$. Thus we see that $S_1$ is not consistent, and this is a contradiction. To conclude, we only need to see that for all $r \in P,$ either $r$ or $r'$ belongs to $S$. If $r \notin S,$ then $r \notin S_1$. By maximality, there is $q \in S_1$ such that $q_1 \leq r'$. (For otherwise, $S_1 \cup \{r\}$ would be a consistent proper superset of $S_1.$) And as $r' \notin S,$ there is $q_2 \in S_1$ such that $q_2 \leq r$. Then as above $q_1 \leq q_2'$, leading to the same contradiction. □

We now present a representation theorem that implies the completeness of the logic. It is due to Calude, Hertling, and Svozil [2]. We also state an additional technical point.

**Theorem 3.30** (Representation Theorem for Orthoposets [2, 11, 41]) Let $\mathbb{P} = (P, \leq, \)'$ be an orthoposet. There is a set $\text{points}(P)$ and a strict morphism of orthoposets $m : \mathbb{P} \to \mathcal{P}(\text{points}(P))$.

Moreover, if $S \cup \{p\} \subseteq P$ has the following two properties, then $m(p) \setminus \bigcup_{q \in S} m(q)$ is non-empty:

1. For all $q \in S, p \nleq q$.
2. For all $q, r \in S, q \nleq r'$.

**Proof** Let $\text{points}(P)$ be the collection of points of $\mathbb{P}$. The map $m$ is defined by $m(p) = \{S : p \in S\}$. The preservation of complement comes from the completeness and consistency requirement on points, and the preservation of order from the up-closed-ness. Clearly $m0 = \emptyset$. We must check that if $q \nleq p$, then there is some point $S$ such that $p \in S$ and $q \notin S$. For this, take $S = \{q\}$ in the “moreover” part. And for that, let $T = \{p\} \cup \{q' : q \in S\}$. Lemma 3.29 applies, and so there is some point $u \supseteq T$. Such $u$ belongs to $m(p)$. But if $q \in S$, then $q' \in T \subseteq U$; so $u$ does not belong to $m(q)$. □

**Example 3.31** Here is a picture which illustrates the workings of the Representation Theorem 3.30. We presented an orthoposet $\mathbb{P}_2$ in Example 3.22. Three
subsets of the underlying set are points, and these are shown in colors below\textsuperscript{4}.

\begin{center}
\begin{tikzpicture}
\filldraw[fill=red!30] (0,0) circle (2pt);
\filldraw[fill=blue!30] (1,0) circle (2pt);
\filldraw[fill=green!30] (2,0) circle (2pt);
\filldraw[fill=gray!30] (3,0) circle (2pt);
\filldraw[fill=red!30] (0,1) circle (2pt);
\filldraw[fill=blue!30] (1,1) circle (2pt);
\filldraw[fill=green!30] (2,1) circle (2pt);
\filldraw[fill=gray!30] (3,1) circle (2pt);
\draw (0,0) -- (0,1);
\draw (1,0) -- (1,1);
\draw (2,0) -- (2,1);
\draw (3,0) -- (3,1);
\draw (0,0) -- (1,1);
\draw (1,0) -- (2,1);
\draw (2,0) -- (3,1);
\draw (3,0) -- (0,1);
\end{tikzpicture}
\end{center}

We call these points $\bullet$, $\cdot$, $\ast$, and $\circ$. The orthoposet
\[ Q = \mathcal{P}(\{\bullet, \cdot, \ast, \circ\}) \]
has sixteen elements, so we shall not display it. But we can illustrate the function
\[ m : \mathcal{P}_2 \rightarrow Q. \]
For example, \[ m(p) = \{\bullet, \cdot\}, \]
because the points to which \[ p \] belongs
are $\bullet$ and $\cdot$. Similarly, \[ m(0) = \emptyset. \]
Here is a picture of $\mathcal{P}_2$ and its image under $m$ inside $Q$:

\begin{center}
\begin{tikzpicture}
\filldraw[fill=red!30] (0,0) circle (2pt);
\filldraw[fill=blue!30] (1,0) circle (2pt);
\filldraw[fill=green!30] (2,0) circle (2pt);
\filldraw[fill=gray!30] (3,0) circle (2pt);
\filldraw[fill=red!30] (0,1) circle (2pt);
\filldraw[fill=blue!30] (1,1) circle (2pt);
\filldraw[fill=green!30] (2,1) circle (2pt);
\filldraw[fill=gray!30] (3,1) circle (2pt);
\draw (0,0) -- (0,1);
\draw (1,0) -- (1,1);
\draw (2,0) -- (2,1);
\draw (3,0) -- (3,1);
\draw (0,0) -- (1,1);
\draw (1,0) -- (2,1);
\draw (2,0) -- (3,1);
\draw (3,0) -- (0,1);
\draw (0,0) -- (1,0);
\draw (1,0) -- (2,0);
\draw (2,0) -- (3,0);
\draw (3,0) -- (0,0);
\end{tikzpicture}
\end{center}

\textbf{3.4.1. Completeness.} The completeness theorem is based on algebraic ma-
chinery that we have just seen.

\textbf{Lemma 3.32 (Pratt-Hartmann)} Suppose that $\Gamma \models \text{Some } p \text{ are } q$. Then there
is some sentence in $\Gamma$ some, say $\text{Some } a \text{ are } b$, such that $\Gamma \cup \{\text{Some } a \text{ are } b\} \models \text{Some } p \text{ are } q$.

\textbf{Proof} If not, then for every $\varphi \in \Gamma$, there is a model $M_{\varphi} \models \Gamma \cup \{\varphi\}$
and $[p] \cap [q] = \emptyset$ in the model. Take the disjoint union of the models $M_{\varphi}$ to get a
model of $\Gamma \cup \Gamma = \Gamma$ where $\text{Some } p \text{ are } q$ fails.\hfill $\square$

The next lemma is the key step in the completeness theorem of this section.

\textbf{Lemma 3.33} Let $\Gamma \subseteq S^1$. There is a model $M = (M, \models)$ such that

1. $M \models \Gamma_{\text{all}}$.
2. If $\varphi \in L(\text{all})$ and $M \models \varphi$, then $\Gamma \models \varphi$.
3. If $\Gamma$ is consistent, then also $M \models \Gamma_{\text{some}}$.

\textbf{Proof} We write $\mathcal{P}_\Gamma$ for the orthoposet $\text{Lit}/\equiv$. (Recall that this was obtained
by from the set of literals by taking quotient by $p \equiv q$ iff $p \leq q \leq p$. See Defini-
tion 3.24.) Recall that $\mathcal{P}$ is the set of unary atoms that is the base of the syntax of
the language at hand. Let $n$ be the natural map of $\mathcal{P}$ into $\mathcal{P}_\Gamma$, taking an atom $p$ to
its equivalence class $[p]$. Even though $\mathcal{P}$ is not an orthoposet with its order $\leq$, it is
a preorder (see Proposition 2.10). This map $n$ is an order-preserving map from one
preorder $\mathcal{P}$ into another. Moreover, $n$ preserves the order in both directions. We
also apply Theorem 3.30, to obtain a strict morphism of orthoposets $m$ as shown below:

\[ \mathcal{P} \xrightarrow{n} \mathcal{P}_\Gamma \xrightarrow{m} \text{points}(\mathcal{P}_\Gamma) \]

\textsuperscript{4}If you cannot see the colors, the four points are $\{p, q, 1\}$, $\{p', q, 1\}$, $\{p, q', 1\}$, and $\{p', q', 1\}$.
3.4. Completeness via Representation of Orthoposets

Let $M = \text{points} (\mathbb{P}_\Gamma)$, and let $[\cdot] : \mathbb{P}_\Gamma \to \mathcal{P}(M)$ be the composition $m \circ n$, regarded as an order-preserving function on preorders. We thus have a model $M = (\text{points} (\mathbb{P}_\Gamma), [\cdot])$.

We check that $M \models \Gamma$. Note that $n$ and $m$ are strict monotone functions. So the semantics has the property that the $\text{All}$ sentences holding in $M$ are exactly the consequences of $\Gamma$. We turn to a sentence in $\Gamma$ some such as $\text{Some } u \text{ are } v$. Assuming the consistency of $\Gamma$, $u \not\leq v'$. Thus $[[u]] \not\subseteq ([[v]])'$. That is, $[[u]] \cap [[v]] \neq \emptyset$. 

Unfortunately, the last step in this proof is not reversible, in the following precise sense. It does not follow from $u \not\leq v'$ that $\Gamma \vdash \text{Some } u \text{ are } v$. (For example, if $\Gamma$ is the empty set we have $u \not\leq v'$, and indeed $M(\Gamma) \models \text{Some } u \text{ are } v$. But $\Gamma$ only derives valid sentences.)

**Theorem 3.34** \( \Gamma \vdash \varphi \text{ iff } \Gamma \models \varphi. \)

**Proof** As always, the soundness half is trivial. Suppose that $\Gamma \models \varphi$; we show that $\Gamma \vdash \varphi$. We may assume that $\Gamma$ is consistent.

If $\varphi \in \mathcal{L}(\text{all})$, consider $M(\Gamma)$ from Lemma 3.33. It is a model of $\Gamma$, hence of $\varphi$; and then by the property the second part of the lemma, $\Gamma \vdash \varphi$.

For the rest of this proof, let $\varphi$ be $\text{Some } p \text{ are } q$. From $\Gamma$ and $\varphi$, we find $a$ and $b$ satisfying the conclusion of Lemma 3.32.

We again use Lemma 3.33 and consider the model $M = M(\mathbb{L}/\equiv_{\text{all}})$ of points on $\mathbb{L}/\equiv_{\text{all}}$. $M \models \Gamma_{\text{all}}$.

Consider $\{[a], [b], [p']\}$. If this set were a subset of a point $x$, then consider $\{x\}$ as a one-point submodel of $M$. In the submodel, $\Gamma_{\text{all}} \cup \{\text{Some } a \text{ are } b\}$ would hold, and yet $\text{Some } p \text{ are } q$ would fail since $[p] = \emptyset$.

We use Lemma 3.29 to divide into cases:

1. \( a \leq a' \).
2. \( a \leq b' \).
3. \( a \leq p \).
4. \( b \leq b' \).
5. \( b \leq p \).
6. \( p' \leq p \).

(More precisely, the first case would be $[a] \leq [a']$. By strictness of the natural map, this means that $a \leq a'$; that is, $\Gamma_{\text{all}} \vdash \text{All } a \text{ are } a'$. In cases (1), (2), and (4), we easily see that $\Gamma$ is inconsistent, contrary to the assumption at the outset. Case (6) implies that both (3) and (5) hold. Thus we may restrict attention to (3) and (5).

Next, consider $\{a, b, q\}$. The same analysis gives two other cases, independently: $a \leq q$, and $b \leq q$. Putting these together with the other two gives four pairs. The following are representative:

- $a \leq p$ and $b \leq q$: Using $\text{Some } a \text{ are } b$, we see that $\Gamma \vdash \text{Some } p \text{ are } q$.
- $a \leq p$ and $a \leq q$: We first derive $\text{Some } a \text{ are } b$, and then again we see $\Gamma \vdash \text{Some } p \text{ are } q$.

This completes the proof.

**Exercise 31** Look back at Exercise 27. When we posed this problem, you had no tools to solve it and were left to merely guess at a solution. At this point, we have a systematic way to solve it using Theorem 3.34 and its key ingredient, Lemma 3.33.
Example 3.25 shows a picture of the syntactic orthoposet $\mathbb{P}_\Gamma$. What are the points of this orthoposet?

(2) Define the model $M$ from Lemma 3.33 explicitly. That is, say what the universe $M$ is, and also $[x]$, $[y]$, and $[z]$.

(3) Verify that all the sentences in $\Gamma$ are true in your model, and that $\text{All } y\text{ are } w$ is false.

3.5. Adding Boolean Connectives on Sentences: $S^\dagger _{bc}$

We continue our development a little further, mentioning a larger system whose completeness can be obtained by using our results which we have seen. Let

$$(3.4) \quad S^\dagger _{bc} = \text{SentLogic}(S^\dagger )$$

(Compare with (3.1) at the beginning of Section 3.2.) This language is just sentential logic built over $S^\dagger$ as the set of atomic propositions. For the semantics, we again use valuations derived from models $M = (M, [] )$ as we have seen them in Section 3.3. We have notions like $M \models \varphi$ and $\Gamma \models \varphi$, defined in the usual ways.

A proof system $S^\dagger _{bc}$ is listed in Figure 3.5. It contains all the axioms of the system $S^\dagger$ in Figure 3.5, and in addition has axiom (6):

$$\text{Some } p\text{ are } q' \leftrightarrow \neg(\text{All } p\text{ are } q)$$

As before, the only rule is modus ponens.

**Lemma 3.35** Let $\Gamma \subseteq S^\dagger$. If $\Gamma \vdash \varphi$ in the logic $S^\dagger$ (Figure 3.4), then $\Gamma \vdash \varphi$ in the logic $S^\dagger _{bc}$ (Figure 3.5).

**Proof** By induction on the proof trees in the logic $S^\dagger$, and then by cases on the rule used at the root of the tree. For a one-point tree $T$, say with label $\varphi$, it must be the case that either $\varphi$ is $\text{All } p\text{ are } p$, or else $\varphi \in \Gamma$. Either way, the same tree is a tree in $S^\dagger _{bc}$.

Next, suppose that our lemma is true for a tree $T$, and let $T'$ come from $T$ by applying $(\text{Some}_2)$ at the root. Suppose that the root of $T$ is $\text{Some } p\text{ are } q$. Let $\alpha$, $\beta$, $\gamma$, and $\varphi$ be defined as follows:

$$\alpha \quad \text{Some } p\text{ are } q\quad \gamma \quad \text{All } q\text{ are } q$$

$$\beta \quad \text{Some } q\text{ are } p\quad \varphi \quad (\gamma \land \alpha) \rightarrow \beta$$

Note that $\gamma$ and $\varphi$ are axioms, as is $\varphi \rightarrow (\gamma \rightarrow (\alpha \rightarrow \beta))$. By induction hypothesis, $\Gamma \vdash \alpha$ in $S^\dagger _{bc}$. Now we have a proof tree for $S^\dagger _{bc}$:

```
(\text{Some}_2) \text{ comes from axioms (1) and (3).}
```

Thus $\Gamma \vdash \beta$, as desired. For $(\text{Antitone})$, use the point about $(\text{Some}_2)$ and also axiom (6) twice. For $(\text{One})$, use axiom (6) to see that $\text{All } p'\text{ are } p \leftrightarrow \neg(\text{Some } p\text{ are } p')$. This along with axiom (5) gives $\text{All } p'\text{ are } p \leftrightarrow \text{All } p'\text{ are } q'$. Now we use our point about $(\text{Antitone})$ to see that $\text{All } p'\text{ are } p \leftrightarrow \text{All } q\text{ are } p$. For $(X)$, use the Deduction Theorem and a sentential tautology. □
3.5. ADDING BOOLEAN CONNECTIVES ON SENTENCES: $S_{bc}^1$

**Theorem 3.36** The logical system $S_{bc}^1$ is sound and complete: $\Gamma \models \varphi$ iff $\Gamma \vdash \varphi$.

**Proof** The soundness is easy; here is a proof of completeness. By Lemma 3.15, we need only prove that a set $\Gamma$ which is maximal consistent in $S_{bc}^1$ has a model. So fix such a set $\Gamma$.

Let $\Gamma_{at}$ and $\Gamma_{+at}$ be as in Definition 3.5. $\Gamma_{+at}$ must be consistent in $S_1^1$; for if not, then it would be inconsistent in $S_{bc}^1$ by Lemma 3.35. By Lemma 3.33, we have a model $M \models \Gamma_{at}$.

We claim that in addition $M \models \Gamma_{+at}$. To see this, let $\varphi \in S_1^1$. and suppose that $\neg \varphi \in \Gamma_{+at}$. If $\varphi$ is *All p are q*, then let $\psi$ be *Some p are q*'. Note that $\vdash \varphi \rightarrow \psi$ in $S_{bc}^1$ using axiom (6). It follows that $\psi \in \Gamma$, by maximality. Indeed $\varphi$ is an atom, so $\varphi \in \Gamma_{at}$. But then $M \models \varphi$. If $\varphi$ is *Some p are q*, then essentially the same argument works.

At this point, we know that $M \models \Gamma_{+at}$. And then it follows from Lemma 3.6 that $M \models \Gamma$. □

It is possible to add boolean compounds of the NPs in this fragment. Once one does this, the axiomatization and completeness result become quite a bit simpler, since the system becomes a variant of boolean algebra.

Sources for this chapter. The completeness of the system in Section 3.2 appears in Lukasiewicz [12] (in work with Slupecki; they also showed decidability), and also by Westerståhl [38], and axioms 1–6 are essentially the system SYLL. The result here is taken from [18].

The source for the logic $S$ in this chapter is Moss [22]. Papers mentioning orthoposets and representation theorems quite similar to Theorem 3.30 include Calude, Hertling, and Svozil [2], Zierler and Schlessinger [41] and Katrnoška [11].

**Figure 3.5.** The axiom added to the system in Figure 3.3. The resulting sentential logical system is called $S_{bc}^1$, and it is sound and complete for $S_{bc}^1$.

(6) Some $p$ are $q' \leftrightarrow \neg (All p are q)$
3. SYLLOGISTIC LOGIC WITH COMPLEMENTS: $S^†$

3.5.1. Summary.

<table>
<thead>
<tr>
<th>Rule of Inference:</th>
<th>$\frac{\varphi}{\psi} \rightarrow \frac{\psi}{\psi}$</th>
</tr>
</thead>
</table>

Axioms:

(0) All substitution instances of sentential tautologies.
(1) All $p$ are $p$
(2) $\text{All } p \text{ are } n \land \text{All } n \text{ are } q \rightarrow \text{All } p \text{ are } q$
(3) $\text{All } q \text{ are } n \land \text{Some } p \text{ are } q \rightarrow \text{Some } n \text{ are } p$
(4) $\text{Some } p \text{ are } q \rightarrow \text{Some } p \text{ are } p$
(5) $\neg (\text{Some } p \text{ are } p) \rightarrow \text{All } p \text{ are } q$
(6) $\text{Some } p \text{ are } q' \leftrightarrow \neg (\text{All } p \text{ are } q)$

Rule of Inference: $\frac{\varphi}{\psi} \rightarrow \frac{\psi}{\psi}$ modus ponens

<table>
<thead>
<tr>
<th>All $p$ are $p$</th>
<th>Axiom</th>
<th>Some $p$ are $q$</th>
<th>Some $p$ are $q$</th>
<th>Some $p$ are $q$</th>
<th>Some $p$ are $q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>All $p$ are $n$</td>
<td>All $n$ are $q$</td>
<td>All $p$ are $q$</td>
<td>Some $p$ are $p$</td>
<td>Some $p$ are $p$</td>
<td>Some $p$ are $p$</td>
</tr>
<tr>
<td>All $q$ are $q'$</td>
<td>All $q$ are $p$</td>
<td>Zero</td>
<td>All $q$ are $q$</td>
<td>All $q$ are $q$</td>
<td>One</td>
</tr>
<tr>
<td>All $p$ are $q'$</td>
<td>All $p$ are $q$</td>
<td>Antitone</td>
<td>All $p$ are $q$</td>
<td>Some $p$ are $q'$</td>
<td>$S$</td>
</tr>
<tr>
<td>All $p$ are $q$</td>
<td>All $p$ are $q$</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The logical systems studied in this chapter are listed above. On the top, we have sentential logical systems. The smallest of these contains only the tautologies as axioms (0), and the only rule is modus ponens. This system is sound and complete for sentential logic. Adding (1)–(5) gives a system which is sound and complete for the language $L_{bc}$, sentential logic interpreted on models. Adding (6) gives a sound and complete system for $S^bc$. The bottom of the figure contains the proof system for $S^†$. This logic is a syllogistic logic rather than a sentential logic.
We next show that it is possible to have complete syllogistic systems for logics which go are not first-order. We regard this as a proof-of-concept; it would be of interest to get complete systems for richer fragments, such the ones in Pratt-Hartmann [26].

We write \( \exists \geq (x, y) \) for There are at least as many \( x \) as \( y \), and we are interested in adding these sentences to our fragments. We are usually interested in sentences in this fragment on finite models. We write \( |S| \) for the cardinality of the set \( S \). The semantics is that \( M \models \exists \geq (x, y) \) iff \( |[x]| \geq |[y]| \) in \( M \).

It might be interesting to note that the compactness theorem fails for the semantics.

\[ \Gamma = \{ \exists \geq (x_1, x_2), \exists \geq (x_2, x_3), \ldots, \exists \geq (x_n, x_{n+1}), \ldots \} \]

Suppose towards a contradiction that \( M \) were a canonical model for \( \Gamma \). In particular, \( M \models \Gamma \). Then \( |[x_1]| \geq |[x_2]| \geq \ldots \). For some \( n \), we have \( |[x_n]| = |[x_{n+1}]| \). Thus \( M \models \exists \geq (x_{n+1}, x_n) \). However, this sentence does not follow from \( \Gamma \).

**Remark** In the remainder of this section, \( \Gamma \) denotes a finite set of sentences.

In this section, we consider \( \mathcal{L}(\text{all}, \exists \geq) \). For proof rules, we take the rules in Figure 4.1 together with the rules for \( \text{All} \) in Figure 2.1. The system is sound. The last rule is perhaps the most interesting, and it uses the assumption that our models are finite. That is, if all \( y \) are \( x \), and there are at least as many elements in the bigger set \( y \) as in \( x \), then the sets have to be the same.

We need a little notation at this point. Let \( \Gamma \) be a (finite) set of sentences. We write \( x \leq_c y \) for \( \Gamma \vdash \exists \geq (y, x) \). We also write \( x \equiv_c y \) for \( x \leq_c y \leq_c x \), and \( x <_c y \) for \( x \leq_c y \) but \( x \neq_c y \). We continue to write \( x \leq y \) for \( \Gamma \vdash \text{All } x \text{ are } y \). And we write \( x \equiv y \) for \( x \leq y \leq y \).

**Proposition 4.1** Let \( \Gamma \subseteq \mathcal{L}(\text{all}, \exists \geq) \) be a (finite) set. Let \( \text{Lit}/\equiv \) be the set of literals in \( \Gamma \).

1. If \( x \leq y \), then \( x \leq_c y \).
2. \((\text{Lit}/\equiv, \leq_c)\) is a preorder: a reflexive and transitive relation.
3. If \( x \leq_c y \leq x \), then \( x \leq y \).
4. If \( x \leq_c y \), \( x \equiv x' \), and \( y \equiv y' \), then \( x' \leq_c y' \).
5. \((\text{Lit}/\equiv, \leq_c)\) is pre-wellfounded: a preorder with no descending sequences in its strict part.

**Proof** Part (1) uses the first rule in Figure 4.1. In part (2), the reflexivity of \( \leq_c \) comes from that of \( \leq \) and part (1); the transitivity is by the second rule of \( \exists \geq \). Part (3) is by the last rule of \( \exists \geq \). Part 4 uses part (1) and transitivity. Part 5 is just a summary of the previous parts. \( \square \)
We claim that if \( \Gamma \) contains \( M \) model \( z \) (4.1) \[ \text{trivial). So \( \text{the second assertion. For this, we may assume that} \] \( w \equiv z \) as desired. 

Theorem 4.2 The logic of Figures 2.1 and 4.1 is complete for \( \mathcal{L}(\text{all}, \exists^\geq) \).

Proof Suppose that \( \Gamma \models \exists^\geq(y, x) \). Let \( \{\star\} \) be any singleton, and define a model \( M \) by taking \( M \) to be a singleton \( \{\star\} \), and

\[
[z] = \begin{cases} M & \text{if } \Gamma \vdash \exists^\geq(z, x) \\ \emptyset & \text{otherwise} \end{cases}
\]

We claim that if \( \Gamma \) contains \( \exists^\geq(w, v) \) or All \( v \) are \( w \), then \( [v] \subseteq [w] \). We only verify the second assertion. For this, we may assume that \( [v] \neq \emptyset \) (otherwise the result is trivial). So \( [v] = M \). Thus \( \Gamma \vdash \exists^\geq(v, x) \). So we see that \( \Gamma \vdash \exists^\geq(w, x) \). From this we conclude that \( [w] = M \). In particular, \( [v] \subseteq [w] \).

Now our claim implies that \( M \models \Gamma \). Therefore \( ||x|| \leq ||y|| \). And \( [x] = M \), since \( \Gamma \vdash \exists^\geq(x, x) \). Hence \( ||y|| = M \) as well. But this means that \( \Gamma \vdash \exists^\geq(y, x) \), just as desired.

We have shown one case of the general completeness theorem that we are after. In the other case, we have \( \Gamma \models \text{All } x \text{ are } y \). We construct a model \( M = \mathcal{M}_\Gamma \) such that for all \( A \) and \( B \),

\[
\begin{align*}
(\alpha) & \ [A] \subseteq [B] \text{ iff } A \leq B. \\
(\beta) & \text{ If } A \leq_c B, \text{ then } ||[A]|| \leq ||[B]||.
\end{align*}
\]

Let \( \text{Lit}/=\equiv \equiv_c \) be the (finite) set of equivalences classes of atoms in \( \Gamma \) under \( \equiv_c \). This set is then well-founded by the natural relation induced on it by \( \leq_c \). It is then standard that we may list the elements of \( \text{Lit}/=\equiv \equiv_c \) in some order

\[
[u_0], [u_2], \ldots, [u_k]
\]

with the property that if \( u_i <_c u_j \), then \( i < j \). (But if \( i < j \), then it might be the case that \( u_i \equiv_c u_j \).)

We define by recursion on \( i \leq k \) the interpretation \( [v] \) of all \( v \in [u_i] \). Suppose we have \( [v] \) for all \( j < i \) and all \( w \equiv_c u_j \). Let

\[
X_i = \bigcup_{j < i, w \equiv u_j} [v],
\]

and note that this is the set of all points used in the semantics of any atom so far. Let \( n = 1 + |X_i| \). For all \( v \equiv_c u_i \), we shall arrange that \( [v] \) be a set of size \( n \).

Now \( [u_i] \) is the equivalence class of \( u_i \) under \( \equiv_c \). It splits into equivalence classes of the finer relation \( \equiv \). For a moment, consider one of those finer classes, say \( [A] \equiv \). We must interpret each atom in this class by the same set. For this \( A \), let

\[
A = \bigcup \{ [B] : (\exists j < i) v_j \equiv_c B \leq A \}.
\]

Note that \( A \subseteq X_i \) so that \( |A| < n \). We set \( [A] \) to be \( A \) together with \( n - |A| \) fresh elements. Moving on to the other \( \equiv \)-classes which partition the \( \equiv_c \)-class of \( u_i \), we
do the same thing. We must insure that for \( A \neq A' \), the fresh elements added into \([A']\) are disjoint from the fresh elements added into \([A]\).

This completes the definition of \( M \). We check that so that conditions (\( \alpha \)) and (\( \beta \)) are satisfied. It is easy to first check that for \( i < j \), \( ||u_i|| < ||u_j|| \). It might also be worth noting that \( ||u_0|| \neq \emptyset \), so no \([A]\) is empty.

For (\( \beta \)), let \( A \leq_c B \). Let \( i \) and \( j \) be such that \( A \equiv_c u_i \) and \( B \equiv_c u_j \). If \( A \equiv_c B \), then \( i = j \) and the construction arranged that \([A]\) and \([B]\) be sets of the same cardinality. If \( A <_c B \), then \( i < j \) by the way we enumerated the \( u \)'s, and so \( ||A|| = ||u_i|| < ||u_j|| = ||B|| \).

Turning to (\( \alpha \)), we argue the two directions separately. Suppose first that \( A \leq B \). Then \( A \leq_c B \). If \( A <_c B \), then \([A] \subseteq B \subseteq [B] \). If \( A \equiv_c B \), then we also have \( A \equiv B \). The construction has then arranged that \([A] = [B] \). In the other direction, assume that \([A] \subseteq [B] \), and let \( i \) and \( j \) be such that \( A \equiv_c u_i \) and \( B \equiv_c u_j \). On cardinality grounds, \( i \leq j \). If \( i < j \), then the construction shows that \( A \leq B \). (For if not, \([A]\) would be a non-empty set disjoint from \([B]\), and this contradicts \([A] \subseteq [B] \).) Finally (for perhaps the most interesting point), if \( i = j \), then we must have \( A \equiv B \): otherwise, the construction arranged that both \( A \) and \( B \) have at least one point that is not in the other, due to the “1+” in the definition of \( \equiv \).

Since (\( \alpha \)) and (\( \beta \)), we know that \( M \models \Gamma \). Recall that we are assuming that \( x \leq y \) holds semantically from \( \Gamma \); we need to show that this assertion is derivable in the logic. But \([x] \subseteq [y] \) in the model, and so by (\( \alpha \)), we indeed have \( \Gamma \vdash x \leq y \).  

**Exercise 32**  
Consider the following two rules:

\[
\begin{align*}
\text{Some } y \text{ are } y & \quad \exists^\ast(x, y) \quad \text{No } y \text{ are } y \\
\text{Some } x \text{ are } x & \quad \exists^\ast(x, y)
\end{align*}
\]

(1) Add the rule on the left to our existing system for \( \mathcal{L}(\text{all, some}) \) and to the rules in Figure 4.1. Prove that the resulting system is complete for \( \mathcal{L}(\text{all, some}, \exists^\ast) \).

(2) Similarly, add the rule on the right to get a complete logic for the resulting language.

(3) Finally, add both rules to again get a complete logic for the resulting language.

**4.1. Adding \( \exists^\ast \) to the Boolean Syllogistic Fragment**

We now put aside Most and return to the study of \( \exists^\ast \) from earlier. We close this chapter with the addition of \( \exists^\ast \) to the fragment of Section 3.2.

Our logical system extends the axioms of Figure 3.3 by those in Figure 4.2. Note that the last new axiom expresses cardinal comparison. Axiom 4 in Figure 4.2 is just a transcription of the rule for No that we saw in Section 32. We do not need to also add the axiom

\[(\text{Some } y \text{ are } y) \land \exists^\ast(x, y) \rightarrow \text{Some } x \text{ are } x\]

because it is derivable. Here is a sketch, in English. Assume that there are some \( y \)'s, and there are at least as many \( x \)'s as \( y \)'s, but (towards a contradiction) that there are no \( x \)'s. Then all \( x \)'s are \( y \)'s. From our logic, all \( y \)'s are \( x \)'s as well. And since there are \( y \)'s, there are also \( x \)'s: a contradiction.
Figure 4.2. Additions to the system in Figure 3.3 for $\exists \geq$ sentences.

Notice also that in the current fragment we can express *There are more x than y*. It would be possible to add this directly to our previous systems.

**Theorem 4.3** The logic of Figures 3.3 and 4.2 is complete for assertions $\Delta \models \varphi$ in the language of boolean combinations of sentences in $L(\text{all, some, no, } \exists \geq)$.

**Proof** We need only build a model for a maximal consistent set $\Delta$ in the language of this section. We take the *basic* sentences to be those of the form *All x are y, Some x and y, J is M, J is an x*, $\exists \geq (x,y)$, or their negations. Let $\Gamma = \{ S : \Delta \models S \text{ and } \varphi \text{ is basic} \}$.

As in Chapter 3, we need only build a model $M \models \Gamma$ (see Lemma 3.6). We construct $M$ such that for all $A$ and $B$,

(a) $[A] \subseteq [B]$ iff $A \leq B$,

(b) $A \leq_c B$ iff $|[A]| \leq |[B]|$.

(c) For $A \leq_c B$, $[A] \cap [B] \neq \emptyset$ iff $A \uparrow B$.

Let $P$ be the set of atoms in $\Gamma$. Let $\leq_c$ and $\equiv_c$ be as in Section 4. Proposition 4.1 again holds, and now the quotient $P/\equiv_c$ is a linear order due to the last axiom in Figure 4.2. We write it as $[u_0] <_c [u_2] <_c \cdots <_c [u_k]$.

We define by recursion on $i \leq k$ the interpretation $[[v]]$ of all $v \in [u_i]$. The case of $i = 0$ is special. If $\Gamma \models \text{No } u_0$ is a $u_0$, then the same holds for all $w \equiv_c u_0$. In this case, we set $[w] = \emptyset$ for all these $w$. Note that by our fourth axiom in Figure 4.2, all of the other atoms $w$ are such that $\Gamma \vdash \exists w$. In any case, we must interpret the atoms in $[u_i]$ even when $\Gamma \vdash (\exists u_0)$. In this case, we may take each $[w]$ to be a singleton, with the added condition that $v \equiv w$ iff $[[v]] = [[w]]$.

Suppose we have $[[w]]$ for all $j \leq i$ and all $w \equiv_c u_j$. Let

$$X_{i+1} = \bigcup_{j \leq i, w \equiv_c u_j} [[w]],$$

and note that this is the set of all points used in the semantics of any atom so far. Let $m = |\Gamma_{some}|$, and let

$$n = 1 + m + |X_{i+1}|$$

(4.2)

For all $v \equiv_c u_{i+1}$, we shall arrange that $[[v]]$ be a set of size $n$. 

Now \([u_{i+1}]\) splits into equivalence classes of the finer relation \(\equiv\). For a moment, consider one of those finer classes, say \([A]_\equiv\). We must interpret each atom in this class by the same set. For this \(A\), let
\[
A = \bigcup\{[B]: (\exists j \leq i) \; v_j \equiv_e B \leq A\}.
\]
Note that \(A \subseteq X_{i+1}\) so that \(|A| \leq |X_{i+1}|\) for all \(A \equiv_e u_{i+1}\). We shall set \([A]\) to be \(A\) plus other points. Let \(A\) be the set of pairs \(\{A, B\}\) with \(B \equiv_e u_{i+1}\) and \(A \uparrow B\). (This is the same as saying that \(\text{Some} \; A \; \text{are} \; B \; \text{in} \; \Gamma_{\text{some}}\).) Notice that if both \(A\) and \(B\) are \(\equiv_e u_{i+1}\) and \(A \uparrow B\), then \(\{A, B\} \in A \cap B\). We shall set \([A]\) to be \(A \cup A\) plus one last group of points. If \(C <_e u_{i+1}\) and \(A \uparrow C\), then we must pick some element of \([C]\) and put it into \([A]\). Note that the number of points selected like this plus \(|A|\) is still \(\leq |\Gamma_{\text{some}}|\). So the number of points so far in \([A]\) is \(\leq |\Gamma_{\text{some}}| + m\).
We finally add fresh elements to \([A]\) so that the total is \(n\).

We do all of this for all of the other \(\equiv\)-classes which partition the \(\equiv_e\)-class of \(u_{i+1}\). We must insure that for \(A \not= A'\), the fresh elements added into \([A']\) are disjoint from the fresh elements added into \([A]\). This is needed to arrange that neither \([A]\) nor \([A']\) will be a subset of the other.

This completes the definition of the model. We say a few words about why requirements \((\alpha)-(\gamma)\) are met. First, and easy induction on \(i\) shows that if \(j < i\), then \(|[u_j]| < |[u_i]|\). The point is that \(|[u_j]| \leq |X_i| < |[u_i]|\). The argument for \((\beta)\) is the same as in the proof of Theorem 4.2. For that matter, the proof of \((\alpha)\) is also essentially the same. The point is that when \(A \equiv_e B\) and \(A \not= B\), then \([A]\) and \([B]\) each contain a point not in the other.

For \((\gamma)\), suppose that \(A \leq_e B\). Let \(i \leq j\) be such that \(A \equiv_e u_i\) and \(B \equiv u_j\). The construction arranged \([A]\) and \([B]\) be disjoint except for the case that \(A \uparrow B\).

So this verifies that \((\alpha)-(\gamma)\) hold. We would like to conclude that \(M \models \Gamma\), but there is one last point: \((\gamma)\) appears to be a touch too weak. We need to know that \([A] \cap [B] \not= \emptyset\) iff \(A \uparrow B\) (without assuming \(A \leq_e B\)). But either \(A \leq_e B\) or \(B \leq_e A\) by our last axiom. So we see that indeed \([A] \cap [B] \not= \emptyset\) iff \(A \uparrow B\). \(\square\)

The next step in this direction would be to consider \(\text{At least as many} \; x \; \text{as} \; y\) are \(z\).

### 4.2. Digression: Most

The semantics of Most is that Most \(x\) are \(y\) are that this is true iff \(|[x] \cap [y]| > \frac{1}{2} |[x]|\). So if \([x]\) is empty, then Most \(x\) are \(y\) is false.

As an example of what is going on, consider the following. Assume that All \(x\) are \(z\), All \(y\) are \(z\), Most \(z\) are \(y\), and Most \(y\) are \(x\). Does it follow that Most \(x\) are \(y\)? As it happens, the conclusion does not follow. One can take \(x = \{a, b, c, d, e, f, g\}\), \(y = \{e, f, g, h, i\}\), and \(z = \{a, b, c, d, e, f, g, h, i\}\). Then \(|x| = 7\), \(|y| = 5\), \(|z| = 9\), \(|y \cap z| = 5 > 9/5\), \(|x \cap y| = 3 > 5/2\), but \(|x \cap y| = 3 < 7/2\). (Another counter-model: let \(x = \{1, 2, 4, 5\}\), \(y = \{1, 2, 3\}\), and \(z = \{1, 2, 3, 4, 5\}\). Then \(|y \cap z| = 3 > 5/2\), \(|y \cap x| = 2 > 3/2\), but \(|x \cap y| = 2 \not= 4/2\).)

On the other hand, the following is a sound rule:

\[
\begin{array}{c}
\text{All } u \text{ are } x & \text{Most } x \text{ are } v & \text{All } v \text{ are } y & \text{Most } y \text{ are } u \\
\hline
\text{Some } u \text{ are } v
\end{array}
\]
4. CARDINALITY COMPARISONS

Here is the reason for this. Assume our hypotheses and also that towards a contradiction that \( u \) and \( v \) were disjoint. We obviously have \(|v| \geq |x \cap v|\), and the second hypothesis, together with the disjointness assumption, tells us that \(|x \cap v| > |x \cap u|\).

By the first hypothesis, we have \(|x \cap u| = |u|\). So at this point we have \(|v| > |u|\).

But the last two hypotheses similarly give us the opposite inequality \(|u| > |v|\). This is a contradiction.

At the time of this writing, I do not have a completeness result for \( \mathcal{L}(\text{all, some, most}) \).

The best that is known is for \( \mathcal{L}(\text{some, most}) \). The rules are are shown in Figure 4.3.

We study these on top of the rules in Figure 2.2.

**PROPOSITION 4.4** The following two axioms are complete for Most.

\[
\begin{align*}
\text{Most } x & \text{ are } y \\
\text{Most } x & \text{ are } x \\
\text{Most } y & \text{ are } y
\end{align*}
\]

Moreover, if \( \Gamma \subseteq \mathcal{L}(\text{most}), \ x \neq y \), and \( \Gamma \not\models \text{Most } x \text{ are } y \), then there is a model \( M \) of \( \Gamma \) which falsifies Most \( x \) are \( y \) in which all sets of the form \([u] \cap [v]\) are nonempty, and \(|M| \leq 5\).

**PROOF** Suppose that \( \Gamma \not\models \text{Most } x \text{ are } y \). We construct a model \( M \) which satisfies all sentences in \( \Gamma \), but which falsifies Most \( x \) are \( y \). There are two cases. If \( x = y \), then \( x \) does not occur in any sentence in \( \Gamma \). We let \( M = \{\ast\}, [x] = \emptyset \), and \([y] = \{\ast\} \) for \( y \neq x \).

The other case is when \( x \neq y \). Let \( M = \{1, 2, 3, 4, 5\}, [x] = \{1, 2, 4, 5\}, [y] = \{1, 2, 3\} \), and for \( z \neq x \) and \( y \), \([z] = \{1, 2, 3, 4, 5\} \). Then the only statement in Most which fails in the model \( M \) is Most \( x \) are \( y \). But this sentence does not belong to \( \Gamma \). Thus \( M \models \Gamma \).

**THEOREM 4.5** The rules in Figure 4.3 together with the first two rules in Figure 2.2 are complete for \( \mathcal{L}(\text{some, most}) \). Moreover, if \( \Gamma \not\models \varphi \), then there is a model \( M \models \Gamma \) with \( M \not\models \varphi \), and \(|M| \leq 6\).

**PROOF** Suppose \( \Gamma \not\models \varphi \), where \( \varphi \) is Some \( x \) are \( y \). If \( x = y \), then \( \Gamma \) contains no sentence involving \( x \). So we may satisfy \( \Gamma \) and falsify \( \varphi \) in a one-point model, by setting \([x] = \emptyset \) and \([z] = \{\ast\} \) for \( z \neq x \).

We next consider the case when \( x \neq y \). Then \( \Gamma \) does not contain \( \varphi \), Some \( y \) are \( x \), Most \( x \) are \( y \), or Most \( y \) are \( x \). And for all \( z \), \( \Gamma \) does not contain both Most \( z \) are \( x \) and Most \( z \) are \( y \). Let \( M = \{1, 2, 3, 4, 5, 6\} \), and consider the subsets \( a = \{1, 2, 3\} \), \( b = \{1, 2, 3, 4, 5\}, c = \{2, 3, 4, 5, 6\}, \) and \( d = \{4, 5, 6\} \). Let \([x] = a \) and \([y] = d \), so that \( M \not\models \varphi \). For \( z \) different from \( x \) and \( y \), if \( \Gamma \) does not contain Most \( z \) are \( x \), let \([z] = c \). Otherwise, \( \Gamma \) does not contain Most \( z \) are \( y \), and so we let \([z] = b \). For all these \( z \), \( M \) satisfies whichever of the sentences Most \( z \) are \( x \) and Most \( z \) are \( y \) (if either) which belong to \( \Gamma \). \( M \) also satisfies all sentences Most \( x \) are \( z \) and Most \( y \) are \( z \).
y are z, whether or not these belong to Γ. It also satisfies Most u are u for all u. Also, for z, z' each different from both x and y, M |= Most z are z'. Finally, M satisfies all sentences Some u are v except for u = x and y = v (or vice-versa). But those two sentences do not belong to Γ. The upshot is that M |= Γ but M |= ϕ.

Up until now in this proof, we have considered the case when ϕ is Some x are y. We turn our attention to the case when ϕ is Most x are y. Suppose Γ ⊬ ϕ. If x = y, then the second rule of Figure 4.3 shows that Γ ⊬ Some x are x. So we take M = ∅ and take [x] = ∅ and for y ≠ x, [y] = M. It is easy to check that M |= Γ.

Finally, if x ≠ y, we clearly have Γmost ⊬ ϕ. Proposition 4.4 shows that there is a model M |= Γmost which falsifies ϕ in which all sets of the form [u] ∩ [v] are nonempty. So all Some sentences hold in M. Hence M |= Γ.

4.3. ON THE NUMERICAL SYLLOGISTIC

I hope to add Ian Pratt-Hartmann’s results in [27] on the complexity of reasoning with the numerical syllogistic, and also his result in [28] that there is no finite syllogistic system for it.

Source for this chapter. The material in this chapter comes from Moss [18].
CHAPTER 5

Verbs: \( \mathcal{R} \)

At this point, we turn to logics with verbs. The basic goal is to have a logical system in which we may represent a valid argument such as

\[
\begin{align*}
\text{Every porter recognizes every porter} \\
\text{No quarterback recognizes any quarterback} \\
\text{No porter is a quarterback}
\end{align*}
\]

(5.1)

As in our previous work, we adopt as minimal a syntax as needed. In fact, the English sentences of interest are listed below, using \( p \) and \( q \) for nouns and \( r \) for verbs:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Symbolic Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>All ( p ) are ( q )</td>
<td>( \forall(p, q) )</td>
</tr>
<tr>
<td>Some ( p ) are ( q )</td>
<td>( \exists(p, q) )</td>
</tr>
<tr>
<td>All ( p ) don't ( r ) all ( q )</td>
<td>( \forall(p, \forall(q, r)) )</td>
</tr>
<tr>
<td>Some ( p ) don't ( r ) any ( q )</td>
<td>( \forall(p, \exists(q, r)) )</td>
</tr>
<tr>
<td>All ( p ) don't ( r ) some ( q )</td>
<td>( \forall(p, \exists(q, r)) )</td>
</tr>
<tr>
<td>Some ( p ) don't ( r ) some ( q )</td>
<td>( \forall(p, \exists(q, r)) )</td>
</tr>
</tbody>
</table>

We aim to have a formal language which looks like the sentences you see above. We also have listed the translations of the English sentences into our language \( \mathcal{R} \).

Here is how we build the syntax of \( \mathcal{R} \): We start with one collection \( \mathcal{P} \) of unary atoms (for nouns), just as we did in Chapter 2. But we also start with another collection, \( \mathcal{R} \), of another of binary atoms. We use these for transitive verbs.

Continuing, we use a complement operation on both unary and binary atoms. As in Chapter 3, we understand this operation to be involutive, so \( \overline{p} \) and \( p \) are taken to be identical, as are \( \overline{r} \) and \( r \). We call the items \( p, \overline{p}, r \) and \( \overline{r} \) literals.

We next introduce set terms by the following grammar:

<table>
<thead>
<tr>
<th>Set Terms</th>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \forall(p, r) )</td>
<td>( p )</td>
<td>( \overline{p} )</td>
</tr>
<tr>
<td>( \exists(p, r) )</td>
<td>( \forall(p, r) )</td>
<td>( \exists(p, r) )</td>
</tr>
</tbody>
</table>

Four of these are new to us, and here is how they are read:

- \( \forall(p, r) \): those who \( r \) all \( p \)
- \( \exists(p, r) \): those who \( r \) some \( p \)
- \( \forall(p, \overline{r}) \): those who fail-to-\( r \) all \( p \)
- \( \exists(p, \overline{r}) \): those who fail-to-\( r \) some \( p \)
- \( \forall(p, \overline{q}) \): those who fail-to-\( r \) all \( p \)
- \( \exists(p, \overline{q}) \): those who don’t \( r \) some \( p \)
5. VERBS: $\mathcal{R}$

<table>
<thead>
<tr>
<th>expression</th>
<th>variables</th>
<th>syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>unary atom</td>
<td>$p, q$</td>
<td>$p</td>
</tr>
<tr>
<td>binary atom</td>
<td>$r$</td>
<td>$\neg p</td>
</tr>
<tr>
<td>set term</td>
<td>$c, d$</td>
<td>$\exists(p, r)</td>
</tr>
<tr>
<td>$\mathcal{R}$ sentence</td>
<td>$\varphi$</td>
<td>$\forall(p, c)</td>
</tr>
<tr>
<td>$\mathcal{R}^1$ sentence</td>
<td>$\varphi$</td>
<td>$\forall(p, c)</td>
</tr>
</tbody>
</table>

Figure 5.1. The syntax of the languages $\mathcal{R}$ and $\mathcal{R}^1$.

We use letters like $c$ and $d$ for set terms.

Finally, we have two kinds of sentences: $\forall(p, c)$, and $\exists(p, c)$, where $p$ is a unary atom, and $c$ is a set term. This completes the syntax of the language which we call $\mathcal{R}$.

The language $\mathcal{R}^1$. This is a larger language which, loosely speaking, allows complemented atoms $\neg p$ to be subject noun phrases of sentences. More precisely, sentences such as $\exists(p, p)$ and $\forall(p, \exists(c, r))$ belong to $\mathcal{R}^1$ but not to $\mathcal{R}$. We shall not be so concerned with this larger language $\mathcal{R}^1$ in this section, and we mention it only to alert you that it is coming. We shall see an important negative result on $\mathcal{R}^1$ in Section 6.1.1, and we shall return to this language in Chapter 8.

We summarize the syntax of both languages in Figure 5.1.

5.0.1. Semantics. The natural interpretation of verbs is as relations on the universe, that is, sets of pairs of individuals. This leads to the following set of definitions.

**Definition 5.1** A model $M$ for $\mathcal{R}$ is a set $M$, together with an interpretation $\llbracket p \rrbracket \subseteq M$ for each noun $p \in P$ and an interpretation $\llbracket r \rrbracket \subseteq M^2$ for each verb $r$. We interpret literals $p$ and $r$ using complements:

$$\llbracket \neg p \rrbracket = M \setminus \llbracket p \rrbracket \quad \llbracket \forall(p, c) \rrbracket = \{ m \in M : \text{for all } n \in \llbracket p \rrbracket, (m, n) \in \llbracket c \rrbracket \}$$

$$\llbracket \exists(p, c) \rrbracket = \{ m \in M : \text{for some } n \in \llbracket p \rrbracket, (m, n) \in \llbracket c \rrbracket \}$$

Finally, we have the definition of truth in a model:

$M \models \forall(p, c) \iff \llbracket p \rrbracket \subseteq \llbracket c \rrbracket$

$M \models \exists(p, c) \iff \llbracket p \rrbracket \cap \llbracket c \rrbracket \neq \emptyset$

Finally, we have definitions of semantic consequence relation $\Gamma \models \varphi$, just as we have seen it for other logical languages.

**Example 5.2** We consider a simple case, with one unary atom $p$, and one binary atom $s$. Consider the following model. We set $M = \{w, x, y, z\}$, and $\llbracket p \rrbracket = \ldots$
An easy induction shows that \( [\forall c, x, y] \). For the relation symbol, \( s \), we take the arrows below:

\[
\begin{array}{ccc}
w & \rightarrow & x \\
& \uparrow & \\
y & \leftrightarrow & z \\
& \downarrow & \\
& \leftarrow & \end{array}
\]

For example, \( [\forall] = \{ z \} \), \( [\forall(p, s)] = \emptyset \), \( [\exists(p, s)] = M \), and \( [\exists(p, \overline{s})] = M \) also. Here are some \( \mathcal{R} \)-sentences true in \( M \): \( \exists(p, p) \) (but note that \( \exists(p, \overline{p}) \) is not in the fragment \( \mathcal{R} \)), and also \( \forall(p, \exists(p, s)) \).

Remark Sentences like “All porters recognize some quarterback” are ambiguous in English. In \( \mathcal{R} \), we can represent the subject wide scope reading: every porter has the property of recognizing some quarterback or other. It is not possible to represent the object wide scope reading, where there is one particular quarterback who every porter recognizes.

5.0.2. Translation of \( L \) into Boolean modal logic. We shall write \( \hat{\mathcal{R}} \) for the following version of Boolean modal logic. \( \hat{\mathcal{R}} \) has each \( p \in \mathcal{P} \) as an atomic proposition, and it has two modal operators, \( \Box_s \) and \( \Diamond_s \), one for each \( s \in \mathcal{R} \). The syntax of \( \mathcal{R} \) is given by

\[
\varphi := p | \neg \varphi | \varphi \land \psi | \Box_s \varphi | \Diamond_s \varphi
\]

The language is interpreted on the same kind of structures that we have been using for \( \mathcal{R} \). Then \( [p] \) is given for all atoms \( p \), and we also set \( [\neg \varphi] = M \setminus [\varphi] \), \( [\varphi \land \psi] = [\varphi] \cap [\psi] \), and

\[
[\Box_s \varphi] = \{ x \in M : \text{for all } y \text{ such that } [s](x, y), y \in [\varphi] \}
\]

\[
[\Diamond_s \varphi] = \{ x \in M : \text{for all } y \text{ such that not } [s](x, y), y \in [\varphi] \}
\]

We write \( \Gamma \models \varphi \) to mean that for all structures \( M \) and all \( x \in M \), if \( x \in [\psi] \) for all \( \psi \in \Gamma \), then again \( x \in [\varphi] \).

Let \( \mathcal{R}_0 \) be the set of sentences of \( \mathcal{R} \) which do not involve constants. We translate \( \mathcal{R}_0 \) into \( \hat{\mathcal{R}} \). First, for each set term \( c \), we define a sentence \( c^* \) of \( \hat{\mathcal{R}} \). The definition is: \( p^* = p, \overline{p}^* = \neg p \),

\[
\forall(c, s)^* = \Diamond_s \neg c^* \quad \forall(c, \overline{s})^* = \Box_s \neg c^*
\]

\[
\exists(c, s)^* = \neg \Box_s \neg c^* \quad \exists(c, \overline{s})^* = \neg \Diamond_s \neg c^*
\]

An easy induction shows that \( [c] = [c^*] \) for all set terms \( c \). Then we translate \( \forall(c, d) \) to \( \forall(c, d)^* \) and \( \exists(c, d) \) to \( \exists(c, d)^* \):

\[
\forall(c, d)^* = \Box_c (c^* \rightarrow d^*) \land \Diamond_c (c^* \rightarrow d^*)
\]

\[
\exists(c, d)^* = \neg \forall(c, \overline{d})^*
\]

where \( s \) is an arbitrary element of \( \mathcal{R} \). Then for all sentences \( \varphi \) of \( \mathcal{R}_0 \), and all models \( M \),

\[
M \models \varphi \iff [\varphi^*] = M \iff [\varphi^*] \neq \emptyset.
\]

It follows easily from this that for \( \Gamma \cup \{ \varphi \} \) a set of \( \mathcal{R}_0 \) sentences, \( \Gamma \models \varphi \) in the semantics of \( \mathcal{R} \) if and only if \( \Gamma^* \models \varphi^* \) in the semantics of \( \hat{\mathcal{R}} \).
5.1. A Logic for $\mathcal{R}$

This section presents our logical system $\mathcal{R}$ for $\mathcal{R}$ and shows that it is sound and complete. Our rules appear in Figure 5.2. We remind the reader that $p$ and $q$ range over unary atoms, $c$ over $c$-terms, and $t$ over binary literals.

Rules (D1), (D2), (D3), (B), (T) and (I) are natural generalizations of their namesakes in $\mathcal{S}$. In contrast, (∀∀), (∃∃), (∀∃) and (II) express genuinely relational logical principles. In some settings, these last rules are called monotonicity principles.

In addition to syllogistic rules, $\mathcal{R}$ contains the rule of *reductio ad absurdum* (RAA), whereby one derives $\varphi$ from $\Gamma$ by temporarily adding the “negation” of $\varphi$ to $\Gamma$, thereby obtaining $\Gamma \cup \{\neg \varphi\}$, and then deriving a contradiction from this larger set $\Gamma \cup \{\neg \varphi\}$. The idea is that This rule (RAA) is *not the same as ex falso quodlibet*, the rule we have called (X) in Chapter 3. To see why, we mention the semantic justifications for both rules.

But before we can do this, let us clarify what we mean by “negation.” We do not have a negation symbol in $\mathcal{R}$, but we do have a *semantic* negation. It is defined
as follows:

<table>
<thead>
<tr>
<th>expression</th>
<th>syntax</th>
<th>negation</th>
</tr>
</thead>
<tbody>
<tr>
<td>set term (c)</td>
<td>(p)</td>
<td>(\overline{p})</td>
</tr>
<tr>
<td>(\overline{p})</td>
<td>(p)</td>
<td>(\overline{p})</td>
</tr>
<tr>
<td>(\exists(p, r))</td>
<td>(\forall(p, \overline{r}))</td>
<td>(\exists(p, \overline{r}))</td>
</tr>
<tr>
<td>(\forall(p, r))</td>
<td>(\exists(p, \overline{r}))</td>
<td>(\forall(p, \overline{r}))</td>
</tr>
<tr>
<td>(\exists(p, \overline{r}))</td>
<td>(\forall(p, r))</td>
<td>(\exists(p, r))</td>
</tr>
<tr>
<td>(\forall(p, \overline{r}))</td>
<td>(\exists(p, r))</td>
<td>(\forall(p, r))</td>
</tr>
</tbody>
</table>

\(\mathcal{R}\) sentence \(\varphi\) 

\[
\begin{array}{c}
\forall(p, c) \\
\exists(p, c) \\
\end{array}
\]

\(\overline{\varphi}\) 

\(\overline{\varphi}\)

\(\overline{\varphi}\)

Note that \(\overline{\varphi} = p\), \(\overline{c} = c\) and \(\overline{\overline{\varphi}} = \varphi\).

Here is why we call \(\varphi\) and \(\overline{\varphi}\) semantic negations: for all models \(M\), \(M \models \varphi\) iff \(M \not\models \overline{\varphi}\).

We also define \(\bot\) to be any contradiction. In \(\mathcal{R}\), this means a sentence of the form \(\exists(p, \overline{p})\).

Here are the ideas behind (RAA) and (X):

1. \(\Gamma \cup \{\overline{\varphi}\} \models \bot\), then \(\Gamma \models \varphi\).
2. \(\Gamma \models \bot\), then \(\Gamma \models \varphi\).

The (RAA) justification changes the assumptions in the derivation from \(\Gamma \cup \{\overline{\varphi}\}\) to \(\Gamma\). This is the exact difference between \textit{reductio ad absurdum} and \textit{ex falso quodlibet}.

We must incorporate this observation into our proof system, and we do so in (RAA). We now can display this rule in natural-deduction-style:

\[
\frac{\varphi}{\overline{\varphi}} \quad \text{RAA}
\]

**Definition 5.3** A proof tree over \(\Gamma\) is a pair \((\mathcal{T}, \text{Can})\), where finite tree \(\mathcal{T}\) whose nodes are labeled with sentences, and \(\text{Can}\) is a set of labeled leaves of \(\mathcal{T}\) called the canceled leaves. Each node \(n\) in the tree must satisfy one of the following conditions:

1. \(n\) is a leaf labeled by an element of \(\Gamma\).
2. \(n\) comes from its parent(s) by an application of a rule other than (RAA).
3. \(n \in \text{Can}\), and there is some node \(m\) on the path from \(n\) to the root such that the parent of \(m\) is labeled \(\bot\), and the label of \(m\) is the semantic negation of the label of \(n\).

We write \(\Gamma \vdash \varphi\) if there is a proof tree \(\mathcal{T}\) with \(\varphi\) at the root whose uncanceled leaves all belong to \(\Gamma\).

**Example 5.4** Here is a derivation showing that

\[
\forall(x, \overline{x}) \vdash \forall(y, \forall(x, r))
\]

In words, if there are no \(x\)s, then all \(y\)’s have any relation whatsoever to all of them. Note that we cannot simply use the rule

\[
\frac{\forall(p, \overline{p})}{\forall(p, c)}
\]
Instead, we use RAA:

\[
\frac{\exists(y, \exists(x, r))}{\exists(x, x)} \quad \forall(x, x) \\
\frac{\exists(x, x)}{\exists(x, r)} \quad \text{(RAA)}
\]

This example also shows that we indicate canceled leaves using bracketing, and we also use numerical superscripts to tell which application of (RAA) has canceled which leaves.

**Example 5.5** Here is a derivation showing that

\[
\forall(x, \forall(y, r)), \forall(p, y), \exists(p, q) \vdash \forall(x, \exists(y, r))
\]

\[
\frac{\exists(p, q)}{\forall(p, y)} \quad \text{(D1)}
\]

\[
\frac{\forall(x, \exists(y, r))}{\forall(x, \exists(p, r))} \quad \forall(x, \exists(p, r)) \quad \forall(p, y) \quad \text{(∀∃)}
\]

**Example 5.6** Here is a formal proof showing that

\[
\forall(x, r) \vdash \forall(y, \forall(x, r))
\]

In words, if there are no \(x\)'s, then all \(y\)'s have any relation whatsoever to all of them.

Note that this does not follow from the rule (A). But here is a derivation:

\[
\frac{\exists(y, \exists(x, r))}{\exists(x, x)} \quad \text{(II)}
\]

\[
\frac{\exists(x, x)}{\exists(x, r)} \quad \text{(D1)}
\]

\[
\frac{\exists(x, r)}{\forall(y, \forall(x, r))} \quad \text{(RAA)}
\]

As we shall see below in Theorem 6.4, the special status of (RAA) is essential: indirect syllogistic systems are in general more powerful than direct syllogistic systems.

Although (RAA) may be used at any point in a derivation, our proof system in this section has the extra property that if \(\Gamma \vdash \varphi\) using (RAA), then there is a proof using (RAA) at most once. In these notes we are not going to be concerned with this stronger property, and for more on it see [29].

**5.1.1. Extending consistent sets to consistent and complete sets.** At this point, we mention a few features of the logical system which play a role in our completeness proof.

**Lemma 5.7** Suppose that \(\Gamma \cup \{\varphi\} \vdash \psi\) and also \(\Gamma \cup \{\overline{\varphi}\} \vdash \psi\). Then \(\Gamma \vdash \psi\).

**Proof** The hypotheses imply that \(\Gamma \cup \{\varphi\} \vdash \bot\). Also \(\Gamma \cup \{\overline{\varphi}\} \vdash \bot\), and by (RAA) we also have \(\Gamma \cup \{\overline{\varphi}\} \vdash \varphi\). Let \(T_1\) be a proof tree showing that \(\Gamma \cup \{\varphi, \overline{\varphi}\} \vdash \bot\), let \(T_2\) be a proof tree showing that \(\Gamma \cup \{\overline{\varphi}\} \vdash \varphi\). Take \(T_1\) and replace every leaf labeled \(\varphi\) with a copy of \(T_2\). This gives a proof tree with root \(\bot\) and whose leaves are labeled in \(\Gamma \cup \{\overline{\varphi}\} \vdash \bot\). Using (RAA) again, we see that \(\Gamma \vdash \psi\). □
5.1. A Logic for \( \mathcal{R} \)

**Definition** 5.8 A set \( \Gamma \) of \( \mathcal{R} \)-sentences is **inconsistent** if \( \Gamma \vdash \bot \). Otherwise, \( \Gamma \) is **consistent**. \( \Gamma \) is **complete** if for all \( \varphi \), either \( \varphi \in \Gamma \) or \( \varphi \not\in \Gamma \).

**Lemma** 5.9 Every consistent set \( \Gamma \) has a complete and consistent superset \( \Gamma^* \).

**Proof** We are going to use Zorn's Lemma. Let \( \mathcal{C} \) be the set of consistent \( \Delta \supseteq \Gamma \), ordered by inclusion. A **chain** in \( \mathcal{C} \) is a set \( X \) of elements of \( \mathcal{C} \) with the property that if \( X \) contains both \( \Delta_1 \) and \( \Delta_2 \), then either \( \Delta_1 \subseteq \Delta_2 \) or \( \Delta_2 \subseteq \Delta_1 \). To use Zorn's Lemma, we must check that every chain \( X \) in \( \mathcal{C} \) has an upper bound. This is immediate: we take the union \( \bigcup X \). The important point is that this is a consistent set, and this point follows from the fact that proofs are finite.

Zorn's Lemma applies and gives us a maximal element \( \Gamma^* \) of \( \mathcal{C} \). \( \Gamma^* \) is thus a consistent superset of \( \Gamma \). We need only check that it is complete. For if not, suppose that neither \( \varphi \) nor \( \varphi \) belong to \( \Gamma^* \). By maximality, both \( \Gamma^* \cup \{ \varphi \} \vdash \bot \) and \( \Gamma^* \cup \{ \varphi \} \vdash \bot \). By Lemma 5.7, \( \Gamma^* \vdash \bot \). This contradicts the consistency of \( \Gamma^* \), and so we conclude that \( \Gamma^* \) is indeed complete. \( \square \)

**Remark** Lemmas 5.7 and 5.9 have nothing to do with the particular proof system, and indeed they hold for logical systems which have (RAA). So versions of these results hold for all of the logical systems in the rest of these notes.

5.1.2. Model construction.

**Theorem** 5.10 For \( \Gamma \cup \{ \varphi \} \subseteq \mathcal{R} \), \( \Gamma \models \varphi \) iff \( \Gamma \vdash \varphi \) in \( \mathcal{R} \).

**Proof** The soundness is an easy induction on derivations. For the completeness, we need only show that a consistent set \( \Gamma \) is satisfiable. (In more detail, suppose we know the general fact that every consistent set is satisfiable. Suppose that \( \Gamma \models \varphi \). Then \( \Gamma \cup \{ \varphi \} \) has no models, so by the general fact, \( \Gamma \cup \{ \varphi \} \) is inconsistent. Thus \( \Gamma \cup \{ \varphi \} \vdash \bot \), and so by (RAA), \( \Gamma \vdash \varphi \).) By Lemma 5.9, we may assume that \( \Gamma \) is \( \mathcal{R} \)-complete.

For such a consistent and \( \mathcal{R} \)-complete set \( \Gamma \), we shall define a model \( M = M(\Gamma) \) as follows: we let

\[
M = \{ x_1, x_2 : \Gamma \vdash \exists(x, x) \} \cup \{ \{ p, q \} : p \neq q \text{ and } \Gamma \vdash \exists(p, q) \}.
\]

We assume the union in (5.4) is disjoint, and we call the elements \( \{ p, q \} \) **pair-elements**. So we have two copies of every variable \( x \) such that \( \Gamma \) entails the existence of \( x \), and also pair elements \( \{ p, q \} \) corresponding to sentences of the form \( \exists(p, q) \) which are provable from \( \Gamma \) and such that \( p \neq q \). Our semantics will insure that the pair-element \( \{ p, q \} \) belongs to \( \lbrack [p] \cap [q] \rbrack \), and so this element will witness the truth of \( \exists(p, q) \) in the model which we are constructing.

---

1Please do not confuse the completeness of a set of sentences with the completeness of the logical system under discussion.

2If you are not comfortable, it is also possible to prove the result in the case that the overall language is countable by a step-by-step procedure. Here is a sketch. One lists the sentences in the language in a list as \( \varphi_0, \varphi_1, \ldots, \varphi_n, \ldots \). One also constructs an infinite sequence \( \Gamma = \Gamma_0 \subseteq \cdots \Gamma_n \subseteq \cdots \) of sets of sentences. \( \Gamma_{n+1} \) is either \( \Gamma_n \cup \{ \varphi_n \} \) or else \( \Gamma_n \cup \{ \varphi_n \} \), whichever is consistent. One must be consistent, lest \( \Gamma_n \) be inconsistent by Lemma 5.7. And then \( \bigcup_n \Gamma_n \) would be as desired.
For the binary atoms $r$ (5.6) $w$ (5.5) the semantics [ ] we shall need a certain set shown in Figure 5.3.

5. VERBS: $R$̸⊢∀ (b) when $\Gamma$̸⊢∀ (d) when $\Gamma$̸⊢∀ (5.11) Proposition

The unary atoms $x$ are interpreted in our models as follows:

$w_i \in [x]$ \iff $\Gamma \vdash \forall (w, x)$

$\{p, q\} \in [x]$ \iff $\Gamma \vdash \forall (p, x)$, or $\Gamma \vdash \forall (q, x)$

For the binary atoms $r$, we first need a definition. For two unary atoms $x$ and $y$ (possibly identical) we shall need a certain set shown in Figure 5.3.

$R^\Gamma_{x,y} \subseteq \{x_1, x_2\} \times \{y_1, y_2\}$

The semantics $[r]$ in $M(\Gamma)$ is then given as follows:

$x_1[r]y_j$ \iff $x_i \rightarrow y_j$ according to $R^\Gamma_{x,y}$ in Figure 5.3

$\{x, y\}[r]w_2$ \iff $\Gamma \vdash \forall (x, \forall (w, r))$ or $\Gamma \vdash \forall (x, \forall (y, r))$

$\{x, y\}[r]w_1$ \iff $\{x, y\}[r]w_2$, or $\Gamma \vdash \forall (x, \exists (w, r))$, or $\Gamma \vdash \forall (y, \exists (w, r))$

$u_1[r]\{x, y\}$ \iff $\Gamma \vdash \forall (u, \forall (x, r))$ or $\Gamma \vdash \forall (u, \forall (y, r))$

$u_2[r]\{x, y\}$ \iff $u_1[r]\{x, y\}$, or $\Gamma \vdash \exists (u, \forall (x, r))$, or $\Gamma \vdash \exists (u, \forall (y, r))$

$\{x, y\}[r]\{p, q\}$ \iff $\Gamma \vdash \forall (x, \forall (p, r))$, or $\Gamma \vdash \forall (x, \forall (q, r))$, or $\Gamma \vdash \forall (y, \forall (p, r))$, or $\Gamma \vdash \forall (y, \forall (q, r))$

**Proposition 5.11** Concerning the relation $[r]$:  
(1) $x_1[r]y_2$ \iff $\Gamma \vdash \forall (x, \forall (y, r))$.
(2) $x_1[r]y_1$ \iff $\Gamma \vdash \forall (x, \forall (y, r))$ or $\Gamma \vdash \forall (x, \exists (y, r))$.
(3) $x_2[r]y_2$ \iff $\Gamma \vdash \forall (x, \forall (y, r))$ or $\Gamma \vdash \exists (x, \forall (y, r))$.
(4) $x_2[r]y_1$ \iff for some $i$ and $j$, $x_i[r]y_j$.

**Figure 5.3.** The relation $R^\Gamma_{x,y}$.
LEMMA 5.12. Let $\Gamma$ be an arbitrary set of $\mathcal{R}$-sentences. Let $\varphi$ be a positive sentence. If $\Gamma \vdash \varphi$, then $\mathcal{M}(\Gamma) \models \varphi$.

Proof. We argue by cases on $\varphi$.

Case 1: $\varphi$ is $\forall(x, y)$. If $z_i \in [x]$, then $z \leq x$. By monotonicity, $z \leq y$. So $z_i \in [y]$. If $\{p, q\} \in [x]$, then without loss of generality $\forall(p, x)$. Again, we see that $\{p, q\} \in [y]$.

Case 2: $\varphi$ is $\exists(x, y)$. This time $\{x, y\}$ is an element of our model. Our logic contains the identity axioms All $x$ are $x$. By our semantics, $\{x, y\} \in [x] \cap [y]$. Thus the model overall satisfies Some $x$ are $y$.

Case 3: $\varphi$ is $\forall(x, \forall(y, r))$. Let $z_i \in [x]$ and $w_j \in [y]$; so we have $z \leq x$ and $w \leq y$. By monotonicity, $\Gamma \vdash \forall(z, \forall(w, r))$. So $z_i[r]w_j$ for $1 \leq i, j \leq 2$. We also must consider pair-elements $\{p, q\} \in [x]$. Without loss of generality, assume $\forall(p, x)$. By monotonicity, $\Gamma \vdash \forall(p, \forall(w, r))$. Again, our semantics insures that $\{p, q\}[r]w_j$ for all $j$. There are two other cases which are argued similarly; these show that if $z_i \in [x]$ and $\{u, w\} \in [y]$, then $z_i[r]\{u, w\}$; also, if $\{p, q\} \in [x]$ and $\{u, w\} \in [y]$, then $\{p, q\}[r]\{u, w\}$.

Case 4: $\varphi$ is $\forall(x, \exists(y, r))$. In this case, we can assume that $[x] \neq \emptyset$. That is, $\Gamma \vdash \exists(x, y)$ as well. We shall show that every element of $[x]$ is related to $y_1$. Let $z_i \in [x]$, so that $z \leq x$. By monotonicity, $\Gamma \vdash \forall(z, \exists(y, r))$. Then by Proposition 5.11, parts (2) and (4), we indeed have $z_i[r]y_1$ and $z_i[r]y_2$. Further, let $\{p, q\} \in [x]$. Without loss of generality, $p \leq x$. By monotonicity, $\Gamma \vdash \forall(p, \exists(y, r))$. Then by the definition of $[r]$, $\{p, q\}[r]y_1$.

Case 5: $\varphi$ is $\exists(x, \forall(y, r))$. By rule (I) of our logic, $\Gamma \vdash \exists(x, x)$. Let $w_j \in [y]$. Then $\Gamma \vdash \forall(y, w)$, hence $\Gamma \vdash \exists(x, \forall(w, r))$. By Proposition 5.11, parts (3) and (4), $x_2[r]w_1$, and also $x_2[r]w_2$. We must also consider pair-elements of $[y]$. Let $\{p, q\} \in [y]$ so that $\Gamma \vdash \exists(p, q)$; and assume $\Gamma \vdash \forall(p, y)$. By monotonicity, $\Gamma \vdash \exists(x, \forall(p, r))$. By construction, $x_2[r]\{p, q\}$. We conclude that $x_2$ is the required witness to $\exists(x, \forall(y, r))$.

Case 6: $\varphi$ is $\exists(x, \exists(y, r))$. Here both $\Gamma \vdash \exists(x, x)$ and also $\exists(y, y)$. By Proposition 5.11, part (4), $x_2[r]y_1$. So $\mathcal{M} \models \exists(x, \exists(y, r))$.

Lemma 5.13. Let $\Gamma$ be complete and consistent. Let $\varphi$ be a positive sentence. If $\mathcal{M}(\Gamma) \models \varphi$, then $\varphi \in \Gamma$.

Proof. We argue by cases on $\varphi$. In each case, we assume that $\mathcal{M}(\Gamma) \models \varphi$, and we then show $\Gamma \vdash \varphi$. Since $\Gamma$ is complete, we indeed have $\varphi \in \Gamma$.

One fact which we shall use frequently is that if $[x] \neq \emptyset$ in $\mathcal{M}(\Gamma)$, then $\Gamma \vdash \exists(x, x)$. For if $y_j \in [x]$, they by the structure of the model, $\Gamma \vdash \exists(y, y)$ and also $\forall(y, x)$. Similar considerations apply to a pair-element $\{u, w\} \in [x]$.

Case 1: $\varphi$ is $\forall(x, y)$. We may assume that $\Gamma \vdash \exists(x, x)$; if not, then $\Gamma \vdash \varphi$ using (A). And then the structure of the model easily tells us that $\Gamma \vdash \varphi$.

Case 2: $\varphi$ is $\exists(x, y)$. The argument is very close to what we do concerning $\exists(x, \exists(y, r))$ in Case 6 below.

Case 3: $\varphi$ is $\forall(x, \forall(y, r))$. By completeness, either $\Gamma \vdash \forall(x, \exists)$; or $\Gamma \vdash \forall(y, \exists)$; or else both $\Gamma \vdash \exists(x, x)$ and $\Gamma \vdash \exists(y, y)$. In the first case, $\Gamma \vdash \varphi$ using the rule (A). In the second case, $\Gamma \vdash \varphi$ as we have seen in Example 5.6. In the last, consider
We begin with a set $\Gamma$ which we assume to be complete and consistent; the goal is to build a model of $\Gamma$. Consider $M = M(\Gamma)$ as we defined it in (5.4), (5.5), and (5.6). By Lemma 5.12, $M$ satisfies the positive sentences in $\Gamma$. We claim that $M$ satisfies the negative sentences in $\Gamma$ as well. For suppose that $\psi$ is positive and $\psi$ belongs to $\Gamma$. If $M \models \psi$, we would have $M \models \psi$. By Lemma 5.12, $\Gamma \vdash \psi$. But then $\Gamma$ is inconsistent, a contradiction. The claim shown, we now see that $M \models \Gamma$. 

This concludes the proof. □

At this point we conclude the proof of the main result in this section, Theorem 5.10. We began with a set $\Gamma$ which we assume to be complete and consistent; the goal is to build a model of $\Gamma$. Consider $M = M(\Gamma)$ as we defined it in (5.4), (5.5), and (5.6). By Lemma 5.12, $M$ satisfies the positive sentences in $\Gamma$. We claim that $M$ satisfies the negative sentences in $\Gamma$ as well. For suppose that $\psi$ is positive and $\psi$ belongs to $\Gamma$. If $M \models \psi$, we would have $M \models \psi$. By Lemma 5.12, $\Gamma \vdash \psi$. But then $\Gamma$ is inconsistent, a contradiction. The claim shown, we now see that $M \models \Gamma$. □
5.2. There is no Finite Syllogistic Logic for $R$

In the previous section, we presented a logic for $R$. The reader might well wonder whether we need (RAA) or some such device, or whether the kinds of logics which we have previously considered were sufficient. Now is the time to answer this. We need to formulate exactly what we mean by a purely syllogistic system, and then prove that there are no purely syllogistic systems for $R$ which are finite, sound, and complete.

5.2.1. General Definitions on Syllogistic Proof Systems. Let $F$ be a syllogistic fragment. A derivation relation $\sim$ in $F$ is a subset of $P(F) \times F$, where $P(F)$ is the power set of $F$. For readability, we write $\Theta \mid \sim \theta$ instead of $\langle \Theta, \theta \rangle \in \sim$.

We say that $\sim$ is sound if $\Theta \mid \sim \theta$ implies $\Theta \models \theta$, and complete (for $F$) if $\Theta \models \theta$ implies $\Theta \mid \sim \theta$.

**Definition 5.14** Let $F$ be a syllogistic fragment, we have been calling a logical language in these notes. We employ the following terminology. A syllogistic rule (sometimes, simply: rule) in $F$ is a pair $\Theta/\theta$, where $\Theta$ is a finite set (possibly empty) of $F$-sentences, and $\theta$ an $F$-sentence. We call $\Theta$ the antecedents of the rule, and $\theta$ its consequent. The rule $\Theta/\theta$ is sound if $\Theta \models \theta$.

A substitution is a function $g = g_1 \cup g_2$, where $g_1 : P \rightarrow P$ and $g_2 : R \rightarrow R$. If $\theta$ is an $F$-formula, denote by $g(\theta)$ the $F$-formula which results by replacing any atom (unary or binary) in $\theta$ by its image under $g$, and similarly for sets of formulas. An instance of a syllogistic rule $\Theta/\theta$ is the syllogistic rule $g(\Theta)/g(\theta)$, where $g$ is a substitution.

A syllogistic proof system is a set of syllogistic rules.

We generally display rules in ‘natural-deduction’ style. For example,

\[
\begin{align*}
\forall(q,o) & \quad \exists(p,q) \\
\exists(p,o) & \quad \forall(q,\bar{o}) \quad \exists(p,q)
\end{align*}
\]

where $p$, $q$ and $o$ are unary atoms, are syllogistic rules in $S$, corresponding to the traditional syllogisms Darii and Ferio, respectively.

Syllogistic rules which differ only with respect to re-naming of unary or binary atoms will be informally regarded as identical, because they have the same instances. Thus, the letters $p$, $q$ and $o$ in (5.7) function, in effect, as variables ranging over unary atoms. It is often convenient to display syllogistic rules using variables ranging over other types of expressions, understanding that these are just more compact ways of writing finite collections of syllogistic rules in the official sense. For example, the two rules (5.7) may be more compactly written

\[
\begin{align*}
\forall(q,l) & \quad \exists(p,q) \\
\exists(p,l) & \quad \forall(q,o)
\end{align*}
\]

where $p$ and $q$ range over unary atoms, but $l$ ranges over unary literals.

Fix a syllogistic fragment $F$, and let $\mathcal{X}$ be a set of syllogistic rules in $F$. Define $\vdash_\mathcal{X}$ to be the smallest derivation relation in $F$ satisfying:

1. if $\theta \in \Theta$, then $\Theta \vdash_\mathcal{X} \theta$;
2. if $\{\theta_1, \ldots, \theta_n\}/\theta$ is a rule in $\mathcal{X}$, $g$ a substitution, $\Theta = \Theta_1 \cup \cdots \cup \Theta_n$, and $\Theta_i \vdash_\mathcal{X} g(\theta_i)$ for all $i$ ($1 \leq i \leq n$), then $\Theta \vdash_\mathcal{X} g(\theta)$.
It is simple to show that the derivation relation \( \vdash_X \) is sound if and only if each rule in \( X \) is sound.

Informally, we imagine chaining together instances of the rules in \( X \) to construct derivations, in the obvious way; and we refer to the resulting proof system as the direct syllogistic system defined by \( X \). We generally display derivations in natural-deduction style, as we have been doing throughout these notes.

**Theorem 5.15** There exists no finite set \( X \) of syllogistic rules in \( R \) such that \( \vdash_X \) is both sound and complete.

**Proof** Let \( X \) be any finite set of syllogistic rules for \( R \), and suppose \( \vdash_X \) is sound. We show that it is not complete. Since \( X \) is finite, fix \( n \in \mathbb{N} \) greater than the number of antecedents in any rule in \( X \).

Let \( p_1, \ldots, p_n \) be distinct unary atoms and \( r \) a binary atom. Let \( \Gamma \) be the following set of \( \mathcal{R} \)-formulas:

\[
\forall (p_i, \exists (p_{i+1}, r)) \quad (1 \leq i < n) \tag{5.8}
\]

\[
\forall (p_1, \forall (p_n, r)) \tag{5.9}
\]

\[
\forall (p, p) \quad (p \in \mathcal{P}) \tag{5.10}
\]

\[
\forall (p_i, \exists) \quad (1 \leq i < j \leq n) \tag{5.11}
\]

and let \( \gamma \) be the \( \mathcal{R} \)-formula \( \forall (p_1, \exists (p_n, r)) \). Observe that \( \Gamma \models \gamma \). To see this, let \( M \models \Gamma \). If \( p_1^M = \emptyset \), then trivially \( M \models \gamma \); on the other hand, if \( p_1^M \neq \emptyset \), a simple induction using formulas (5.8) shows that \( p_i^M \neq \emptyset \) for all \( 1 \leq i \leq n \), whence \( M \models \gamma \) by (5.9).

For \( 1 \leq i < n \), let \( \Delta_i = \Gamma \setminus \{ \forall (p_i, \exists (p_{i+1}, r)) \} \).

**Claim** If \( \varphi \in \mathcal{R} \) and \( \Delta_i \models \varphi \), then \( \varphi \in \Gamma \).

It follows from this claim that \( \Gamma \not\models \gamma \). For, since no rule of \( X \) has more than \( n - 1 \) antecedents, any instance of those antecedents contained in \( \Gamma \) must be contained in \( \Delta_i \) for some \( i \). Let \( \delta \) be the corresponding instance of the consequent of that rule. Since \( \vdash_X \) is sound, \( \Delta_i \models \delta \). By Claim 5.2.1, \( \delta \in \Gamma \). By induction on the number of steps in derivations, we see that no derivation from \( \Gamma \) leads to a formula not in \( \Gamma \). But \( \gamma \not\in \Gamma \).

**Proof of Claim** Certainly, \( \Delta_i \) has a model, for instance the model \( M_i \) given by:

\[
\begin{array}{c}
p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_i \rightarrow p_{i+1} \rightarrow \cdots \rightarrow p_n
\end{array}
\tag{5.12}
\]

Here, \( A = \{ p_1, \ldots, p_n \} \), \( p_j^{M_i} = \{ p_j \} \) for all \( j \) \( (1 \leq j \leq n) \), and \( r^{M_i} \) is indicated by the arrows. All other atoms (unary or binary) are assumed to have empty extensions. Note that there is no arrow from \( p_i \) to \( p_{i+1} \).

We consider the various possibilities for \( \varphi \) in turn and check that either \( \varphi \in \Gamma \) or there is a model of \( \Delta_i \) in which \( \varphi \) is false.

(i) \( \varphi \) is of the form \( \forall (p, p) \). Then \( \varphi \in \Gamma \) by (5.10).

(ii) \( \varphi \) is not of the form \( \forall (p, p) \), and involves at least one unary or binary atom other than \( p_1, \ldots, p_n, r \). In this case, it is straightforward to modify \( M_i \) so as to
5.3. AN ALTERNATIVE FOR THE POSITIVE FRAGMENT

We now present an alternative logic for the positive fragment of $\mathcal{R}$, the set of sentences with no complementation on either nouns or verbs. The fragment cannot have a finite axiomatization by purely syllogistic rules, and we have seen a logic reason! which (RAA) gives us the capability of handling the full $\mathcal{R}$. The point here is to go a different route, by proposing a logic with no (RAA), indeed not even with ex falso quodlibet, and instead with infinitely many rules. (However, the rules themselves are a regular set.)

Our logic is listed in Figure 5.4.

We write $x \sim y$ as a shorthand for a sequence of sentences such as

\[(5.13) \forall(x, \exists(c_1, r)), \forall(c_1, \exists(c_2, r)), \ldots, \forall(c_n, \exists(y, r)).\]
Figure 5.4. A logical system for the positive fragment of \( R \) which includes neither \textit{ex falso quodlibet} nor (RAA). The system is not axiomatizable using only finitely many rules.

So the last two rules in Figure 5.4 may be abbreviated thus:

\[
\begin{align*}
\exists(p, q) & \quad \forall(q, c) \\
\hline
\exists(p, c) & \quad \forall(p, c) \\
\end{align*}
\]

(B)

\[
\begin{align*}
\forall(p, q) & \quad \exists(p, c) \\
\hline
\exists(q, c) & \quad \forall(q, c) \\
\end{align*}
\]

(T)

\[
\begin{align*}
\forall(p, q) & \quad \exists(p, q, c) \\
\hline
\exists(q, q) & \quad \forall(q, q) \\
\end{align*}
\]

(II)

\[
\begin{align*}
\forall(p, q) & \quad \exists(q, t) \\
\hline
\exists(q, q') & \quad \forall(q, q') \\
\end{align*}
\]

(∃)

\[
\begin{align*}
\forall(p, q) & \quad \exists(q, t) \\
\hline
\exists(q', q) & \quad \forall(q', q) \\
\end{align*}
\]

(I)

\[
\begin{align*}
\forall(p, q) & \quad \exists(q, t) \\
\hline
\exists(q, q') & \quad \forall(q, q') \\
\end{align*}
\]

(∃)

\[
\begin{align*}
\forall(p, q) & \quad \exists(q, t) \\
\hline
\exists(q, q') & \quad \forall(q, q') \\
\end{align*}
\]

(∃)

\[
\begin{align*}
\forall(p, q) & \quad \exists(q, t) \\
\hline
\exists(q, q') & \quad \forall(q, q') \\
\end{align*}
\]

(∃)

\[
\begin{align*}
\forall(p, q) & \quad \exists(q, t) \\
\hline
\exists(q, q') & \quad \forall(q, q') \\
\end{align*}
\]

(∃)

\[
\begin{align*}
\forall(p, q) & \quad \exists(q, t) \\
\hline
\exists(q, q') & \quad \forall(q, q') \\
\end{align*}
\]

(∃)

\[
\begin{align*}
\forall(p, q) & \quad \exists(q, t) \\
\hline
\exists(q, q') & \quad \forall(q, q') \\
\end{align*}
\]

(∃)

\[
\begin{align*}
\forall(p, q) & \quad \exists(q, t) \\
\hline
\exists(q, q') & \quad \forall(q, q') \\
\end{align*}
\]

(∃)

\[
\begin{align*}
\forall(p, q) & \quad \exists(q, t) \\
\hline
\exists(q, q') & \quad \forall(q, q') \\
\end{align*}
\]

(∃)

\[
\begin{align*}
\forall(p, q) & \quad \exists(q, t) \\
\hline
\exists(q, q') & \quad \forall(q, q') \\
\end{align*}
\]

(∃)

\[
\begin{align*}
\forall(p, q) & \quad \exists(q, t) \\
\hline
\exists(q, q') & \quad \forall(q, q') \\
\end{align*}
\]

(∃)

\[
\begin{align*}
\forall(p, q) & \quad \exists(q, t) \\
\hline
\exists(q, q') & \quad \forall(q, q') \\
\end{align*}
\]

(∃)

\[
\begin{align*}
\forall(p, q) & \quad \exists(q, t) \\
\hline
\exists(q, q') & \quad \forall(q, q') \\
\end{align*}
\]

(∃)

\[
\begin{align*}
\forall(p, q) & \quad \exists(q, t) \\
\hline
\exists(q, q') & \quad \forall(q, q') \\
\end{align*}
\]

(∃)

\[
\begin{align*}
\forall(p, q) & \quad \exists(q, t) \\
\hline
\exists(q, q') & \quad \forall(q, q') \\
\end{align*}
\]

(∃)

\[
\begin{align*}
\forall(p, q) & \quad \exists(q, t) \\
\hline
\exists(q, q') & \quad \forall(q, q') \\
\end{align*}
\]

(∃)

\[
\begin{align*}
\forall(p, q) & \quad \exists(q, t) \\
\hline
\exists(q, q') & \quad \forall(q, q') \\
\end{align*}
\]

(∃)

\[
\begin{align*}
\forall(p, q) & \quad \exists(q, t) \\
\hline
\exists(q, q') & \quad \forall(q, q') \\
\end{align*}
\]

(∃)

\[
\begin{align*}
\forall(p, q) & \quad \exists(q, t) \\
\hline
\exists(q, q') & \quad \forall(q, q') \\
\end{align*}
\]

(∃)

\[
\begin{align*}
\forall(p, q) & \quad \exists(q, t) \\
\hline
\exists(q, q') & \quad \forall(q, q') \\
\end{align*}
\]

(∃)

\[
\begin{align*}
\forall(p, q) & \quad \exists(q, t) \\
\hline
\exists(q, q') & \quad \forall(q, q') \\
\end{align*}
\]

(∃)

\[
\begin{align*}
\forall(p, q) & \quad \exists(q, t) \\
\hline
\exists(q, q') & \quad \forall(q, q') \\
\end{align*}
\]

(∃)

\[
\begin{align*}
\forall(p, q) & \quad \exists(q, t) \\
\hline
\exists(q, q') & \quad \forall(q, q') \\
\end{align*}
\]

(∃)

\[
\begin{align*}
\forall(p, q) & \quad \exists(q, t) \\
\hline
\exists(q, q') & \quad \forall(q, q') \\
\end{align*}
\]

(∃)
Let \( \Gamma \) be a set of sentences in our fragment, let \( S \) be \( \Gamma \)-EC assume that \( \mathcal{E}(\Gamma) \subseteq S \). We define a model \( M = M(\Gamma, S) \) by taking
\[
(5.14) \quad M = \{x_1, x_2 : x \in S\} \cup \{\{p, q\} : p \neq q \text{ and } \Gamma \vdash \exists(p, q)\}.
\]
(This differs from the construction in the previous section in that the elements in the first part of the union are those from \( S \); so for \( x_i \in M \), we need not have \( \Gamma \vdash \exists(x, x_i) \).) But the rest of the definition of the structure is exactly the same as in our previous construction.

**Proposition 5.16** If \( [x] \neq \emptyset \) in \( M(\Gamma, S) \), then \( x \in S \).

**Proof** First, assume that \( y_i \in [x] \), so that \( y \in S \) and \( \Gamma \vdash \forall(y, x) \). Then \( \emptyset \)

**Lemma 5.17** Let \( \Gamma \) be a theory in positive-\( \mathcal{R} \), and let \( S \) be \( \Gamma \)-EC. Then \( \mathcal{M}(\Gamma, S) \models \Gamma \).

**Proof** Most of the details are the same as in Lemma 5.12. The one difference is in the case of a sentence in \( \Gamma \) of the form \( \forall(x, \exists(y, r)) \). We may assume \([x] \neq \emptyset\), or our result is trivial. An element of \([x]\) is either of the form \( z_i \), or else of the form \( \{u, v\} \). Let \( z_1 \in [x] \), so that \( z \in S \) and \( \forall(z, x) \). Then \( \Gamma \vdash \forall(z, \exists(y, r)) \). As \( S \) is \( \Gamma \)-EC, \( y \in S \). Then \( z_1[r]y \) by construction. The other case is for \( \{u, v\} \in [x] \), so that \( \Gamma \vdash \exists(u, v) \). Without loss of generality, \( u \leq x \). Then \( \Gamma \vdash \forall(u, \exists(y, r)) \). By construction, \( \{u, v\} \{r\}y \).

**Theorem 5.18** The logical system determined by the rules in Figure 5.4 is sound and complete for positive-\( \mathcal{R} \).

**Proof** Suppose \( \Gamma \models \varphi \). We show that \( \Gamma \vdash \varphi \). The argument splits into cases according to the syntax of \( S \).

The first case: \( \varphi \in \mathcal{L}(all, some) \). We omit this case and leave it as an exercise to the reader.

\( \varphi \) is \( \exists(x, \exists(y, r)) \). Let \( S = \mathcal{E}(\Gamma) \), and consider \( M(\Gamma, S) \). By Lemma 5.12, \( M \models \Gamma \).

Hence \( M \models \varphi \). There are several cases here, depending on the data witnessing \( \varphi \). One case is when there are \( u_i \in [x] \) and \( w_j \in \{y\} \) related by \([r]\). We argue from \( \Gamma \). We have \( \forall(u, x), \forall(w, y), \exists u, \exists w \), and then easily we see that \( \Gamma \vdash \exists(u, \exists(w, r)) \).

Then by monotonicity, \( \Gamma \vdash \exists(x, \exists(y, r)) \), as desired.

We must also consider the case \([r]\) contains a pair such as \((\{a, b\}, \{c, d\})\). There would be four subcases here, and we go into details on only one of them: suppose \( \{a, b\} \{r\} \{c, d\} \). Without loss of generality, \( \Gamma \vdash \forall(a, \forall(c, r)) \). Since \( \Gamma \) also derives \( \exists a, \forall(a, x), \exists b, \) and \( \forall(b, y) \), we easily get the desired \( \exists(x, \exists(y, r)) \). The reasoning is similar when \([r]\) contains a pair such as \((\{a, b\}, y_j)\) or one such as \((x_i, \{a, b\})\).

\( \varphi \) is \( \forall(x, \forall(y, r)) \). Let
\[
S = \mathcal{E}(\Gamma) \cup \{u : x \sim u \text{ or } y \sim u\}.
\]

\( S \) is \( \Gamma \)-EC. As a result, \( M(\Gamma, S) \models \Gamma \). Hence also \( M(\Gamma, S) \models \varphi \). Note that \( x \) and \( y \) belong to \( S \), and also \( x \in [x] \) and \( y \in \{y\} \). Since \( M(\Gamma, S) \models \varphi \), \( x_1[r]y_2 \). Then by Proposition 5.11, part 1, \( \Gamma \vdash \varphi \).

\( \varphi \) is \( \forall(x, \exists(y, r)) \). Here we take
\[
(5.15) \quad S = \mathcal{E}(\Gamma) \cup \{u : x \sim u\}.
\]

As in our previous cases, \( M(\Gamma, S) \models \varphi \). We have \( x \in S \), by definition. In particular, \( x_1 \in M \). Every witness to \( \varphi \) in \([y]\) for \( x_1 \) is either of the form \( z_i \) or a pair \( \{a, b\} \).
This paragraph only presents the details for a witness of the first form. Let $z_i$ be such that $x_1[r]z_i$ and $z \leq y$. $\Gamma$ must derive one of the following: $\forall(x, \exists(z, r))$, or $\forall(x, \forall(z, r))$. In the first case, we easily get $\forall(x, \exists(y, r))$. We thus concentrate on the second case. Since $z \in S$, either $z \in \mathcal{E}(\Gamma)$, or else $x \sim z$. If $z \in \mathcal{E}(\Gamma)$, then from $\forall(x, \forall(z, r))$, we also get $\forall(x, \exists(z, r))$. By monotonicity, we have $\varphi$. The more interesting case is when $x \sim z$. When $x \sim z$, we use (J) to immediately derive $\varphi$.

We have another overall case, going back to $x_2 \in M$. Suppose that $\{a, b\} \in [y]$ (so that $\exists(a, b)$) and $x_2[r]\{a, b\}$. There are four cases here, perhaps one representative one is when $\Gamma \vdash \forall(a, y)$ and $\forall(x, \forall(b, r))$. Then from $\Gamma$ we have $\forall(x, \exists(a, r))$.

Finally, by monotonicity we have $\forall(x, \exists(y, r))$.

$\varphi$ is $\exists(x, \forall(y, r))$. On our assumption that $\Gamma \models \varphi$, we have $\Gamma \models \exists x$. Thus by the first case on $\varphi$ in this overall proof of this theorem, we know that $x \in \mathcal{E}(\Gamma)$. In the model construction, we take $S = \mathcal{E}(\Gamma) \cup \{u : y \sim u\}$. The resulting model $M(\Gamma, S)$ then satisfies $\varphi$. Suppose that $z_i \in [x]$ is related to all elements of $[y]$. (In the next paragraph, we consider an element of the form $\{a, b\}$.) By the structure of the model, we either have $\Gamma \vdash \exists(z, \forall(y, r))$ (and then we are easily done because $z \leq x$) or $\Gamma \vdash \forall(z, \forall(y, r))$. In the latter case, if $z \in \mathcal{E}(\Gamma)$, we easily see that $\Gamma \vdash \varphi$. So we are left with the case that $y \sim z$. And here we use (HH).

We also need to consider the situation when a pair element such as $\{a, b\} \in [x]$ is related to all elements of $[y]$. One representative case is when $a \leq x$. By definition of $S$, $y \in S$ and so $y_2 \in M$. In particular, $\{a, b\}[r]y_2$. By the structure of the model, we have $\forall(a, \forall(y, r))$ or $\forall(b, \forall(y, r))$; without loss of generality, it is $\forall(b, \forall(y, r))$. Since $\Gamma \vdash \exists(a, b)$, we have $\exists(a, \forall(y, r))$. We would then have $\Gamma \vdash \exists(x, \forall(y, r))$.

This concludes the proof of Theorem 5.18.

### 5.4. Incorporating Background Facts, I

Suppose we have a stock of background facts about verbs, such as

\[(5.16)\]  

*hit*ting entails *touching*

We mean that every act of hitting is also an act of touching. This background fact cannot be stated in any of the languages which we have so far studied. Nevertheless, it can be made into a semantic requirement: we would require of a model that the interpretation of *hit*ting be a sub-relation of the interpretation of *touching*. And then we might like to study the entailment relation on this smaller class of models.

Even though we cannot state (5.16) as an *axiom*, it does yield a rule of inference. To state it more abstractly, suppose we have a rule like

\[(5.17)\]  

$r \Rightarrow s$

and we restrict attention to the models where $[r] \subseteq [s]$. Then Figure 5.5 lists sound rules of inference.
We add these to the system $R$ as listed in Figure 5.2.

**Proposition 5.19** The system $R$ together with the rules in Figure 5.5 give a sound and complete logic: $\Gamma \vdash \varphi$ iff $\Gamma \models \varphi$.

**Proof** The soundness being easy, we need only show that every consistent set in the logic is satisfiable. Let $\Gamma$ be consistent in the current system. Then $\Gamma$ is consistent in $R$, and the proof of Theorem 5.10 shows how to build a model $M(\Gamma)$ of it. It remains to check that whenever $r \Rightarrow s$ is one of our constraints, then $[r] \subseteq [s]$. And for this, we must look back at Figure 5.3 and the semantics of verbs in $M(\Gamma)$ defined on page 56.

Figure 5.3 defines a relation $R^r_{x,y}$ for atoms $x$ and $y$. There is an implicit dependence to a relation, and so in this proof we shall write $R^r_{x,y}$, again under the assumption that $r \Rightarrow s$.

If (a) holds for $r$, then by our logic, (a) also holds for $s$. So in this case, $R^r_{x,y} = R^s_{x,y}$.

If (b) holds for $r$, then either (a) or (b) holds for $s$, and the pictures in Figure 5.3 tell us that $R^r_{x,y} \subseteq R^s_{x,y}$.

If (c) holds for $r$, then either (a) or (b) or (c) holds for $s$, and again $R^r_{x,y} \subseteq R^s_{x,y}$.

If (d) holds for $r$, then either (a) or (b) or (d) holds for $s$. (It is important to see that (c) cannot hold for $s$ in this case.) Again $R^r_{x,y} \subseteq R^s_{x,y}$.

If (e) holds for $r$, then one of (a) – (e) holds for $s$; (f) cannot hold. Once again $R^r_{x,y} \subseteq R^s_{x,y}$.

If (f) holds for $r$, then $R^r_{x,y} \subseteq R^s_{x,y}$ because $R^r_{x,y}$ is empty.

So in all cases, we have checked that $R^r_{x,y} \subseteq R^s_{x,y}$. At this point, we return to the definitions of $[r]$ and $[s]$ to see that $[r] \subseteq [s]$. Recall that these relations involve pair-elements, and they are defined in a case-by-case fashion. However, all of the cases are positive, (that is, the atoms occur positively rather than negatively), and so it is easy to check by inspection that indeed $[r] \subseteq [s]$.

In this way, we have verified that the model $M(\Gamma)$ indeed respects background information. This concludes the proof. □

**Exercise 33** In this section we gave a precise definition of the derivability relation $\Gamma \vdash \varphi$. Here is another characterization of this relation. Let $\models_{alt}$ be the the smallest relation on $P(R) \times R$ such that

1. If $\varphi \in \Gamma$, then $\Gamma \models_{alt} \varphi$.
2. If $\Gamma \models_{alt} \psi_1, \ldots, \psi_n$ and one of the rules of $R$ other than (RAA) has as a substitution instance $\psi_1, \ldots, \psi_n \setminus \varphi$, then $\Gamma \models_{alt} \varphi$.
3. If $\Gamma \cup \{ \overline{\varphi} \} \models_{alt} \bot$, then $\Gamma \models_{alt} \varphi$.

Prove that $\Gamma \vdash \varphi$ iff $\Gamma \models_{alt} \varphi$.

Sources. The first syllogistic logic to employ verbs was Nishihara, Morita, and Iwata [24]. The negative result on $R$ is due to Pratt-Hartmann and Moss [29]. The logical system for $R$ comes from Moss [21] and from Pratt-Hartmann and Moss [29]. The proof in the latter paper is somewhat different, and it has the advantage of also giving the NLogSpace complexity result. The work on incorporating background facts is new here.
CHAPTER 6

Relative Clauses: \( \mathcal{RC} \)

6.1. \( \mathcal{RC} \): an Indirect System

This chapter continues the study of verbs with a presentation of logics containing relative clauses.

We begin with a language called \( \mathcal{RC} \). Figure 6.1 exhibits a set of syllogistic rules which we call \( \mathcal{RC} \) used in connection with the language \( \mathcal{RC} \). We shall show that the indirect derivation relation \( \vdash_{\mathcal{RC}} \) is complete. In the following, the variables \( b^+, c^+ \) range over positive \( c \)-terms, and \( d \) over \( c \)-terms. (As usual, \( p, q \) range over unary atoms and \( r \) over binary atoms.)

Rules (D1), (D2), (B), (T) and (I) are natural generalizations of their namesakes in \( \mathcal{R} \). Rules (J), (K), and (L) embody logical principles that are intuitively clear, yet not familiar when taken as single steps. If all porcupines are brown animals, then everything which attacks all brown animals attacks all porcupines (J), and everything which photographs some porcupine photographs some brown animal (K). And if some porcupines are brown animals, then everything which caresses all porcupines caresses some brown animals (L). We have already seen (II). Rule (Z) tells that if there are no porcupines (say), then all farmers love all porcupines. Rule (W) tells us that under this same assumption, there is something which loves all porcupines (simply because we assume the universe is non-empty, and everything in it vacuously loves all porcupines). If we did not assume that the universe of a model is non-empty, then we would drop (W), and the completeness of the resulting system would be proved the same way.

All the rules of \( \mathcal{R} \) are derivable in \( \vdash_{\mathcal{RC}} \). For example, here is a proof of (A):

\[
\frac{\exists(p, d)}{\exists(p, p)} \quad \frac{\exists(p, p)}{\forall(p, \overline{p})} \quad \frac{\exists(p, \overline{p})}{\forall(p, d)}
\]

\( \text{(D1)} \)

\( \text{(RAA)} \)

Theorem 6.1 The derivation relation \( \vdash_{\mathcal{RC}} \) is sound and complete for \( \mathcal{RC} \).

Proof To prove completeness, we need only show that every theory \( \Gamma \) in the fragment \( \mathcal{RC} \) which is consistent in the logic \( \mathcal{RC} \) is satisfiable. Also, by Lemma 5.9, we may assume that \( \Gamma \) is \( \mathcal{RC} \)-complete. For the remainder of this section, fix some \( \mathcal{RC} \)-complete set of formulas \( \Gamma \) which is consistent with respect to \( \vdash_{\mathcal{RC}} \). We simplify our notation to write \( \vdash \) for \( \vdash_{\mathcal{RC}} \).
The main work concerns c-terms of the form resulting fact that if $\Gamma$
all c-terms referred to in this proof are positive $M$ be the set of positive c-terms. Then we define $68 6. RELATIVE CLAUSES: RC$
from $\Gamma$ as in the tree on the left below:

$$
\begin{array}{llll}
\forall (c^+, c^+) & \Rightarrow & \exists (c^+, d) & \Rightarrow \\
\forall (b^+, c^+) & \Rightarrow & \forall (b^+, c^+) & \Rightarrow \\
\exists (b^+, c^+) & \Rightarrow & \forall (c^+, d) & \Rightarrow \\
\forall (p, q) & \Rightarrow & \forall (p, q) & \Rightarrow \\
\forall (p, q) & \Rightarrow & \exists (p, q) & \Rightarrow \\
\exists (p, q) & \Rightarrow & \forall (p, p) & \Rightarrow \\
\forall (p, p) & \Rightarrow & \forall (c^+, \forall (p, r)) & \Rightarrow \\
\forall (c^+, \forall (p, r)) & \Rightarrow & \forall ((\forall (p, r), \forall (p, r)) & \Rightarrow \\
\end{array}
$$

Figure 6.1. Proof rules for the system RC.

We shall construct a structure $M$ and prove that it satisfies $\Gamma$. First, let $C^+$
be the set of positive c-terms. Then we define $M$ by:

$$
\begin{array}{l}
A = \{ (c_1, c_2, Q) \in C^+ \times C^+ \times \{\forall, \exists\} : \Gamma \vdash \exists (c_1, c_2) \}
\end{array}
$$

$$
\begin{array}{l}
[p] = \{ (c_1, c_2, Q) \in A : \Gamma \vdash \forall (c_1, p) \text{ or } \Gamma \vdash \forall (c_2, p) \}
\end{array}
$$

iff either (a) for some $i$, $j$, and $q \in P$, $\Gamma \vdash \forall (c_i, \forall (q, r))$ and $\Gamma \vdash \forall (d_j, q)$; or else (b) $Q_2 = \exists$, and for some $i$ and $q \in P$, $d_1 = d_2 = q$, and $\Gamma \vdash \forall (c_i, \exists (q, r))$.

Note that the set $A$ is non-empty. For let $p \in P$. If $\Gamma \vdash \exists (p, p)$, then $(p, p, \forall) \in A$. Otherwise, $\Gamma \vdash \forall (p, p)$, and so for all binary atoms $r$, $\Gamma \vdash \exists (\forall (p, r), \forall (p, r))$ by (W).

Thus $(c, c, \forall) \in A$, where $c$ is $\forall (p, r)$.

**Lemma 6.2** For all $c \in C^+$, $[c] = \{ (d_1, d_2, Q) \in A : either \Gamma \vdash \forall (d_1, c), or \Gamma \vdash \forall (d_2, c) \}$.

Proof The result for $c$ a unary atom is immediate. We often shall use the resulting fact that if $\Gamma \vdash \exists (p, p)$, then $[p] \neq \emptyset$; it contains both $(p, p, \forall)$ and $(p, p, \exists)$. The main work concerns c-terms of the form $\forall (p, r)$ and $\exists (p, r)$. We remark that all c-terms referred to in this proof are positive.

We begin with $c = \forall (p, r)$. Let $(d_1, d_2, Q) \in [\forall (p, r)]$. Now, either $\Gamma \vdash \exists (p, p)$ or $\Gamma \not\vdash \exists (p, p)$. If the former, then $(p, p, \forall) \in [p]$. By the semantics of our fragment, $(d_1, d_2, Q) [r] (p, p, \forall)$. By the structure of $M$, there are $i$ and $q$ giving the derivation from $\Gamma$ as in the tree on the left below:

$$
\begin{array}{l}
\vdots \\
\forall (d_i, \forall (q, r)) \Rightarrow \forall (p, q) \Rightarrow \\
\forall (p, q) \Rightarrow \forall (\forall (q, r), \forall (p, r)) \Rightarrow \\
\forall (\forall (q, r), \forall (p, r)) \Rightarrow \forall (p, p) \Rightarrow \\
\forall (p, p) \Rightarrow \forall (\forall (p, r)) \Rightarrow \\
\forall (\forall (p, r)) \Rightarrow \forall (p, p, \forall) \Rightarrow \\
\end{array}
$$

(Z).


This shows that $\Gamma \vdash \forall(d_i, \forall(p, r))$. On the other hand, if $\Gamma \not\vdash \exists(p, p)$, we use the assumption that $\Gamma$ is complete to assert that $\Gamma \not\vdash \forall(p, p)$. And then we have the derivation from $\Gamma$ on the right above, for both $j$.

Conversely, fix $i$ and suppose that $\Gamma \vdash \forall(d_i, \forall(p, r))$. We claim that $\langle d_1, d_2, Q \rangle$ belongs to $[\forall(p, r)]$. For this, take any $\langle b_1, b_2, Q' \rangle \in [p]$ so that $\Gamma \vdash \forall(b_j, p)$ for some $j$. (We are thus using $b_1$ and $b_2$ to range over positive c-terms, just as the $c$'s and $d$'s do.) Then $p$, $i$ and $j$ show that $\langle d_1, d_2, Q[r] \rangle \langle b_1, b_2, Q' \rangle$. This for all elements of $[p]$ shows that $\langle d_1, d_2, Q \rangle \in [\forall(p, r)]$.

We next prove the statement of our lemma for $c = \exists(p, r)$.

Let $\langle d_1, d_2, Q \rangle \in [\exists(p, r)]$. Thus we have $\langle d_1, d_2, Q[r] \rangle \langle b_1, b_2, Q' \rangle$ for some $\langle b_1, b_2, Q' \rangle \in [p]$. We first consider case (a) in the definition of our structure $M$: there are $i$, $j$, and $q$ so that $\Gamma \vdash \forall(d_i, \exists(q, r))$ and $\Gamma \vdash \forall(b_j, q)$. We have $\Gamma \vdash \exists(b_1, b_2)$, since $\langle b_1, b_2, Q' \rangle \in A$. Further, let $k$ be such that $\Gamma \vdash \forall(b_k, p)$. We show the desired conclusion using a derivation from $\Gamma$:

\[
\begin{array}{c}
\vdots \\
\exists(b_1, b_2) \\
\exists(b_k, q) \\
\exists(q, p) \\
\vdots \\
\forall(d_i, \forall(q, r)) \\
\forall(d_i, \exists(p, r)) \\
\forall(b_j, q) \\
\forall(b_k, p) \\
\end{array}
\]

\[
\text{(D1)} \quad \text{(D1)} \quad \text{(L)} \quad \text{(B)}
\]

This concludes the work in case (a). In case (b), $Q' = \exists$, there is some $q \in P$ such that $b_1 = b_2 = q$, and for some $i$, $\Gamma \vdash \forall(d_i, \exists(q, r))$. Again we have $\Gamma \vdash \forall(q, p)$. So we have a derivation from $\Gamma$ as follows:

\[
\begin{array}{c}
\vdots \\
\forall(q, p) \\
\forall(\exists(q, r), \exists(p, r)) \\
\forall(d_i, \exists(p, r)) \\
\end{array}
\]

\[
\text{(K)} \quad \text{(B)}
\]

At this point, we know that if $\langle d_1, d_2, Q \rangle \in [\exists(p, r)]$, then $\Gamma \vdash \forall(d_i, \exists(p, r))$ for some $i$. We now verify the converse. Let $\langle d_1, d_2, Q \rangle \in A$, and fix $i$ such that $\Gamma \vdash \forall(d_i, \exists(p, r))$. Then $\Gamma \vdash \exists(d_1, d_2)$. We thus have a derivation from $\Gamma$:

\[
\begin{array}{c}
\exists(d_1, d_2) \\
\exists(d_i, d_i) \\
\exists(d_i, \exists(p, r)) \\
\end{array}
\]

\[
\text{(I)} \quad \text{(II)} \quad \text{(D1)}
\]

This goes to show that $\langle p, p, \exists \rangle \in A$. By the construction of $M$, $\langle d_1, d_2, Q[r] \rangle \langle p, p, \exists \rangle$, and $\langle p, p, \exists \rangle \in p^M$. So $\langle d_1, d_2, Q \rangle \in [\exists(p, r)]$. This completes the proof.

**Lemma 6.3** $M \models \Gamma$.

**Proof** The proof is by cases on the various formula types in $\mathcal{RC}$. Using the fact that formulas $\exists(e, f)$ and $\exists(f, e)$ are identified, and similarly for $\forall(e, f)$ and
∀(f, e), we may take all \( \mathcal{R} \)-formulas to have one of the forms:
\[
\forall(c^+, d^+), \quad \forall(c^+, \overline{d^+}), \quad \exists(c^+, d^+), \quad \exists(c^+, \overline{d^+}),
\]
where \( c^+ \) and \( d^+ \) range over positive \( c \)-terms. In the remainder of the proof, we
omit the \( + \)-superscripts for clarity: i.e. \( c \) and \( d \) range over positive \( c \)-terms.

Let \( \varphi \in \Gamma \) be \( \forall(c, d) \). Using (B) and Lemma 6.2, we see that \( [c] \subseteq [d] \).

Finally, consider the case when \( \varphi \in \Gamma \) is of the form \( \exists(c, d) \). Then, using (I),
\( \Gamma \vdash \exists(c, c) \), so \( (c, c, \forall) \in A \). Suppose towards a contradiction that \( M \models \exists(d, \overline{d}) \).
Then \( (c, c, \forall) \in \overline{[d]} \). But then we have \( \Gamma \vdash \exists(d, \overline{d}) \), by Lemma 6.2 again. One
application of (D2) now shows that \( \Gamma \vdash \exists(d, \overline{d}) \). Thus we have a contradiction to
the consistency of \( \Gamma \).

This completes the proof of Theorem 6.1.

**6.1. Negative results for larger fragments.** Recall that the languages
\( \mathcal{R}^\dagger \) and \( \mathcal{R}^\mathcal{C}^\dagger \) have full noun-level negation. (This is the point of the \( \dagger \) notation.)
Recall also from Theorem 5.15 that there is no logical system for \( \mathcal{R} \) which is simul-
taneously finite, sound and complete. In fact, an even stronger result holds for \( \mathcal{R}^\dagger \) and \( \mathcal{R}^\mathcal{C}^\dagger \).

**Theorem 6.4** There exists no finite set \( X \) of syllogistic rules in \( \mathcal{R}^\dagger \) such that
\( \models X \) is both sound and complete.

That is, even allowing for reductio ad absurdum, there are still no logical systems
for \( \mathcal{R}^\dagger \) and \( \mathcal{R}^\mathcal{C}^\dagger \) which are simultaneously finite, sound and complete.

The proof of Theorem 6.4 is similar to what we saw earlier in Theorem 5.15
for \( \mathcal{R} \), but the argument here is more intricate and so we are not going to give it in
these notes.

**6.2. Complexity Results for \( \mathcal{R}, \mathcal{R}^\dagger, \) and \( \mathcal{R}^\mathcal{C}^\dagger \)**

In this section, we study the computational complexity of the logical systems
with which we have been concerned. We are mainly interested in the complexity of the
consequence relation, that is
\[
\{(\Gamma, \varphi) : \Gamma \text{ is a finite set, and } \Gamma \vdash \varphi \}.
\]

Recall that we defined syllogistic proof systems in Section 5.2.1. Derivation
relations defined by direct proof-systems are easily seen to have polynomial-time
complexity.

**Lemma 6.5** Let \( \mathcal{F} \) be a syllogistic fragment, and \( X \) a finite set of syllogistic rules
in \( \mathcal{F} \). The problem of determining whether \( \Theta \vdash_X \theta \), for a given set of \( \mathcal{F} \)-formulas
\( \Theta \) and \( \mathcal{F} \)-formula \( \theta \), is in \( \text{PTime} \).

**Proof** Let \( \Sigma \) be the set of all atoms (unary or binary) occurring in \( \Theta \cup \{ \theta \} \),
together with one additional binary atom \( r \). We first observe that, if there is a
derivation of \( \theta \) from \( \Theta \) using the rules \( X \), then there is such a derivation involving
only the atoms occurring in $\Sigma$. For, given any derivation of $\theta$ from $\Theta$, uniformly replace any unary atom that does not occur in $\Theta \cup \{\theta\}$ with one that does. Similarly, uniformly replace any binary atom which does not occur in $\Theta \cup \{\theta\}$ with one which does (or with $r$ in case $\Theta \cup \{\theta\}$ contains no binary atoms). This process obviously leaves us with a derivation of $\theta$ from $\Theta$, using the rules $\mathcal{X}$.

To prove the lemma, let the total number of symbols occurring in $\Theta \cup \{\theta\}$ be $n$. Certainly, $|\Sigma| \leq n$. Let $\mathcal{X}$ comprise $k_1$ proof-rules, each of which contains at most $k_2$ atoms (unary or binary). The number of rule instances involving only atoms in $\Sigma$ is bounded by $p(n) = k_1 n^{k_2}$. Hence, we need never consider derivations with ‘depth’ greater than $p(n)$. Let $\Theta_i$ be the set of formulas involving only the atoms in $\Sigma$, and derivable from $\Theta$ using a derivation of depth $i$ or less ($0 \leq i \leq p(n)$). Evidently, $|\Theta_i| \leq |\Theta| + p(n)$. It is then straightforward to compute the successive $\Theta_i$ in total time bounded by a polynomial function of $n$.

**Theorem 6.6 (McAllester and Givan [15])** The satisfiability problem for a sequent $\Gamma$ in $\mathbb{R}^+$ is $\text{NPTime}$-complete. (Thus the validity problem is co-$\text{NPTime}$-complete.)

**Proof** It follows from the proof of Theorem 6.1, that if $\Gamma$ is a satisfiable theory in $\mathbb{R}^+$, then $\Gamma$ has a model whose size is at most $2n$, where $n$ is the number of positive $c$-terms in the language. (This bound is independent of $\Gamma$.) It follows that satisfiability is in $\text{NPTime}$. The main work is in showing the $\text{NPTime}$-hardness.

Our proof is a small variation on the original argument. We use a reduction from the monotone exactly-1 $3\text{SAT}$ problem. This problem is defined as follows. We are given a conjunction of 3-CNF clauses without negation, so each clause is of the form $U \lor V \lor W$. The problem is to find a truth assignment $f$ to the variables making one variable in each clause $T$ and the other two variables $F$. We call this a 1-valued assignment on $S$. This problem was shown to be $\text{NPTime}$-complete in Schaefer [31].

Consider a sentence $S$ which is a conjunction of clauses, each of which is a disjunction of three variables without negation. We define an $\mathbb{R}^+$-theory $\Gamma = \Gamma(S)$ via two clauses below. It uses unary atoms which correspond to the variables of $S$: we use $u$ to correspond with $U$, etc. $\Gamma$ also uses a number of new unary and binary atoms. It is defined as follows:

1. For each clause of $S$, say $c \equiv U \lor V \lor W$, add to $\Gamma$ the seven sentences

   \[
   \forall (x_c, \forall (u, r^1_c)) \quad \forall (z_c, \forall (w, r^3_c)) \\
   \forall (\exists (u, r^1_c), y_c) \quad \forall (\exists (w, r^3_c), a_c) \\
   \forall (y_c, \forall (v, r^2_c)) \quad \exists (x_c, a_c) \\
   \forall (\exists (v, r^2_c), z_c)
   \]

   Here $x_c, y_c, z_c$ and $a_c$ are new unary atoms, and $r^1_c, r^2_c,$ and $r^3_c$ are new binary atoms.

2. Let $P$ and $Q$ be any two distinct variables which occur together in some clause $c$. Then add to $\Gamma$ the sentence

   \[
   \forall (\exists (p, r_{p,q}), \exists (q, r'_{p,q}))
   \]

   Here $r_{p,q}$ and $r'_{p,q}$ are new binary atoms.

So if $S$ has $k$ clauses, then the first point will add $4k$ new unary atoms and $3k$ new binary atoms. The second clause will add at most $2 \cdot \binom{3k}{2} < 18k^2$ new binary atoms. We assume that all of the atoms listed are distinct.
The main claim is that $S$ has a 1-valued assignment iff $\Gamma$ is satisfiable. In one direction, assume that $M \models \Gamma$. Define a truth assignment $f$ by $f(U) = F$ iff $[u] \neq \emptyset$. Consider a clause $c \equiv U \lor V \lor W$ of $S$. If $f(U) = f(V) = f(W) = F$, then $[u]$, $[v]$, and $[w]$ are all non-empty. By the first six points in (1), $[x_c] \subseteq [y_c] \subseteq [z_c] \subseteq [a_c]$. But this contradicts the last point in (1). Thus we know that at least one variable in $c$ is assigned the value $T$ by $f$. We claim that only one variable can be $T$. For suppose towards a contradiction that (for example) $f(U) = f(V) = T$. Then $[u] = [v] = \emptyset$. So $[\neg v(u, r_{p,q})] = M$ and $[\exists v, r'_{p,q}] = \emptyset$. By the sentence in (2), $M$ is empty. But this is impossible, since as soon as $S$ has at least one clause, $\Gamma$ has an existential assertion via (1). In this way, $f$ is 1-valued on $S$.

We conclude by checking the converse assertion. Suppose $f$ is 1-valued on $S$. We must find a model $M \models \Gamma$. Let $M$ be the set of variables $U$ such that $f(U) = F$. For a variable $X$, define $[x] = \emptyset$ if $f(X) = T$, and $[x] = \{x\}$ if $f(X) = F$. We still need to define the interpretations of all of the binary atoms, and all of the other unary atoms. Suppose that $P$ and $Q$ are distinct variables which happen to belong to the same clause. We know that either $f(P) = F$ or $f(Q) = F$ (or both). In the first case, set $[r_{p,q}] = \emptyset$ so that $[\neg v(p, r_{p,q})] = \emptyset$. (The interpretation of $r'_{p,q}$ is arbitrary in this case; we no longer mention this point.) This makes $M \models \forall (v(p, r_{p,q}), \exists (q, r'_{p,q}))$.

In the second case, $[r'_{p,q}] = M \times M$, so that $[\exists (q, r'_{p,q})] = M$. In this way, the sentence in (2) holds in $M$.

Finally, we consider the sentences in (1). If $f(U) = T$, $f(V) = F$, and $f(W) = F$, then we already have $[u] = \emptyset$, $[v] = \{v\}$, and $[w] = \{w\}$. We set $[x_c] = M$, $[y_c] = \emptyset$, $[z_c] = \emptyset$, $[a_c] = \{a_c\}$, $[b_c] = \{b_c\}$, $[c_c] = \{c_c\}$, and $[w, w] = \{w, w\}$.

If $f(U) = F$, $f(V) = T$, and $f(W) = F$, set $[x_c] = M$, $[y_c] = M$, $[z_c] = \{w\}$, $[a_c] = \{a_c\}$, $[b_c] = \{b_c\}$, $[c_c] = \{c_c\}$, and $[w, w] = \{w, w\}$.

If $f(U) = F$, $f(V) = F$, and $f(W) = T$, set $[x_c] = M$, $[y_c] = M$, $[z_c] = M$, $[a_c] = \emptyset$, $[b_c] = \emptyset$, $[c_c] = \emptyset$. In all cases, the resulting model $M$ satisfies all sentences in (1), hence all sentences in $\Gamma$.

\[\square\]

**Lemma 6.7** (Pratt-Hartmann and Moss [29]) The problem of determining the validity of a sequent in $\mathcal{R}^1$ is EXPSPACE-hard.

**Proof** The logic $K^U$ is the basic modal logic $K$ together with an additional modality $U$ (for “universal”), whose semantics are given by the standard relational (Kripke) semantics, plus

$$\models_w U \varphi \text{ if and only if } \models_w \varphi \text{ for all worlds } w'.$$

The satisfiability problem for $K^U$ is EXPSPACE-hard. (The proof is an easy adaptation of the corresponding result for propositional dynamic logic; see, e.g. Harel et al. [9]: 216 ff.) It suffices, therefore, to reduce this problem to satisfiability in $\mathcal{R}^1$. Let $\varphi$ be a formula of $K^U$.

We first transform $\varphi$ into an equisatisfiable set of formulas $T_\varphi \cup S_\varphi$ of first-order logic; then we translate the formulas of $T_\varphi \cup S_\varphi$ into an equisatisfiable set of $\mathcal{R}^1$-formulas. To simplify the notation, we shall take unary atoms (in $\mathcal{R}^1$) to be unary predicates (in first-order logic); similarly, we take binary atoms to do double duty as binary predicates. Let $r$ and $c$ be binary atoms. For any $K^U$-formula $\psi$, let $p_\psi$ be a unary atom, and define the set of first-order formulas $T_\psi$ inductively as
follows:

\[ T_p = \emptyset \text{ (where } p \text{ is a proposition letter)} \]
\[ T_{\psi \land \pi} = T_{\psi} \cup T_{\pi} \cup \{ \forall x(p_{\psi}(x) \land p_{\pi}(x) \rightarrow p_{\psi \land \pi}(x)) \} \]
\[ T_{\neg \psi} = T_{\psi} \cup \{ \forall x(p_{\neg \psi}(x) \rightarrow \neg p_{\psi}(x)), \forall x(\neg p_{\neg \psi}(x) \rightarrow p_{\psi}(x)) \} \]
\[ T_{\forall \psi} = T_{\psi} \cup \{ \forall x(p_{\forall \psi}(x) \rightarrow \forall y(\neg p_{\psi}(y) \rightarrow r(x, y))) \} \]
\[ T_{\exists \psi} = T_{\psi} \cup \{ \forall x(p_{\exists \psi}(x) \rightarrow \exists y(\neg p_{\psi}(y) \land e(x, y))) \} \]

Now let \( S_\varphi \) be the collection of five first-order formulas

\[ \exists x(p_{\varphi}(x) \land p_{-\varphi}(x)), \quad \forall x(\pm p_{\varphi}(x) \rightarrow \forall y(\pm p_{\varphi}(y) \rightarrow e(x, y))) \]

(Although the first formula looks like it has a redundant conjunct, we state it in this way only to make our work below a little easier.) We claim that the modal formula \( \varphi \) is satisfiable if and only if the set of first-order formulas \( T_\varphi \cup S_\varphi \) is satisfiable. For let \( M \) be any (Kripke) model of \( \varphi \) over a frame \( (W, R) \). Define the first-order structure \( M \) with domain \( W \), by setting \( r^M = R \), \( e^M = A^2 \), and \( p_{\psi}^M = \{ w \mid M \models_w \psi \} \), for any subformula \( \psi \) of \( \varphi \). It is then easy to check that \( M \models T_\varphi \cup S_\varphi \). Conversely, suppose \( M \models T_\varphi \cup S_\varphi \). We build a Kripke structure \( M \) over the frame \( (A, r^M) \) by setting, for any proposition letter \( o \) mentioned in \( \varphi \), \( M \models_o \psi \) if and only if \( o \in p_{\psi}^M \). A straightforward structural induction establishes that for any subformula \( \psi \) of \( \varphi \), \( M \models_o \psi \) if and only if \( a \in p_{\psi}^M \). The formula \( \exists x(p_{\varphi}(x) \land p_{-\varphi}(x)) \in S_\varphi \) then ensures that \( \varphi \) is satisfied in \( M \).

Now, all of the formulas in \( T_\varphi \cup S_\varphi \) are of one of the forms

\begin{align*}
(6.18) \quad & \forall x(p(x) \rightarrow \pm q(x)) & \forall x(\pm p(x) \rightarrow \forall y(\pm q(y) \rightarrow \pm r(x, y))) \\
(6.19) \quad & \exists x(p(x) \land p(x)) & \forall x(\pm p(x) \rightarrow \exists y(\pm q(y) \land r(x, y))) \\
(6.20) \quad & \forall x(p(x) \land q(x) \rightarrow o(x)) \\
\end{align*}

Notice that formulas of the forms (6.18) and (6.19) translate (in the obvious sense) directly into the fragment \( \mathcal{R}^1 \); those of form (6.20), by contrast, do not. The next step is to eliminate formulas of this last type.

Let \( o^* \) be a new unary relation symbol. For \( \theta \in T_\varphi \cup S_\varphi \) of the form (6.20), let \( r_\theta \) be a new binary atom, and define \( R_\theta \) to be the set of formulas

\begin{align*}
(6.21) \quad & \forall x(\neg o(x) \rightarrow \exists z(o^*(z) \land r_\theta(x, z))) \\
(6.22) \quad & \forall x(p(x) \rightarrow \forall z(\neg p(z) \rightarrow \neg r_\theta(x, z))) \\
(6.23) \quad & \forall x(q(x) \rightarrow \forall z(p(z) \rightarrow \neg r_\theta(x, z))) ,
\end{align*}

which are all of the forms in (6.18) or (6.19). It is easy to check that \( R_\theta \models \theta \). For suppose (for contradiction) that \( M \models R_\theta \) and \( a \) satisfies \( p \) and \( q \) but not \( o \) in \( M \). By (6.21), there exists \( b \) such that \( M \models r_\theta(a, b) \). If \( M \models p[b] \), then (6.22) is false in \( M \); on the other hand, if \( M \models p[b] \), then (6.23) is false in \( M \). Thus, \( R_\theta \models \theta \) as claimed. Conversely, if \( M \models \theta \), expand \( M \) to a structure \( M' \) by interpreting \( o^* \) and \( r_\theta \) as follows:

\[
(6.2) \quad (o^*)^M = A \\
r_\theta^M = \{ (a, a) \mid M \not\models o[a] \}.\]
We check that \( M' \models R_0 \). Formula (6.21) is true, because \( M' \models o[a] \) implies \( M' \models r_0[a, a] \). Formula (6.22) is true, because \( M' \models r_0[a, b] \) implies \( a = b \). To see that Formula (6.23) is true, suppose \( M' \models q[a] \) and \( M' \models r_0[a, b] \). If \( a = b \), then \( M \models o[a] \) (since \( M' \models \theta \)); that is, either \( a \neq b \) or \( M \models o[a] \). By construction, then, \( M' \not\models r_0[a, b] \).

Now let \( T^*_\varphi \) be the result of replacing all formulas \( \theta \) in \( T_\varphi \) of form (6.20) with the corresponding trio \( R_\theta \). (The binary atoms \( r_\theta \) for the various \( \theta \) are assumed to be distinct; however, the same unary atom \( o^* \) can be used for all \( \theta \).) By the previous paragraph, \( T^*_\varphi \cup S_\varphi \) is satisfiable if and only if \( T_\varphi \cup S_\varphi \) is satisfiable, and hence if and only if \( \varphi \) is satisfiable. But \( T^*_\varphi \cup S_\varphi \) is a set of formulas of the forms (6.18) and (6.19), and can evidently be translated into a set of \( R^\dagger \)-formulas satisfied in exactly the same structures. Moreover, this set can be computed in time bounded by a polynomial function of \( \| \varphi \| \). This completes the reduction. \( \square \)

We note the following fact. (We omit a detailed proof, since subsequent developments do not hinge on this result.)

**Lemma 6.8** The problem of determining the validity of a sequent in \( \mathcal{R}^\dagger \mathcal{C} \) is in \( \text{ExpTime} \).

**Proof** Trivial adaptation of Pratt-Hartmann [25], Theorem 3, which considers a fragment obtained by adding relative clauses to the relational syllogistic. \( \square \)

**Theorem 6.9** The validity problem for \( \mathcal{R}^\dagger \) and \( \mathcal{R}^\dagger \mathcal{C} \) are \( \text{ExpTime-complete} \).

**Proof** Lemmas 6.7 and 6.8. \( \square \)

**Corollary 6.10** There exists no finite set \( \mathcal{X} \) of syllogistic rules in either \( \mathcal{R}^\dagger \) or \( \mathcal{R}^\dagger \mathcal{C} \) such that \( \vdash \mathcal{X} \) is both sound and refutation-complete.

**Proof** It is a standard result that \( \text{PTIME} \neq \text{ExpTime} \). The result is then immediate by Lemmas 6.5 and 6.7. \( \square \)

Of course, Corollary 6.10 leaves open the possibility that there exist indirect syllogistic systems that are sound and complete for \( \mathcal{R}^\dagger \) and \( \mathcal{R}^\dagger \mathcal{C} \). To show that there do not, stronger methods are required.

### 6.3. Complete Subsystems

**different point below.** Moss [21] also analyses a syllogistic logic with negated nouns (not verbs) and only using All. So in our notation, its formulas are \( \forall(l, m) \) and \( \forall(l, \forall(m, r)) \), with \( l \) and \( m \) unary literals. In addition to rules we have seen, it uses the following additional rules which might be taken to be forms of the law of the excluded middle:

\[
\begin{align*}
\forall(n, \forall(q, r)) & \quad \forall(\bar{n}, \forall(q, r)) & \quad \forall(p, \forall(n, r)) & \quad \forall(p, \forall(\bar{n}, r)) & \quad \forall(p, \forall(q, r)).
\end{align*}
\]

The system is completed by a rule with three premises:

\[
\begin{align*}
\forall(p, \forall(n, r)) & \quad \forall(n, \forall(q, r)) & \quad \forall(\bar{n}, \forall(\bar{n}, r)) & \quad \forall(p, \forall(q, r)).
\end{align*}
\]

A sub-fragment of the McAllester and Givan fragment was considered in Moss [21]. In our terms, the formulas would be of the forms \( \forall(c, d) \) and \( \exists(c, d) \), where \( c \) and \( d \) are class-terms of the forms \( p \) (an atom), \( \exists(c, r) \), or \( \forall(d, r) \). Thus the system lacks
negation. The rules are versions of rules we have seen in Section 6.1: (T), (I), (B),
(D), (J), (K), (L), and (II); it also employs a rule allowing for reasoning-by-cases,
which is not a syllogistic rule in our sense. An informal example shows that the
system allows non-trivial inferences, and inferences with more than one verb.

\[
\begin{align*}
\text{All porcupines are mammals} \\
\text{All who respect all mammals respect all porcupines} \\
\text{All who dislike all who respect all porcupines dislike all who respect all mammals.}
\end{align*}
\]

6.4. Incorporating Background Facts, II

As in Section 5.4, we can consider background facts among verbs expressed
semantically as constraints of the form \( [r] \subseteq [s] \). In the case of the system \( RC \) we
obtain two new rules:

\[(6.24) \quad \forall(\exists(c, r), \exists(c, s)) \quad \forall(\forall(c, r), \forall(c, s))\]

We add these to the system \( RC \) as listed in Figure 6.1 in order to accommodate
background facts.

Sources. Most of the material in this chapter is from Moss [21], Pratt-Hartmann
and Moss [29].

To the best of my knowledge, the first presentation of a complete proof-system
for a fragment close to the relational syllogistic seems to be Nishihara, Morita, and
Iwata [24]. This logic is in effect a relational version of Lukasiewicz’, in that formulas
roughly similar to those of \( R \) are treated as atoms of a propositional calculus.
The authors provide axiom-schemata which, together with the usual axioms of
propositional logic, yield a complete proof-system for the language in question.
Actually, the propositional atoms in this language are allowed to feature \( n \)-ary
predicates for all \( n \geq 1 \). However, the rather strange restrictions on quantifier-scope
(existentials must always outscope universals), mean that this language is primarily
of interest for atoms featuring only unary and binary predicates; these atoms (and
their negations) then essentially correspond the formulas of our fragment \( R \).
CHAPTER 7

Comparative Adjectives

This chapter proposes and studies logical systems in which one can express arguments such as the following:

(7.1)

\begin{align*}
\text{Every giraffe is taller than every gnu} \\
\text{Some gnu is taller than every lion} \\
\text{Some lion is taller than some zebra} \\
\text{Every giraffe is taller than some zebra}
\end{align*}

We encourage the reader to work out an argument showing that the conclusion of (7.1) follows from the premises. A picture would do. Any convincing argument must appeal in some way to the transitivity of is taller than; without transitivity, the conclusion does not follow. (The argument also depends on the subject-wide scope reading in the second sentence, and indeed all sentences in this chapter are read with the subject having wide scope.) Although transitivity is salient, it is also silent: it is not explicitly mentioned, and it even seems unnecessary to add the transitivity as an additional premise. Every comparative adjective phrase is most naturally interpreted by a transitive relation, and so this point will be of central importance in this chapter. Further, the interpretations are irreflexive, corresponding to the fact that sentences like Every woman is not taller than some woman (with the subject wide scope reading) are logical truths. (This is because for every woman \(w\), \(w\) is definitely not taller than herself.) Returning to the topic of our chapter, we wish to examine reasoning in simple fragments of language, and the transitivity will give us many sound principles for comparative adjective phrases.

We are especially interested in complete and decidable logical systems in which one can represent the main features of arguments such as (7.1). Here is a related point. One might take a formal representation of (7.1) to just be a translation into first-order logic, together with a proof of the conclusion from the hypotheses in some formal system or other. For the purposes of this chapter, this will not do: we are interested only in decidable logical systems. A more sophisticated point might be to translate (7.1) into the two-variable fragment \(\text{FO}^2\) of first-order logic. This fragment is indeed decidable, by Mortimer's Theorem [17]. But the requirement of transitivity cannot be expressed in \(\text{FO}^2\), as shown in Theorem 20 of Purdy [30]. And a related point: \(\text{FO}^2\) enriched by atomic sentences saying \(R\) is a transitive relation is also undecidable, by a result in Grädel et al [8].

Furthermore, the same premises in (7.1) also entail Everything which is taller than some giraffe is taller than some zebra. This last sentence is important due to the relative clause in the subject. For another example, consider:

(7.2)

\begin{align*}
\text{Every hyena is taller than some jackal} \\
\text{Everything taller than some jackal is not heavier than any warthog} \\
\text{Everything which is taller than some hyena is not heavier than any warthog}
\end{align*}
The reader wishing to get an idea of the subject is encouraged to devise a logical system in which one can carry out the reasoning in (7.1) and (7.2) and which has no individual variables.

Our plan in this chapter is not to propose any new logical languages, but rather to use the languages $\mathcal{R}$ and $\mathcal{RC}$ which we have seen in Chapter 5. Recall that these languages have “binary atoms” $r$ which in $\mathcal{R}$ and $\mathcal{RC}$ were taken to correspond to English transitive verbs (verbs taking direct objects). The difference is that $\mathcal{RC}$ allows relative clauses in subject NP’s, a point hinted at above. That is, the sentences in (7.1) are formalizable in $\mathcal{R}$, while those in (7.2) require the larger machinery of $\mathcal{RC}$. Previously, binary atoms were interpreted as arbitrary relations on some domain. We saw logical completeness results for semantic interpretations where the verbs were interpreted as arbitrary relations. To pursue the topic of this chapter, we therefore restrict attention to interpretations where binary atoms are required to be interpreted by transitive and irreflexive relations. This restriction means that more logical laws will be valid. This holds for both $\mathcal{R}$ and $\mathcal{RC}$. The point of the chapter is to characterize these valid inference patterns by syllogistic proof systems.

We find it convenient to separate transitivity from irreflexivity in our work. Accordingly, we study (i) $\mathcal{R}$ interpreted on transitive models (Sections 7.1.2 and 7.1.1), (ii) $\mathcal{RC}$ on transitive models (Section 7.2.1), and (iii) $\mathcal{RC}$ on transitive and irreflexive models (Section 7.2.2).

Incidentally, interpreting $\mathcal{R}$ on transitive and irreflexive relations immediately gives an axiom of infinity: if every child is taller than some child, and there is some child to begin with, then there are infinitely many children. The model construction in Section 7.2.2 reflects this. To the best of my knowledge, this is the first time infinite models have turned up in the study of syllogistic systems.

Relation to other work. The model construction in Section 7.1.2 is due to Ziegler [40]. Specifically, the relations in Figure 5.3 are essentially from [40].

The two logical languages studied in this chapter are the languages $\mathcal{R}$ and $\mathcal{RC}$ introduced in [29]. Although we shall use the same formal languages, we gloss them in English in a different way. In previous chapters of these notes (starting with Chapter 5), we took the binary atoms to represent English transitive verbs, in this chapter we take them to be comparative adjective phrases.

### 7.1. Requiring Transitivity in $\mathcal{R}$

The first system in this chapter is an extension of $\mathcal{R}$ which enforces transitivity of the relation. We say “the” relation, because in languages like $\mathcal{R}$, every sentence has at most one binary atom. Thus by Craig Interpolation Theorem (for example), if $\Gamma \models \varphi$, then $\varphi$ indeed follows semantically from the sentences in $\Gamma$ with the same binary atom (or no atom). So we might as well formulate $\mathcal{R}$ in terms of a single binary atom. (None of these points hold for $\mathcal{RC}$ studied below; in that system one may reason about several relations.)

#### 7.1.1. $\mathcal{R}(tr)$

In Figure 7.1 we introduce a set of eight rules $\mathcal{R}(tr)$ which are easily seen to be sound for transitive interpretations. These all make intuitively clear points that reflect transitivity. For example, ($\rho6$) may be exemplified as follows: if some girl is taller than all boys, and some boy is taller than some teacher, then some girl is taller than some teacher.
7.1. REQUIRING TRANSITIVITY IN $R$

\[
\begin{align*}
\forall(x, \forall(y, r)) &\quad \exists(y, \forall(z, r)) & (\rho_1) &\quad \forall(x, \forall(y, r)) &\quad \exists(y, \exists(z, r)) & (\rho_2) \\
\forall(x, \exists(y, r)) &\quad \forall(y, \exists(z, r)) & (\rho_3) &\quad \forall(x, \exists(y, r)) &\quad \forall(y, \forall(z, r)) & (\rho_4) \\
\exists(x, \forall(y, r)) &\quad \exists(y, \forall(z, r)) & (\rho_5) &\quad \exists(x, \forall(y, r)) &\quad \exists(y, \exists(z, r)) & (\rho_6) \\
\exists(x, \exists(y, r)) &\quad \forall(y, \forall(z, r)) & (\rho_7) &\quad \exists(x, \exists(y, r)) &\quad \forall(y, \exists(z, r)) & (\rho_8)
\end{align*}
\]

**Figure 7.1.** Additions to $R$ to get the system $R(tr)$.

We should mention that some variations on the rules in Figure 7.1 which appear to be sound really are unsound. For example, consider

\[
\forall(x, \forall(y, r)) \quad \forall(y, \forall(z, r))
\]

If there is one $x$, one $y$, and no $z$, then the premises hold vacuously. But the conclusion fails when the $x$ does not have the relation $r$ to the $z$. Indeed worries about the emptiness of interpretations are the main source of complications in proofs in this subject.

We use $R(tr)$ for the set of rules containing those in Figures 5.2 and 7.1. And, as earlier in these lecture notes, we also use the same notation for the derivation relation determined by this set of rules.

We shall prove the completeness result for $R(tr)$ in Theorem 7.2 below. Before starting in on that, let us check that the system can represent valid arguments such as that found in (7.1) and repeated below:

- Every giraffe is taller than every gnu
- Some gnu is taller than every lion
- Some lion is taller than some zebra
- Every giraffe is taller than some zebra

Here is the relevant derivation:

\[
\begin{align*}
\forall(giraffe, \forall(gnu, taller)) &\quad \exists(gnu, \forall(lion, taller)) & (\rho_1) &\quad \exists(lion, \exists(zebra, taller)) & (\rho_2) \\
\forall(giraffe, \forall(lion, taller)) &\quad \exists(zebra, taller)
\end{align*}
\]

7.1.2. Model construction. We shall show that every set $\Gamma$ of sentences which is consistent in $R(tr)$ is satisfiable on a transitive model. The proof is nearly the same as the one we saw earlier for $R$, except that we need the additional information that the model which we construct is actually transitive. This is the content of the following result.
Let \( \Gamma \) be complete. Then \([r]\) is transitive in \( \mathcal{M}(\Gamma) \).

**Proof** We are first going to check the transitivity of \([r]\) when restricted to the elements \( x_i \); after this we shall consider also the pair-elements \( \{p, q\} \).

The idea is to look at all of the possible ways that instances of our six types of diagrams in Figure 5.3 can appear next to each other. We need not worry about \( (f) \), but we still must examine \( 5 \times 5 = 25 \) cases. These are shown in the table below:

<table>
<thead>
<tr>
<th>( (a) )</th>
<th>( (b) )</th>
<th>( (c) )</th>
<th>( (d) )</th>
<th>( (e) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_4, \exists )</td>
<td>( p_3, p_5, \exists )</td>
<td>( p_1, \exists )</td>
<td>( p_3, \exists )</td>
<td>( p_2 )</td>
</tr>
<tr>
<td>( p_4 )</td>
<td>( p_3, p_5 )</td>
<td>( p_5 )</td>
<td>( p_3 )</td>
<td>( p_6 )</td>
</tr>
<tr>
<td>( p_5, \exists )</td>
<td>( p_5 )</td>
<td>( p_5 )</td>
<td>( p_6, \exists )</td>
<td>( p_6 )</td>
</tr>
<tr>
<td>( p_4 )</td>
<td>( p_3 )</td>
<td>( - )</td>
<td>( p_3 )</td>
<td>( - )</td>
</tr>
<tr>
<td>( p_2, p_7, \exists )</td>
<td>( p_2, p_7 )</td>
<td>( - )</td>
<td>( - )</td>
<td>( - )</td>
</tr>
</tbody>
</table>

The entry in the \( i \)th row and \( j \)th column indicates which rule(s) are needed in the verification of transitivity when an instance of the \( i \)th letter of the alphabet is placed on the immediate left of an instance of the \( j \)th letter. A dash \( - \) in the table means that no instances of transitivity arise. For the other entries, it is best to give some examples; these also explain the notation. Consider first an instance of \( (c) \) to the left of an instance of \( (b) \), shown on the left below:

\[
(7.3) \quad \begin{array}{c}
\downarrow \quad \downarrow \\
\downarrow \quad \downarrow \\
\downarrow \quad \downarrow \\
\end{array}
\]

We need to check that \( x_2 \rightarrow z_1 \) and \( x_2 \rightarrow z_2 \). But we have \( \Gamma \vdash \exists (x, \exists (y, r)) \); and also \( \Gamma \vdash \exists (y, \forall (z, r)) \). We use \( (\rho_5) \) to see that \( \Gamma \vdash \exists (x, \exists (z, r)) \). This implies that \( x_2 \rightarrow z_1 \) and \( x_2 \rightarrow z_2 \), as desired.

Next, consider an instance of \( (c) \) to the left of an instance of \( (d) \); this is shown in the right of \( (7.3) \) above. This time, \( \Gamma \vdash \exists (x, \forall (y, r)) \) and also \( \Gamma \vdash \forall (y, \exists (z, r)) \). It does not follow from these alone that \( \Gamma \vdash \exists (x, \exists (z, r)) \). However, since the \( y \) points belong to the model, \( \Gamma \vdash \exists (y, y) \). So \( \Gamma \vdash \exists (y, \exists (z, r)) \). Now we may use \( (\rho_6) \) to see that \( \Gamma \vdash \exists (x, \exists (z, r)) \). (In the chart, the steps of the argument where we need to know that \( \Gamma \) derives the existence of some object are indicated with a \( \exists \) symbol.)

We have only given two of the 25 cases, but the rest are very similar. It remains to verify transitivity when one or two pair-elements \( \{p, q\} \) are involved. Here is one of the many similar verifications. Suppose that \( u_2 \rightarrow \{p, q\} \rightarrow w_1 \). There are several ways that this could happen, and to be concrete, let us assume that from \( \Gamma \) we have \( \exists (u, \forall (z, r)), \exists (p, q) \), and \( \forall (q, \exists (w, r)) \). From the first two of these, we have \( \exists (u, \exists (q, r)) \). Then with the last we indeed have \( \exists (u, \forall (w, r)) \). This means that \( u_2 \rightarrow w_1 \), just as desired.

All of the remaining points are similar. \( \square \)

And as in Theorem 5.10, we have the following result:

**Theorem 7.2** For \( \Gamma \cup \{\varphi\} \subseteq \mathcal{R} \), \( \Gamma \models \varphi \) on transitive models iff \( \Gamma \vdash \varphi \) in \( \mathcal{R}(\text{tr}) \).

### 7.2. Requiring Transitivity and Irreflexivity in \( \mathcal{R} \)

Section 7.1 studied what happens when one requires the interpretations of binary atoms in \( \mathcal{R} \) to be transitive relations. An expansion of \( \mathcal{R} \) to a language \( \mathcal{R} \)
was introduced in Pratt-Hartmann and Moss [29]. That chapter also axiomatized \( \mathfrak{R}C \) by the logical system called RC which is shown in Figure 7.2. We therefore continue this chapter by studying the effect of transitivity in the models of \( \mathfrak{R}C \). The effect is to add the additional set of rules shown in Figure 6.1. We prove the completeness of the resulting system in Section 7.2.1 below.

### 7.2.1. RC(tr)

We use \( \text{RC}(\text{tr}) \) for the set of rules in Figures 6.1 and 7.2, and we use the same symbol for the logical system associated to it. The transitivity rules of \( \text{RC}(\text{tr}) \) found in Figure 7.2 are similar to the ones we have seen in \( R(\text{tr}) \).

One should check the soundness of (tr1) – (tr4) for transitive models, and we omit these routine details.

Our formulation of (RAA) for \( R \) must be more general than the one for \( \mathfrak{R} \), since we now allow contradictions of the form \( \exists c, c \), where \( c \) might be a (complex) \( c \)-term.

Here is an example of a derivation in \( \text{RC}(\text{tr}) \), corresponding to the informal example presented in (7.2):

\[
\forall (\text{hyena}, \exists (\text{jackal}, \text{taller})) \\
\forall(\exists (\text{hyena}, \text{taller}), \exists (\text{jackal}, \text{taller})) \quad \text{(tr1)} \\
\forall(\exists (\text{jackal}, \text{taller}), \forall (\text{warthog}, \text{heavier})) \\
\forall (\exists (\text{hyena}, \text{taller}), \forall (\text{warthog}, \text{heavier})) \quad \text{(B)}
\]

The application of (tr1) corresponds to using the premise every hyena is taller than some jackal to derive the sentence everything which is taller than some hyena is taller than some jackal. The second step corresponds to the transitivity of predication (is).

Our completeness result is that the logical system \( \text{RC}(\text{tr}) \) is complete for the class of transitive models. The proof is a modification of the corresponding result for \( \mathfrak{R}C \) in [29]. That result showed that \( \mathfrak{R}C \) is complete for the class of all models. Rather than reproduce all of the details, we review the main idea and quote without proofs two lemmas from [29]. In the sequel, the variables \( b, c \) range over positive \( c \)-terms, and \( d \) over the larger class of all \( c \)-terms. As earlier, \( p, q, x, y, \) and \( z \) range over unary atoms and \( r \) over binary atoms.

The relation between \( \text{RC}(\text{tr}) \) and \( R(\text{tr}) \). We have already seen that RC derives all of the rules in \( R \). We mention here one point not in [29], the reasoning behind
a derivation of (D3) in Figure 5.2. A derivation may be found on the left below:

\[
\frac{\exists(p,c) \quad \forall(p,\overline{\tau})}{\exists(p,\overline{\tau})} \quad \text{(D1)}
\]

Our application of (D1) uses the identification of \( \forall(p,\overline{\tau}) \) with \( \forall(c,\overline{p}) \).

This observation that RC derives the rules in \( R \) extends: the eight rules in Figure 7.1 are derivable in \( \text{RC}(tr) \). On the right above, we have a derivation of (\( \rho 3 \)). The derivations of all the other rules in Figure 7.1 are similar.

We should also add that \( \text{RC}(tr) \) derives versions of (\( \rho 1 \))–(\( \rho 8 \)) in which the subject noun phrase might contain a \( c \)-term.

**Theorem 7.3**  For \( \Gamma \cup \{ \varphi \} \subseteq \mathcal{R}C \), \( \Gamma \models \varphi \) on transitive models iff \( \Gamma \vdash \varphi \) in \( \text{RC}(tr) \).

The rest of this section is devoted to the proof of Theorem 7.3. It is based on the proof of of Theorem 6.1, the same result for \( \mathcal{R}C \) but without any transitivity assumptions in the semantics. Fortunately, the same proof idea works. (This contrasts with our work earlier in the chapter; the constructions in Section 7.1 were new, and they made for a much longer argument than what we shall see below.) We need only show that every consistent set \( \Gamma \) in \( \text{RC}(tr) \) is satisfiable on a transitive model. Also, by Lemma 5.9, we may assume that \( \Gamma \) is \( \text{RC}(tr) \)-complete; that is, every sentence or its negation belongs to \( \Gamma \).

We construct a model \( M \) and prove that it satisfies \( \Gamma \). Since we need the details on this, we must review the construction. First, let \( C^+ \) be the set of positive \( c \)-terms. Then we define \( M \) by setting:

\[
M = \{ \langle c_1, c_2, Q \rangle \in C^+ \times C^+ \times \{ \forall, \exists \} : \Gamma \vdash \exists(c_1, c_2) \} \quad (7.4)
\]

\[
[p] = \{ \langle c_1, c_2, Q \rangle \in M : \Gamma \vdash \forall(c_1, p) \text{ or } \Gamma \vdash \forall(c_2, p) \} \quad (7.5)
\]

\[
[d_1, d_2, Q_2] \text{ if and only if either:}
\]

(a): for some \( i, j \) and \( q \in P \), \( \Gamma \vdash \forall(c_i, \forall(q, r)) \) and \( \Gamma \vdash \forall(d_j, q) \); or

(b): \( Q_2 = \exists \), and for some \( i \) and \( q \in P \), \( d_1 = d_2 = q \), and \( \Gamma \vdash \forall(c_i, \exists(q, r)) \).

One then checks easily that \( A \) is non-empty (using (\( W \))).

**Lemma 7.4**  For all \( c \in C^+ \),

\[
[c] = \{ \langle d_1, d_2, Q \rangle \in M : \text{either } \Gamma \vdash \forall(d_1, c), \text{ or } \Gamma \vdash \forall(d_2, c) \}.
\]

**Lemma 7.5**  \( M \models \Gamma \).

**Lemma 7.6**  The interpretation of each binary atom \( r \) is transitive.

**Proof**  Assume that

\[
\langle b_1, b_2, Q_1 \rangle [r] \langle c_1, c_2, Q_2 \rangle [r] \langle d_1, d_2, Q_3 \rangle.
\]

We shall use several times the fact that since \( \langle c_1, c_2, Q_2 \rangle \) belongs to the model, \( \Gamma \vdash \exists(c_1, c_2) \). We have four cases.
7.2. Requiring Transitivity and Irreflexivity in \( \mathcal{R} \)

Case 1: for some \( i, j, k, l, \) and \( q_1, q_2 \in \mathbf{P} \), \( \Gamma \vdash \forall(b_i, \forall(q_1, r)) \), \( \Gamma \vdash \forall(c_j, q_1) \), \( \Gamma \vdash \forall(c_k, \forall(q_2, r)) \), and \( \Gamma \vdash \forall(d_l, \forall(q_2, r)) \). We have \( \Gamma \vdash \exists(c_j, c_k) \) also. Now we have derivation from \( \Gamma \):

\[
\begin{align*}
\vdots & \\
\forall(c_k, \forall(q_2, r)) & \quad \vdots \\
\vdots & \\
\forall(c_j, q_1) & \quad \exists(c_j, c_k) \\
\exists(q_1, \forall(q_2, r)) & \quad (\text{tr3}) \\
\forall(q_1, r), \forall(q_2, r) & \quad (\text{B}) \\
\forall(b_i, \forall(q_1, r)) & \quad (\Gamma) \\
\end{align*}
\]

And now we see that \( \langle b_1, b_2, Q_1 \rangle[r] \langle d_1, d_2, Q_3 \rangle \), using alternative (a) in the definition of \( [r] \).

Case 2: for some \( i, j, \) and \( q_1 \in \mathbf{P} \), \( \Gamma \vdash \forall(b_i, \forall(q_1, r)) \) and \( \Gamma \vdash \forall(c_j, q_1) \); \( Q_3 = \exists, \) and for some \( k \) and \( q_2 \in \mathbf{P} \), \( d_1 = d_2 = q_2 \), and \( \Gamma \vdash \forall(c_k, \exists(q_2, r)) \). This time, we have \( \Gamma \vdash \exists(q_1, c_k) \), and by a derivation similar to what we saw in Case 1, \( \Gamma \vdash \forall(b_i, \exists(q_2, r)) \). In this case, alternative (b) shows that \( \langle b_1, b_2, Q_1 \rangle[r] \langle d_1, d_2, Q_3 \rangle \).

Case 3: \( Q_2 = \exists, \) and for some \( i \) and \( q_1 \in \mathbf{P} \), \( c_1 = c_2 = q_1, \) and \( \Gamma \vdash \forall(b_i, \exists(q_1, r)) \); and also \( Q_3 = \exists, \) and for some \( j, k, \) and \( q_2 \in \mathbf{P} \), \( d_1 = d_2 = q_2 \), and \( \Gamma \vdash \forall(c_j, \exists(q_2, r)) \). This time we see that \( \Gamma \vdash \forall(b_i, \exists(q_2, r)) \), and we use (b) to see that \( \langle b_1, b_2, Q_1 \rangle[r] \langle d_1, d_2, Q_3 \rangle \).

This completes the proof. \( \square \)

And with the work of Lemma 7.6 done, we have also completed the proof of Theorem 7.3.

7.2.2. Irreflexivity: the system \( \mathbf{R}^\ast(\text{tr, irr}) \). We now continue the study of the syllogistic logic of comparative adjectives by adding a requirement to the semantics that the interpretation of each binary atom be irreflexive. Notice that as soon as we add this requirement, the logic no longer has the finite model property. Specifically, \( \forall(p, \exists(p, r)) \) is satisfiable, but not by any finite (transitive and irreflexive) model. (In Section 7.2.3 below, we discuss what happens if we want to restrict to transitive, irreflexive, and finite models.)

The additional requirement also results in the soundness of the following irreflexivity rule

\[
\forall(p, \exists(p, r)) \quad (\text{Irr})
\]

This looks like a very weak consequence of irreflexivity, so perhaps it will be surprising that it leads to a complete system. The point, perhaps, is that without variables there is not much that one can actually do with irreflexivity besides (Irr) and its consequences.

Let \( \mathbf{R}^\ast(\text{tr, irr}) \) be the logical system whose rules are those of Figures 6.1 and 7.2, and also (Irr).

**Theorem 7.7** For \( \Gamma \cup \{ \varphi \} \subseteq \mathcal{R} \), \( \Gamma \models \varphi \) on transitive, irreflexive models iff \( \Gamma \vdash \varphi \) in \( \mathbf{R}^\ast(\text{tr, irr}) \).
We write a model some formulas which follow from these, such as \( \forall c_1, c_2, n \in C^+ \times C^+ \times N : \Gamma \vdash \exists (c_1, c_2), \)

(7.7) \[
M = \{\langle c_1, c_2, n \rangle \in C^+ \times C^+ \times N : \Gamma \vdash \exists (c_1, c_2), \text{and if } n \geq 0, \text{then } c_1 = c_2, \text{ and } c_1 \text{ is a unary atom} \} \\
[p] = \{\langle c_1, c_2, n \rangle \in M : \Gamma \vdash \forall (c_1, p) \text{ or } \Gamma \vdash \forall (c_2, p) \}
\]

Additional notation. Before going further we need some extra relations on \( C^+ \).

We write \( c \supseteq d \) if \( \Gamma \vdash \exists (c, c) \) and also \( \Gamma \vdash \exists (d, d) \), \( d \) is a unary atom, and \( \Gamma \vdash \forall (c, \exists (d, r)) \). The relation \( \supseteq \) is not reflexive, but it is transitive. We write \( c \sqsupseteq d \) for \( c \subseteq d \sqsubseteq c \). We also write \( c \sqsupseteq d \) if \( c \supseteq d \) and not \( c \equiv d \). We write \( c \Rightarrow d \) if if \( \Gamma \vdash \exists (c, c) \) and also \( \Gamma \vdash \exists (d, d) \), \( d \) is a unary atom, and \( \Gamma \vdash \forall (c, \forall (d, r)) \). Notice that if \( c \Rightarrow d \) and \( d \) is an atom, then \( c \sqsupseteq d \). (This uses \( (\rho 7) \) and \( (\text{Irr}) \).) And recall the convention to write \( c \leq d \) for \( \Gamma \vdash \forall (c, d) \).

The interpretation of binary atoms. We say \( \langle c_1, c_2, n \rangle \vdash \langle d_1, d_2, m \rangle \) if and only if one of the conditions below holds:

- (a): for some \( i \) and \( j \) and some unary atom \( p \), \( c_i \Rightarrow p \) and \( d_j \leq p \).
- (b): \( d_1 = d_2 \) is a unary atom \( p \), and for some \( i \), \( c_i \equiv p \) and \( m > 0 \).
- (c): \( d_1 = d_2 \) is a unary atom \( p \), and for some \( i \), \( c_i \equiv p \) and \( m > n \).

Note that the conditions above are not exclusive.

EXAMPLE 7.8 Suppose \( \Gamma \) has \( \exists (p, p) \), \( \exists (q, q) \), \( \exists (p, q) \), \( \forall (p, \exists (q, r)) \), \( \forall (q, \exists (p, r)) \), \( \exists (p, \exists (q, r)) \), \( \forall (\exists (p, r), q) \), \( \exists (\exists (p, r), q) \), and \( \exists (\exists (p, r), q) \). (We have again left off some formulas which follow from these, such as \( \forall (p, \exists (p, r)) \).) To save on some notation, we’ll abbreviate \( \exists (p, r) \) by \( c \) and \( \exists (q, r) \) by \( d \). Then \( M \) contains \( \langle p, p, n \rangle \) and \( \langle q, q, n \rangle \) for \( n \geq 0 \), and to save on notation, we write these as \( p_n \) and \( q_n \). \( M \) also contains \( \langle p, c, 0 \rangle \), \( \langle c, c, 0 \rangle \), \( \langle q, d, 0 \rangle \), \( \langle d, d, 0 \rangle \), and \( \langle q, q, 0 \rangle \). Then \( \langle M \rangle \) contains \( \langle p, c, 0 \rangle \), \( \langle q, q, 0 \rangle \), and all \( p_n : \langle q \rangle \) is similar. Moreover, \( \langle r \rangle \) makes \( \langle M \rangle \) look as follows:

\[
\langle c, c, 0 \rangle \;
\langle q, c, 0 \rangle \;
\langle p, c, 0 \rangle \;
\langle p, q, 0 \rangle \\
\langle d, d, 0 \rangle \;
\langle q, d, 0 \rangle \;
\langle q, q, 0 \rangle \\
\langle p, q \rangle \;
\langle p, n \rangle \rightarrow \cdots \rightarrow \langle p, n+1 \rangle \\
\langle q, q \rangle \rightarrow \cdots \rightarrow \langle q, q+1 \rangle \\
\langle p, q \rangle \\
\langle p, n \rangle \\
\langle q, q \rangle \\
\langle p, n+1 \rangle \\
\langle q, q+1 \rangle
\]

The relation in the model is actually the transitive closure of the arrows above.

LEMMA 7.9 The interpretation \( \langle r \rangle \) of each binary atom \( r \) is transitive.

PROOF Assume that \( \langle b_1, b_2, n_1 \rangle \vdash \langle c_1, c_2, n_2 \rangle \vdash \langle d_1, d_2, n_3 \rangle \), and write these three triples as \( b, c, \) and \( c, \) respectively.

Case 1: \( b \vdash c \) via \( (a) \), \( c \vdash d \) via \( (a) \). Let \( i, j, p, \) and \( q \) be such that \( b_i \Rightarrow p, c_j \leq p, c_k \Rightarrow q, \) and \( d_l \leq q \). See Case 1 of Lemma 7.6.

Case 2: \( b \vdash c \) via \( (a) \), \( c \vdash d \) via \( (b) \). This time, \( d_1 = d_2 \) is an atom, and we have \( i, j, k \) and \( p \) so that \( b_i \Rightarrow p, c_j \leq p, c_k \equiv d \). Also, \( m_3 > 0 \). The derivation
below shows that $b_i \sqsupset d$:

$$
\begin{align*}
& \vdots \\
& \forall(b_i, \forall(p, r)) \quad \exists(c_j, c_k) \\
& \forall(c_k, \exists(d, r)) \quad \exists(p, c_k) \\
& \vdash \exists(p, \exists(d, r))
\end{align*}
$$

The rule we are quoting as \((\rho2)\) here is more general than the rule we used earlier in connection with $\mathcal{RC}$ in that we are allowing the subject noun phrase to be complex. But the form above is again derivable in two steps using \((tr\ 2)\) and \((B)\). The same comments apply below in several other cases of this proof.) We claim now that $b_i \sqsupset d$. (For suppose that $d \sqsupset b_i$. Since $b_i \implies p$ and $\exists(p, c_k)$ we see that $b_i \sqsupset c_k$. By transitivity, we have $d \sqsupset c_k$. This contradicts $c_k \sqsupset d$.) Recalling that $m_3 > 0$, we see that $b[r]d$ via alternative (b) in the definition of \([r]\).

**Case 3**: $b[r]c$ via (a), and $c[r]d$ via (c). The argument is as in Case 2.

**Case 4**: $b[r]c$ via (b), and $c[r]d$ via (a). Write $c$ for $c_1 = c_2$; this is an atom. And the same holds for $d$. We have $i$ and $j$ and $p$ such that $b_i \sqsupset c$, $c \implies p$, $p \leq d_j$. (Also, $n_2 > 0$, but this is irrelevant.) Using \((\rho4)\), we see that $b_i \sqsupset p$. It follows from this that $b[r]d$ via alternative (a) in the definition of \([r]\).

**Case 5**: $b[r]c$ via (b), and $c[r]d$ via (b). Write $c$ for $c_1 = c_2$; this is an atom. And the same holds for $d$. For some $i$, we have $b_i \sqsupset c \sqsupset d$. Using \((\rho3)\), we see that $b_i \sqsupset d$. Also, $m_3 > 0$. It now follows that $b[r]d$ via alternative (b) in the definition of \([r]\).

**Case 6**: $b[r]c$ via (b), and $c[r]d$ via (c). The argument is as in Case 5. (The only difference is that this time we have $b_i \sqsupset c \sqsupset d$. But this is sufficient to imply $b_i \sqsupset d$.)

**Case 7**: $b[r]c$ via (c), and $c[r]d$ via (a). The argument is as in Case 4.

**Case 8**: $b[r]c$ via (c), and $c[r]d$ via (b). The argument is as in Case 5.

**Case 9**: $b[r]c$ via (c), and $c[r]d$ via (c). As in Case 5, we write $c$ and $d$ for the evident atoms. For some $i$, we have $b_i \equiv c \equiv d$, and we also have $n_1 > n_2 > n_3$. Since the order on natural numbers is transitive, $n_1 > n_3$. Thus $b[r]d$ via alternative (c) in the definition of \([r]\).

This concludes the proof.

**Lemma 7.10** The interpretation \([r]\) of each binary atom $r$ is irreflexive.

**Proof** Suppose towards a contradiction that

$$(7.8) \quad \langle c_1, c_2, n \rangle [r] \langle c_1, c_2, n \rangle.$$

We have $\Gamma \vdash \exists(c_i, c_j)$ for all $i$ and $j$. The reason for (7.8) cannot be (a) in the definition of \([r]\) because if it were, we would have $\Gamma \vdash \forall(c_i, (\forall(c_j, r))$ for some $i$ and $j$. With the irreflexivity rule (Irr), this would contradict the consistency of $\Gamma$.

Suppose that the reason for (7.8) were (b). Then $c_1 = c_2$ is a unary atom, say $c$, and $c \sqsupset c$. But this is impossible: if $c \sqsupset c$, then $c \equiv c$.

Finally, it is impossible that (c) be the reason for (7.8), since $n \not\geq n$. \(\square\)
Lemmas 7.9 and 7.10 show that \( \mathcal{M} \) is a transitive and irreflexive model for the interpretation of our fragment. Recall that our completeness proof for the logic \( R^*(\text{tr, irr}) \) on the intended class reduces to showing that every complete, consistent set \( \Gamma \) has a transitive and irreflexive model. The next two results show that \( \mathcal{M} \) serves as such a model.

**Lemma 7.11** For all \( c \in C^+ \), \([c] = \{ \langle d_1, d_2, Q \rangle \in M : \text{either } \Gamma \vdash \forall(d_1, c), \text{ or } \Gamma \vdash \forall(d_2, c) \} \).

**Lemma 7.12** \( \mathcal{M} \models \Gamma \).

The proofs of Lemmas 7.11 and 7.12 are nearly the same as that of Lemmas 5.3 and 5.4 in [29], so we shall not reproduce them. The main notational change is that where [29] mentions an element of the model such as \( \langle c, c, v \rangle \), we would use the element \( \langle c, c, 0 \rangle \) in this chapter.

The foregoing work proves Theorem 7.7.

**7.2.3. Finite models.** The requirement of transitivity alone does not allow us to state an axiom of infinity: the proof of Theorem 7.3 shows that every finite consistent set in \( \mathcal{R}(\text{tr}) \) has a finite model. However, adding to transitivity the further requirement of irreflexivity leads to theories with only infinite models, such as \( 3(p, p), \forall(p, 3(p, r)) \).

Therefore, if we want to axiomatize the finite models, then we must also add something to the logic. Here is one way to do that, in the form of an axiom:

\[
\frac{\exists(p, p)}{\exists(p, \forall(p, \neg r))} \text{ (Fin)}
\]

Let \( \mathcal{R}(\text{tr, irr, fin}) \) be the logic which adds this new axiom to \( R^*(\text{tr, irr}) \).

**Theorem 7.13** For \( \Gamma \cup \{ \varphi \} \subseteq \mathcal{R}_c \), \( \Gamma \models \varphi \) on transitive, irreflexive and finite models iff \( \Gamma \vdash \varphi \) in \( \mathcal{R}(\text{tr, irr, fin}) \).

**Proof** We use the same construction as in Theorem 7.7 but with one change. The definition of the model \( \mathcal{M} \) began in (7.7), and of course the structure is infinite. To keep straight the old and new models, we'll now define a model to be called \( N \).

The domain of the model is the set \( N \) given by

\[
(7.9) \quad N = \{ \langle c_1, c_2, n \rangle \in C^+ \times C^+ \times \{0, 1\} : \Gamma \vdash 3(c_1, c_2), \\
\text{and if } n \geq 0, \text{ then } c_1 = c_2, \text{ and } c_1 \text{ is a unary atom} \}
\]

The rest of the structure is defined in the same way as with \( \mathcal{M} \): the unary atoms are interpreted as in (7.7), and the binary ones from the three alternatives (a)–(c) stated before Lemma 7.9. This way, \( N \) is a submodel of \( \mathcal{M} \), so it is transitive and irreflexive; clearly it is also finite; this is the point of changing \( N \) to \( \{0, 1\} \). The only point to check is that the analog of Lemma 7.11 goes through. And for this, the only point to check is that for \( c \) a positive c-term of the form \( 3(p, r) \),

\[
[c] = \{ \langle d_1, d_2, n \rangle \in N : \text{either } \Gamma \vdash \forall(d_1, c), \text{ or } \Gamma \vdash \forall(d_2, c) \}.
\]

And here, only half needs to be checked. Suppose that \( \langle d_1, d_2, n \rangle \in N \) and \( \Gamma \vdash \forall(d_3, 3(p, r)) \). We need to see that \( \langle d_1, d_2, n \rangle \in \underline{3(p, r)} \). That is, we need some \( c \in [p] \) such that \( \langle d_1, d_2, n \rangle [r] c \). We claim that \( c = \langle p, p, 1 \rangle \) works using alternative (b) in the definition of \([r]\). We have \( d \sqsupseteq p \), and we only need to see that the converse assertion \( p \sqsupseteq d \) does not hold. Suppose toward a contradiction that it did.
Then $d$ would be a unary atom, and also we would have $\forall(p, \exists(p, r))$ using $(\rho3)$. But this directly contradicts the finiteness axiom.

The remaining verifications are as above. □

Source for this chapter. The material in this chapter comes from Moss [20].
CHAPTER 8

Logic Beyond the Aristotle Border

There are only two languages in this chapter (actually they are families of languages parameterized by sets of basic symbols): the language $\mathcal{R}C^\dagger$ which we have seen already, and an extension $\mathcal{L}(adj)$ studied in Section 8.2. We reformulate $\mathcal{R}C^\dagger$ a bit, adding constants and also being allowing for recursive constructs. To avoid confusion, we do not speak of $\mathcal{R}C^\dagger$ but instead call the language of this chapter $\mathcal{L}$. $\mathcal{L}$ is based on three pairwise disjoint sets called $P$, $R$, and $K$. These are called unary atoms, binary atoms, and constant symbols.

8.1. Fitch-style Proof System for $\mathcal{R}C^\dagger$

We review the syntax of $\mathcal{L}$ in Figure 8.1. Sentences are built from constant symbols, unary and binary atoms using an involutive symbol for negation, a formation of set terms, and also a form of quantification. The second column indicates the variables that we shall use in order to refer to the objects of the various syntactic categories. Because the syntax is not standard, it will be worthwhile to go through it slowly and to provide glosses in English for expressions of various types.

One might think of the constant symbols as proper names such as John and Mary. The unary atoms may be glossed as one-place predictates such as boys, girls, etc. And the relation symbols correspond to transitive verbs (that is, verbs which take a direct object) such as likes, sees, etc. They also correspond to comparative adjective phrases such as is bigger than. (However, later on in Section 8.2, we introduce a new syntactic primitive for the adjectives.)

Unary atoms appear to be one-place relation symbols, especially because we shall form sentences of the form $p(j)$. However, we do not have sentences $p(x)$, since we have no variables at this point in the first place. Similar remarks apply to binary atoms and two-place relation symbols. So we chose to change the terminology from relation symbols to atoms.

We form unary and binary literals using the bar notation. We think of this as expressing classical negation. So we take it to be involutive, so that $\overline{\overline{p}} = p$ and $\overline{\overline{s}} = s$.

The set terms in this language are the only recursive construct. If $b$ is read as boys and $s$ as sees, then one should read $\forall(b, s)$ as sees all boys, and $\exists(b, s)$ as sees some boys. Hence these set terms correspond to simple verb phrases. We also allow negation on the atoms, so we have $\forall(b, \overline{s})$; this can be read as fails to see all boys, or (better) sees no boys or doesn’t see any boys. We also have $\exists(b, \overline{s})$, fails to see some boys. But the recursion allows us to embed set terms, and so we have set terms like $\exists(\forall(\forall(b, \overline{s}), h), a)$ which may be taken to symbolize a verb phrase such as admires someone who hates everyone who does not see any boy.
We should note that the relative clauses which can be obtained in this way are all “missing the subject”, never “missing the object”. The language is too poor to express predicates like $\lambda x.\text{all boys see } x$.

The main sentences in the language are of the form $\forall(b, c)$ and $\exists(b, c)$; they can be read as statements of the inclusion of one set term extension in another, and of the non-empty intersection. We also have sentences using the constants, such as $\forall(g, s)(m)$, corresponding to Mary sees all girls. But we are not able to say all girls see Mary; the syntax again is too weak. (However, in our Conclusion we shall see how to extend our system to handle this.) This weakness in expressive power corresponds to a less complex decidability result, as we shall see.

Semantics. A model (for this language $L$) is a pair $M = \langle M, \llbracket \rrbracket \rangle$, where $M$ is a non-empty set, $\llbracket p \rrbracket \subseteq M$ for all $p \in P$, $\llbracket r \rrbracket \subseteq M^2$ for all $r \in R$, and $\llbracket j \rrbracket \in M$ for all $j \in K$.

Given a model $M$, we extend the interpretation function $\llbracket \rrbracket$ to the rest of the language by setting:

- $\llbracket \overline{p} \rrbracket = M \setminus \llbracket p \rrbracket$
- $\llbracket \overline{r} \rrbracket = M^2 \setminus \llbracket r \rrbracket$
- $\llbracket \exists(l, t) \rrbracket = \{x \in M : \text{ for some } y \text{ such that } \llbracket l \rrbracket(y), \llbracket t \rrbracket(x, y)\}$
- $\llbracket \forall(l, t) \rrbracket = \{x \in M : \text{ for all } y \text{ such that } \llbracket l \rrbracket(y), \llbracket t \rrbracket(x, y)\}$

We define the truth relation $\models$ between models and sentences by:

- $M \models \forall(c, d)$ if $\llbracket c \rrbracket \subseteq \llbracket d \rrbracket$
- $M \models \exists(c, d)$ if $\llbracket c \rrbracket \cap \llbracket d \rrbracket \neq \emptyset$
- $M \models c(j)$ if $\llbracket c \rrbracket(\llbracket j \rrbracket)$
- $M \models r(j, k)$ if $\llbracket r \rrbracket(\llbracket j \rrbracket, \llbracket k \rrbracket)$

If $\Gamma$ is a set of formulas, we write $M \models \Gamma$ if for all $\varphi \in \Gamma$, $M \models \varphi$.

Example 8.1 For example, look back to Example 5.2. This was the model with $M = \{w, x, y, z\}$, $\llbracket p \rrbracket = \{w, x, y\}$, and with $\llbracket s \rrbracket$ shown below.

\begin{figure}[h]
\centering
\begin{tikzpicture}
\node (w) at (0,0) {$w$};
\node (x) at (1,1) {$x$};
\node (y) at (-1,0) {$y$};
\node (z) at (1,-1) {$z$};
\draw[->] (w) to (x);
\draw[->] (w) to (y);
\draw[->] (x) to (z);
\draw[->] (y) to (z);
\end{tikzpicture}
\caption{Example model $M$.}
\end{figure}
Note at this point that \([\exists \forall (\exists(p, s), s)] = \emptyset\). We also set \([j] = w\) and \([k] = x\). We get additional sentences true in \(M\) such as \(s(j, k), \exists(k, j)\), and \(\exists(p, s)(k)\).

Here is a point that will be important later. For all terms \(c, M \models c(j)\) iff \(M \models c(k)\). (The easiest way to check this is to show that for all set terms \(c, [c] is one of the following four sets: \(\emptyset, M, \{w, x, y\}, \text{ or } \{z\}\.) However, \(M \models s(j, k)\) and \(M \models \exists(k, j)\).

The satisfiability problem for the language is decidable for a very easy reason: the language \(\mathcal{L}\) translates into the two-variable fragment \(\text{FO}^2\) of first-order logic. (We shall see this shortly.) Thus we have the finite model property (by Mortimer [17]) and decidability of satisfiability in non-deterministic exponential time (Grädel et al [7]). It might therefore be interesting to ask whether the smaller fragment \(\mathcal{L}\) is of a lower complexity. As it happens, it is. Pratt-Hartmann [25] showed that the satisfiability problem for a certain fragment \(\mathcal{E}_2\) of \(\text{FO}^2\) can be decided in \(\text{ExpTime}\) in the length of the input \(\Gamma\), and his fragment was essentially the same as the one in this chapter.

The bar notation. We have already seen that our unary and binary atoms come with negative forms. We extend this notation to all sentences in the following ways: \(\bar{p} = p, \bar{s} = s, \exists(l, r) = \forall(l, r), \forall(l, r) = \exists(l, r), \forall(c, d) = \exists(c, d), \exists(c, d) = \forall(c, d), c(j) = \tau(j), \text{ and } r(j, k) = r(j, k)\).

Translation of the syllogistic into \(\mathcal{L}\). We indicate briefly a few translations to orient the reader. First, the classical syllogistic translates into \(\mathcal{L}\):

- All \(p\) are \(q\) \(\iff\) \(\forall(p, q)\)
- No \(p\) are \(q\) \(\iff\) \(\forall(p, \bar{q})\)
- Some \(p\) are \(q\) \(\iff\) \(\exists(p, q)\)
- Some \(p\) aren’t \(q\) \(\iff\) \(\exists(p, \bar{q})\)

We can also translate \(\mathcal{L}\) to \(\text{FO}^2\), the fragment of first order logic using only the variables \(x\) and \(w\). We do this by mapping the set terms two ways, called \(c \mapsto \varphi_{c,x}\) and \(c \mapsto \varphi_{c,y}\). Here are the recursion equations for \(c \mapsto \varphi_{c,x}\):

\[
\begin{align*}
p & \mapsto P(x) & \forall(c, r) & \mapsto (\forall y)(\varphi_{c,y}(y) \rightarrow r(x, y)) \\
\bar{p} & \mapsto \neg P(x) & \exists(c, r) & \mapsto (\exists y)(\varphi_{c,y}(y) \wedge r(x, y))
\end{align*}
\]

The equations for \(c \mapsto \varphi_{c,y}\) are similar. Then the translation of the sentences into \(\text{FO}^2\) follows easily.

8.1.1. Proof System. We present our system in natural-deduction style in Figure 8.3. It makes use of introduction and elimination rules, and more critically of variables. For a textbook account of a proof system for first-order logic presented in this way, see van Dalen [36].

General sentences. in this fragment are what usually are called formulas. We prefer to change the standard terminology to make the point that here, sentences are not built from formulas by quantification. In fact, sentences in our sense do not have variable occurrences. But general sentences do include variables. They are only used in our proof theory.

The syntax of general sentences is given in Figure 8.2. What we are calling individual terms are just variables and constant symbols. (There are no function symbols here.) Using terms allows us to shorten the statements of our rules, but this is the only reason to have terms.

An additional note: we don’t need general sentences of the form \(r(j, x)\) or \(r(x, j)\). In larger fragments, we would expect to see general sentences of these forms, but our proof theory will not need these.
### Figure 8.2. Syntax of general sentences of $\mathcal{L}$, with $\varphi$ ranging over sentences.

<table>
<thead>
<tr>
<th>Expression</th>
<th>Variables</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>individual variable</td>
<td>$x, y$</td>
<td>$x</td>
</tr>
<tr>
<td>individual term</td>
<td>$t, u$</td>
<td></td>
</tr>
<tr>
<td>general sentence</td>
<td>$\alpha$</td>
<td>$\varphi</td>
</tr>
</tbody>
</table>

The bar notation, again. We have already seen the bar notation $\overline{c}$ for set terms $c$, and $\overline{\varphi}$ for sentences $\varphi$. We extend this to formulas $\overline{b(x)} = b(x)$, $\overline{r(x,y)} = r(x,y)$. We technically have a general sentence $\bot$, but this plays no role in the proof theory.

We write $\Gamma \vdash \varphi$ if there is a proof tree conforming to the rules of the system with root labeled $\varphi$ and whose axioms are labeled by elements of $\Gamma$. (Frequently we shall be sloppy about the labeling and just speak, e.g., of the root as if it were a sentence instead of being labeled by one.) Instead of giving a precise definition here, we shall content ourselves with a series of examples in Section 8.1.2 just below.

The system has two rules called ($\forall E$), one for deriving general sentences of the form $c(x)$ or $c(j)$, and one for deriving general sentences $r(x,y)$ or $r(j,k)$. (Other rules are doubled as well, of course.) It surely looks like these should be unified, and the system would of course be more elegant if they were. But given the way we are presenting the syntax, there is no way to do this. That is, we do not have a concept of substitution, and so rules like ($\forall E$) cannot be formulated in the usual way. Returning to the two rules with the same name, we could have chosen to use different names, say ($\forall E_1$) and ($\forall E_2$). But the result would have been a more cluttered notation, and it is always clear from context which rule is being used.

Although we are speaking of trees, we don’t distinguish left from right. This is especially the case with the ($\exists E$) rules, where the canceled hypotheses may occur in either order.

Side Conditions. As with every natural deduction system using variables, there are some side conditions which are needed in order to have a sound system.

In ($\forall I$), $x$ must not occur free in any uncanceled hypothesis. For example, in the version whose root is $\forall(c, d)$, one must cancel all occurrences of $c(x)$ in the leaves, and $x$ must not appear free in any other leaf.

In ($\exists E$), the variable $x$ must not occur free in the conclusion $\alpha$ or in any uncanceled hypothesis in the subderivation of $\alpha$.

In contrast to usual first-order natural deduction systems, there are no side conditions on the rules ($\forall E$) and ($\exists I$). The usual side conditions are phrased in terms of concepts such as free substitution, and the syntax here has no substitution to begin with. To be sure on this point, one should check the soundness result of Lemma 8.6.

Formal proofs in the Fitch style. The proof system in this chapter is presented in a standard Gentzen-style format. But it may easily be re-formatted to look more like a Fitch system, as we shall see in Example 8.3 and Figure 8.5. These examples might give the impression that we have merely re-presented Fitch-style natural deduction proofs. The difference is that our syntax is not a special case of the syntax of first-order logic. Corresponding to this, our proof rules our rather
restrictive, and the system cannot be used for much of anything beyond the language \( \mathcal{L} \). However, the fact that our Fitch-style proofs look like familiar formal proofs is a virtue: for example, it means that one could teach logic using this material.

### 8.1.2. Examples

We present a few examples of the proof system at work, along with comments pertaining to the side conditions. Many of these are taken from the proof system \( \mathbb{R}^* \) for the language \( \mathcal{R} \) of [29]. That system \( \mathbb{R}^* \) is among the strongest of the known syllogistic systems, and so it is of interest to check the current proof system is at least as strong.

**Example 8.2** Here is a proof of the classical syllogism *Darii*: \( \forall(b, d), \exists(c, b) \vdash \exists(c, d) \). First, in Fitch-style:

<table>
<thead>
<tr>
<th>Step</th>
<th>Formula</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( \forall(b, c) )</td>
<td>hyp</td>
</tr>
<tr>
<td>2</td>
<td>( \exists(b, d) )</td>
<td>hyp</td>
</tr>
<tr>
<td>3</td>
<td>( b(x) )</td>
<td>( \exists \mathcal{E}, 2 )</td>
</tr>
<tr>
<td>4</td>
<td>( d(x) )</td>
<td>( \exists \mathcal{E}, 2 )</td>
</tr>
<tr>
<td>5</td>
<td>( c(x) )</td>
<td>( \forall \mathcal{E}, 1, 3 )</td>
</tr>
<tr>
<td>6</td>
<td>( \exists(c, d) )</td>
<td>( \exists \mathcal{I}, 4, 5 )</td>
</tr>
</tbody>
</table>

---

**Figure 8.3.** Proof rules. See the text for the side conditions in the (\( \forall \mathcal{I} \)) and (\( \exists \mathcal{E} \)) rules.
Example 8.3 Next we study a principle called (K) in [29]. Intuitively, if all watches are expensive items, then everyone who owns a watch owns an expensive item. The formal statement in our language is $\forall(c, d) \vdash \forall(\exists(c, r), \exists(d, r))$. See Figure 8.4. We present a Fitch-style proof on the left and the corresponding one in our formalism on the right. One aspect of the Fitch-style system is that $(\exists E)$ gives two lines; see lines 3 and 4 on the left in Figure 8.4.

Example 8.4 Here is an example of a derivation using (RAA). It shows $\forall(c, \tau) \vdash \forall(d, \forall(c, r))$.

Example 8.5 Here is a statement of the rule of proof by cases: If $\Gamma + \varphi \vdash \psi$ and $\Gamma + \varphi \vdash \psi$, then $\Gamma \vdash \psi$. (Here and below, $\Gamma + \varphi$ denotes $\Gamma \cup \{\varphi\}$.) Instead of giving a derivation, we only indicate the ideas. Since $\Gamma + \varphi \vdash \psi$, we have $\Gamma + \varphi + \neg \psi \vdash \bot$ using ($\bot I$). From this and (RAA), $\Gamma, \neg \varphi \vdash \neg \psi$. Take a derivation showing $\Gamma + \neg \varphi \vdash \psi$, and replace the labeled $\neg \varphi$ with derivations from $\Gamma + \neg \varphi$. We thus see that $\Gamma + \neg \varphi \vdash \psi$. 
Using ($\bot I$), $\Gamma + \psi \vdash \bot$. And then using (RAA) again, $\Gamma \vdash \psi$. (This point is from [29].)

### 8.1.3. Soundness

Before presenting a soundness result, it might be good to see an improper derivation. Here is one, purporting to infer some men see some men from some men see some women:

\[
\frac{[s(x, x)]^1 \quad [m(x)]^2}{\exists(s, m)(x) \quad \exists I} \exists I
\]

\[
\frac{\exists(m, \exists(w, s)) \quad \exists(m, \exists(m, s)) \quad \exists E^1}{\exists(m, \exists(m, s)) \quad \exists E^2}
\]

The specific problem here is that when $[s(x, x)]$ is withdrawn in the application of $\exists I^1$, the variable $x$ is free in the as-yet-uncanceled leaves labeled $m(x)$.

To state a result pertaining to the soundness of our system, we need to define the truth value of a general sentence under a variable assignment. First, a variable assignment in a model $M$ is a function $v : V \rightarrow M$, where $V$ is the set of variable symbols and $M$ is the universe of $M$. We need to define $M \models \alpha[v]$ for general sentences $\alpha$. If $\alpha$ is a sentence, then $M \models \alpha[v]$ iff $M \models \alpha$ in our earlier sense. If $\alpha$ is $b(x)$, then $M \models \alpha[v]$ iff $[b](v(x))$. If $\alpha$ is $r(x, y)$, then $M \models \alpha[v]$ iff $[r](v(x), v(y))$. If $\alpha$ is $\bot$, then $M \models \bot$ for all models $M$.

**Lemma 8.6** Let $\Pi$ be any proof tree for this fragment all of whose nodes are labeled with $\mathcal{L}$-formulas, let $\varphi$ be the root of $\Pi$, let $M$ be a structure, let $v : X \rightarrow M$ be a variable assignment, and assume that for all uncanceled leaves $\psi$ of $\Pi$, $M \models \psi[v]$. Then also $M \models \varphi[v]$.

**Proof** By induction on $\Pi$. We shall only go into details concerning two cases. First, consider the case when the root of $\Pi$ is

\[
\frac{[c(x)] \quad [r(t, x)]}{\exists(c, r)(t) \quad \alpha \quad \exists E}
\]

To simplify matters further, let us assume that $t$ is a variable. Let $v$ be a variable assignment making true all of the leaves of the tree, except possibly $c(x)$ and $r(t, x)$. By induction hypothesis, $M \models \exists(c, r)(t)[v]$. Let $a \in A$ witness this assertion. In the obvious notation, $[c](a)$ and $[r](t^M, v, a)$. Let $w$ be the same variable assignment as $v$, except that $w(x) = a$. Then since $x$ is not free in any leaves except those labeled $c(x)$ and $r(t, x)$, we have $M \models \psi[w]$ for all those $\psi$. And so $M \models \alpha[w]$, using the induction hypothesis applied to the subtree on the right. And since $x$ is not free in the conclusion $\alpha$, we also have $M \models \alpha[v]$, as desired.

Second, let us consider the case when the root is

\[
\frac{c(y) \quad \forall(c, r)(x)}{r(x, y) \quad \forall E}
\]

(That is, we are considering an instance of $(\forall E)$ when the terms $t$ and $u$ are variables.) The variables $x$ and $y$ might well be the same. Let $M$ be a structure, and $v$ be a variable assignment making true the leaves of the tree. By induction hypothesis, $[c](v(y))$ and also $[r](v(x), m)$ for all $m \in [c]$. In particular, $[r](v(x), v(y))$.

The remaining cases are similar. □ □
8.1.4. The Henkin Property. The completeness of the logic parallels the Henkin-style completeness result for first-order logic. Given a consistent theory $\Gamma$, we get a model of $\Gamma$ in the following way: (1) take the underlying language $L$, add constant symbols to the language to witness existential sentences; (2) extend $\Gamma$ to a maximal consistent set in the larger language; and then (3) use the set of constant symbols as the carrier of a model in a canonical way. In the setting of this chapter, the work is in some ways easier than in the standard setting, and in some other ways the work here is easier.

Given two languages $L$ and $L'$, we say that $L' \supseteq L$ if every symbol (of any type) in $L$ is also a symbol (of the same type) in $L'$. In this chapter, the main case is when $P(L) = P(L')$, $R(L) = R(L')$, and $K(L) \subseteq K(L')$; that is, $L'$ arises by adding constants to $L$.

A theory in a language is just a set of sentences in it. Given a theory $\Gamma$ in a language $L$, and a theory $\Gamma^*$ in an extension $L' \supseteq L$, we say that $\Gamma^*$ is a conservative extension of $\Gamma$ if for every $\varphi \in L$, if $\Gamma^* \vdash \varphi$, then $\Gamma \vdash \varphi$.

**Lemma 8.7** Let $\Gamma$ be a consistent $L$-theory, and let $j \notin K(L)$.

1. If $\exists(c,d) \in \Gamma$, then $\Gamma + c(j) + d(j)$ is a conservative extension of $\Gamma$.
2. If $\exists(c,r)(j) \in \Gamma$, then $\Gamma + r(j,k) + c(k)$ is a conservative extension of $\Gamma$.

**Proof** For (1), suppose that $\Gamma$ contains $\exists(c,d)$ and that $\Gamma + c(j) + d(j) \vdash \varphi$. Let $\Pi$ be a derivation tree. Replace the constant $j$ by an individual variable $x$ which does not occur in $\Pi$. The result is still a derivation tree, except that the leaves are not labeled by sentences. (The reason is that our proof system has no rules specifically for constants, only for terms which might be constants and also might be individual variables.) Call the resulting tree $\Pi'$. Now the following proof tree shows that $\Gamma \vdash \varphi$:

\[
\begin{array}{c}
\exists(c,d) \\
\vdots \\
\varphi \\
\exists E
\end{array}
\]

The subtree on the right is $\Pi'$. The point is that the occurrences of $c(x)$ and $d(x)$ have been canceled by the use of $\exists E$ at the root.

This completes the proof of the first assertion, and the proof of the second is similar. \qed

**Definition 8.8** An $L$-theory $\Gamma$ has the Henkin property if the following hold:

1. If $\exists(c,d) \in \Gamma$, then for some constant $j$, $c(j)$ and $d(j)$ belong to $\Gamma$.
2. If $r$ is a literal of $L$ and $\exists(c,r)(j) \in \Gamma$, then for some constant $k$, $r(j,k)$ and $c(k)$ belong to $\Gamma$.

**Lemma 8.9** Let $\Gamma$ be a consistent $L$-theory. Then there is some $L^* \supseteq L$ and some $L^*$-theory $\Gamma^*$ such that $\Gamma^*$ is a maximal consistent theory with the Henkin property. Moreover, if $s \in R(L)$, $j \in K(L^*)$ and $k \in K(L)$, and if $s(j,k) \in \Gamma^*$, then $j \in K(L)$.

**Proof** This is a routine argument, using Lemma 8.7. One dovetails the addition of constants which is needed for the Henkin property together with the addition of sentences needed to insure maximal consistency. The formal details would use
Lemma 8.7 for steps of the first kind, and for the second kind we need to know that if \( \Gamma \) is consistent, then for all \( \varphi \), either \( \Gamma + \varphi \) or \( \Gamma + \neg \varphi \) is consistent. This follows from the derivable rule of proof by cases; see Example 8.5 in Section 8.1.2. \( \square \) \( \square \)

The last point in Lemma 8.9 states a technical property that will be useful in Section 8.2.1.

It might be worthwhile noting that the extensions produced by Lemma 8.9 add infinitely many constants to the language.

**8.1.5. Completeness Via Canonical Models.** In this section, fix a language \( \mathcal{L} \) and a maximal consistent Henkin \( \mathcal{L} \)-theory \( \Gamma \). We construct a canonical model \( \mathcal{M} = \mathcal{M}(\Gamma) \) as follows: \( \mathcal{M} = \mathcal{K}(\mathcal{L}) \): \( [p](j) \) iff \( p(j) \in \Gamma \); \( [s](j, k) \) iff \( s(j, k) \in \Gamma \);

and \( [j] = j \). That is, we take the constant symbols of the language to be the points of the model, and the interpretations of the atoms are the natural ones. Each constant symbol is interpreted by itself.

**Lemma 8.10** For all set terms \( c \), \( [c] = \{ j : c(j) \in \Gamma \} \).

**Proof** By induction on \( c \). The base case of unary atoms \( p \) is by definition of \( \mathcal{M} \).

Before we turn to the induction proper, here is a preliminary point. Assuming that \( [c] = \{ j : c(j) \in \Gamma \} \), we check that \( [\exists c] = \{ j : \exists c(j) \in \Gamma \} \):

\[
j \in [\exists c] \iff j \notin [c] \iff c(j) \notin \Gamma \iff c(j) \in \Gamma.
\]

The last point uses the maximal consistency of \( \Gamma \).

Turning to the inductive steps, assume our result for \( c \); we establish it for \( \forall (c, s) \) and \( \exists (c, s) \); it then follows from the preliminary point that we have the same fact for \( \forall (c, s) \) and \( \exists (c, s) \).

Let \( j \in [\forall (c, s)] \). We claim that \( \forall (c, s)(j) \in \Gamma \). For if not, then \( \exists (c, s) \) \( (j) \in \Gamma \).

By the Henkin property, let \( k \) be such that \( \Gamma \) contains \( c(k) \) and \( s(j, k) \). By the induction hypothesis, \( k \in [c] \), and by the definition of \( \mathcal{M} \), \( [s](j, k) \) is false. Thus \( j \notin [\forall (c, s)] \). This is a contradiction.

In the other direction, assume that \( \forall (c, s)(j) \in \Gamma \); this time we claim that \( j \in [\forall (c, s)] \).

By induction hypothesis, \( \Gamma \) contains \( c(j) \). By \( \forall E \), we see that \( \Gamma \vdash s(j, k) \). Hence \( \Gamma \) contains \( s(j, k) \). So in \( \mathcal{M} \), \( [s](j, k) \). Since \( k \) was arbitrary, we see that indeed \( j \in [\forall (c, s)] \).

The other induction step is for \( \exists (c, s) \). Let \( j \in [\exists (c, s)] \). We thus have some \( k \in [c] \) such that \( [s](j, k) \) is true. That is, \( s(j, k) \in \Gamma \).

Using \( \exists I \), we have \( \Gamma \vdash \exists (c, s)(j) \); from this we see that \( \exists (c, s)(j) \in \Gamma \), as desired.

Finally, assume that \( \exists (c, s)(j) \in \Gamma \). By the Henkin condition, let \( k \) be such that \( \Gamma \) contains \( c(k) \) and \( s(j, k) \). Using the derivation above, we have the desired conclusion that \( j \in [\exists (c, s)] \).

This concludes the proof. \( \square \) \( \square \)

**Lemma 8.11** \( \mathcal{M} \models \Gamma \).

**Proof** We check the sentence types in turn. Throughout the proof, we shall use Lemma 8.10 without mention.

First, let \( \Gamma \) contain the sentence \( \forall (c, d) \). Let \( j \in [c] \), so that \( c(j) \in \Gamma \). We have \( d(j) \in \Gamma \) using \( \forall E \). This for all \( j \) shows that \( \mathcal{M} \models \forall (c, d) \).

Second, let \( \exists (c, d) \in \Gamma \). By the Henkin condition, let \( j \) be such that both \( c(j) \) and \( d(j) \) belong to \( \Gamma \). This element \( j \) shows that \( [c] \cap [d] \neq \emptyset \). That is, \( \mathcal{M} \models \exists (c, d) \).
Continuing, consider a sentence \( c(j) \in \Gamma \). Then \( j \in \mathbb{c} \), so that \( \mathcal{M} \models c(j) \).

Finally, the case of sentences \( r(j,k) \in \Gamma \) is immediate from the structure of the model.

\[
\text{Theorem 8.12} \quad \text{If } \Gamma \models \varphi, \text{ then } \Gamma \vdash \varphi.
\]

**Proof** We rehearse the standard argument. Due to the classical negation, we need only show that consistent sets \( \Gamma \) are satisfiable. Let \( \mathcal{L} \) be the language of \( \Gamma \).

Let \( \mathcal{L}' \supseteq \mathcal{L} \) be an extension of \( \mathcal{L} \), and let \( \Gamma^* \supseteq \Gamma \) be a maximal consistent theory in \( \mathcal{L}' \) with the Henkin property (see Lemma 8.9). Consider the canonical model \( \mathcal{M}(\Gamma^*) \) as defined in this section. By Lemma 8.11, \( \mathcal{M}(\Gamma^*) \models \Gamma^* \). Thus \( \Gamma^* \) is satisfiable, and hence so is \( \Gamma \).

\[
8.1.6. \text{The Finite Model Property.} \quad \text{Let } \Gamma \text{ be a consistent finite theory in some language } \mathcal{L}. \quad \text{As we now know, } \Gamma \text{ has a model. Specifically, we have seen that there is some } \Gamma^* \supseteq \Gamma \text{ which is a maximal consistent theory with the Henkin property in an extended language } \mathcal{L}^* \supseteq \mathcal{L}. \quad \text{Then we may take the set of constant symbols of } \mathcal{L}^* \text{ to be the carrier of a model of } \Gamma^*, \text{ hence of } \Gamma. \quad \text{The model obtained in this way is finite. It is of interest to build a finite model, so in this section } \Gamma \text{ must be finite. The easiest way to see that } \Gamma \text{ has a finite model is to recall that our overall language is a sub-language of the two variable fragment } \text{FO}^2 \text{ of first-order logic. And } \text{FO}^2 \text{ has the finite model property by Mortimer's Theorem [17].}
\]

However, it is possible to give a direct argument for the finite model property, along the lines of filtration in modal logic (but with some differences). We sketch the result here because we shall use the same method in Section 8.2.1 below to prove a finite model property for our second logical system \( \mathcal{L}(adj) \) with respect to its natural semantics; that result does not follow from others in the literature.

Let \( \mathcal{M} = \mathcal{M}(\Gamma^*) \) be the canonical model as defined in Section 8.1.5. Let \( \text{Sub}(\Gamma) \) be the collection of set terms occurring in any sentence in the original finite theory \( \Gamma \). So \( \text{Sub}(\Gamma) \) is finite, and if \( \forall(c,r) \in \text{Sub}(\Gamma) \) or \( \exists(c,r) \in \text{Sub}(\Gamma) \), then also \( c \in \text{Sub}(\Gamma) \).

For constant symbols \( j \) and \( k \) of \( \mathcal{L}^* \), write \( j \equiv k \) iff the following conditions hold:

1. If either \( j \) or \( k \) is a constant of \( \mathcal{L} \), then \( j = k \).
2. For all \( c \in \text{Sub}(\Gamma), c(j) \in \Gamma \) iff \( c(k) \in \Gamma \).

**Remark** The equivalence relation \( \equiv \) may be defined on any structure. It is not necessarily a congruence, as Example 5.2 shows. Specifically, we had constant symbols \( j \) and \( k \) such that \( j \equiv k \), and yet in our structure \( s(j,k) \) and \( s(k,j) \). In the case of \( \mathcal{M}(\Gamma^*) \), we have no reason to think that \( \equiv \) is a congruence. That is, the construction in Section 8.1.5 did not arrange for this.

Let \( N = \{[k] : k \in \mathcal{K}(\mathcal{L}) \} \times \{\forall, \exists\} \). (We use \( \forall \) and \( \exists \) as tags to give two copies of the quotient \( \mathcal{K}/\equiv \).) We endow \( N \) with an \( \mathcal{L} \)-structure as follows:

\[
\begin{align*}
\text{(8.1)} & \quad [p] = \{(j, Q) : p(j) \in \Gamma^* \} \text{ and } Q \in \{\forall, \exists\}. \\
\text{(8.2)} & \quad [s]((j, Q), ([k], Q')) \text{ iff one of the following two conditions holds:} \\
& \text{(1) There is a set term } c \text{ such that } \Gamma^* \text{ contains } c(k) \text{ and } \forall(c, s)(j). \\
& \text{(2) } Q' = \exists, \text{ and for some } j_\ast \equiv j \text{ and } k_\ast \equiv k, \Gamma^* \text{ contains } s(j_\ast, k_\ast). \\
\text{(8.3)} & \quad \text{For a constant } j \text{ of } \mathcal{L}, [j] = ([j], \exists). \quad \text{(Of course, } [j] \text{ is the singleton set } \{j\}).
\end{align*}
\]
Before going on, we note that the first of the two alternatives in the definition of $[s][((j), Q), ((k), Q')]$ is independent of the choice of representatives of equivalence classes. And clearly so is the second alternative.

We shall write $N$ for the resulting $L$-structure, hiding the dependence on $\Gamma$ and $\Gamma^*$.

**Lemma 8.13** For all $c \in \text{Sub}(\Gamma)$, 
\[ [c] = \{([j], Q) : c(j) \in \Gamma^* \text{ and } Q \in \{\forall, \exists\} \} \]

**Proof** By induction on set terms $c$. We are not going to present any of the details here because in Lemma 8.21 below, we shall see all the details on a more involved result.

**Lemma 8.14** $N \models \Gamma$.

**Proof** Again we only highlight a few details, since the full account is similar to what we saw in Lemma 8.11, and to what we shall see in Lemma 8.22. One would check the sentence types in turn, using Lemma 8.13 frequently. We want to what we saw in Lemma 8.11, and to what we shall see in Lemma 8.22. One involved result.

Before going on, we note that the first of the two alternatives in the definition of $\Gamma$ of the form $\Gamma$ would check the sentence types in turn, using Lemma 8.13 frequently. We wish to show that $\Gamma$ contains $\forall(c, s)(j)$. By the definition of $[\forall](j, k, \exists)$, we have $\forall(c, s)(j)$. By the way binary atoms and constants are interpreted in $N$, we have $N \models s(j, k)$, as desired.

We conclude with the consideration of a sentence in $\Gamma$ of the form $\forall(c, s)(j)$. We wish to show that $N \models \forall(c, s)(j)$. Suppose towards a contradiction that $N \models s(j, k)$. Then we have $[s](([j], \exists), ([k], \exists))$. There are two possibilities, corresponding to the alternatives in the semantics of $\forall$. The first is when there is a set term $c$ such that $\Gamma^*$ contains $c(k)$ and $\forall(c, s)(j)$. By $(\forall E)$, $\Gamma^*$ then contains $s(j, k)$. But recall that $\Gamma$ contains $\forall(c, s)(j)$. So in this alternative, $\Gamma^* \supset \Gamma$ is inconsistent. In the second alternative, there are $j_*, k_*$ such that $s(j_*, k_*) \in \Gamma^*$. But recall that the equivalence classes of constant symbols from the base language $L$ are singletons. Thus in this alternative, $j_*, k_*$ are constant symbols of that language. Also recall that $[\forall] = ([j], \exists)$, and similarly for $k$.

First, consider sentences in $\Gamma$ of the form $s(j, k)$. By the definition of $[s][((j), \exists), ((k), \exists)]$. By the way binary atoms and constants are interpreted in $N$, we have $N \models s(j, k)$, as desired.

We conclude with the consideration of a sentence in $\Gamma$ of the form $\forall(c, s)(j)$. We wish to show that $N \models \forall(c, s)(j)$. Suppose towards a contradiction that $N \models s(j, k)$. Then we have $[s](([j], \exists), ([k], \exists))$. There are two possibilities, corresponding to the alternatives in the semantics of $\forall$. The first is when there is a set term $c$ such that $\Gamma^*$ contains $c(k)$ and $\forall(c, s)(j)$. By $(\forall E)$, $\Gamma^*$ then contains $s(j, k)$. But recall that $\Gamma$ contains $\forall(c, s)(j)$. So in this alternative, $\Gamma^* \supset \Gamma$ is inconsistent. In the second alternative, there are $j_*, k_*$ such that $s(j_*, k_*) \in \Gamma^*$. But recall that the equivalence classes of constant symbols from the base language $L$ are singletons. Thus in this alternative, $j_*, k_*$ are constant symbols of that language. Also recall that $[\forall] = ([j], \exists)$, and similarly for $k$.

**Theorem 8.15 (Finite Model Property)** If $\Gamma$ is consistent, then $\Gamma$ has a model of size at most $2^{2^n}$, where $n$ is the number of set terms in $\Gamma$.

Complexity notes. Theorem 8.15 implies that the satisfiability problem for our language is in NExpTime. We can improve this to an ExpTime-completeness result by quoting the work of others. Pratt-Hartmann [25] define a certain logic $\mathcal{E}_2$ and showed that the complexity of its satisfiability problem is ExpTime-complete. $\mathcal{E}_2$ corresponds to a fragment of first-order logic, and it is somewhat bigger than the language $L$. (It would correspond to adding converses to the binary atoms in $L$, as we mention at the very end of this chapter.) Since satisfiability for $\mathcal{E}_2$ is ExpTime-complete, the same problem for $L$ is in ExpTime.

A different way to obtain this upper bound is via the embedding into Boolean modal logic which we saw in Section 5.0.2. For this, see Theorem 7 of Lutz and Sattler [13]. We shall use an extension of that result below in connection with an extension $L(adj)$ of $L$. 
The \textsc{ExpTime}-hardness for $\mathcal{L}$ follows from Lemma 6.1 in [29]. That result dealt with a language called $\mathcal{R}^\dagger$, and $\mathcal{R}^\dagger$ is a sub-language of $\mathcal{L}$.

### 8.2. Adding Transitivity: $\mathcal{L}(\mathit{adj})$

Before going further, let us briefly recapitulate the overall problem of this chapter and point out where we are and what remains to be done. We aim to formalize a fragment of first-order logic in which one may represent arguments as complex as that in (8.4) in the Introduction. We are especially interested in decidable systems, and so the systems must be weaker than first-order logic. We presented in Section 8 a language $\mathcal{L}$ and a proof system for it. Validity in the logic cannot be captured by a purely syllogistic proof system, and so our proof system uses variables. But the use is very special and restricted. The proof system is complete and decidable in exponential time. To our knowledge, it is the first system with these properties.

There are a number of ways in which one can go further. In this section, we want to explore one such way, connected to the example in (8.4) below:

\begin{align*}
\text{Every sweet fruit is bigger than every ripe fruit} \\
\text{Every pineapple is bigger than every kumquat} \\
\text{Every non-pineapple is bigger than every unripe fruit} \\
\text{Every fruit bigger than some sweet fruit is bigger than every kumquat}
\end{align*}

One key feature of this example is that comparative adjectives such as \textit{bigger than} are transitive. This is true for all comparative adjectives.

We extend our language $\mathcal{L}$ to a language $\mathcal{L}(\mathit{adj})$ by taking a basic set $\mathbf{A}$ of comparative adjective phrases in the base. The proof system simply extends the one we have already seen with a rule corresponding to the transitivity of comparatives. Our completeness result, Theorem 8.12, extends to the new setting. The next section does this. The decidability of the language is a more delicate matter than before, since it does not follow from Mortimer’s Theorem [17] on the finite model property for \textsc{FO}$^2$. Indeed, adding transitivity statements to \textsc{FO}$^2$ renders the logic undecidable, as shown in Grädel, Otto, and Rosen [8]. Instead, one could use Theorem 12 of Lutz and Sattler [13] on the decidability of a variant on Boolean modal logic in which some of the relations are taken to be transitive. This would indeed give the \textsc{ExpTime}-completeness of $\mathcal{L}(\mathit{adj})$ with our semantics. However, we have decided to present a direct proof for several reasons. First, Lutz and Sattler’s result does not give a finite model property, and our result does do this. Second, our argument is shorter. Finally, our treatment connects to modal filtration arguments and is therefore different; [13] uses automata on infinite trees and is based on Vardi and Wolper [37].

I do not wish to treat the transitivity of comparison with adjectives as an enthymeme (missing premise) because the transitivity seems more fundamental, more ‘logical’ somehow. Hence it should be treated on a deeper level. The decidability considerations give a supporting argument: if we took the transitivity to be a meaning postulate, then it would seem that the underlying language would have to be rich enough to state transitivity. This requires three universal quantifiers. For other reasons, we want our languages to be closed under negation. It thus seems very likely that any logical system with these properties is going to be undecidable. The
upshot is a system in which the transitivity turns out to be a proof postulate rather than a meaning postulate. We turn to the system itself.

Syntax and semantics. We start with four pairwise disjoint sets $A$ (for comparative adjective phrases) and the three that we saw before: $P$, $R$, and $K$. We use $a$ as a variable to range over $A$ in our statement of the syntax and the rules.

For the syntax, we take elements $a \in A$ to be binary atoms, just as the elements $s \in R$ are. Thus, the binary literals are the expressions of the form $s, s, a$, or $a$.

The syntax is the same as before, except that we allow the binary atoms to be elements of $A$ in addition to elements of $R$. So in a sense, we have the same syntax is before, except that some of the binary atoms are taken to render transitive verbs, and some are taken to render comparative adjective phrases. The only difference is in the semantics. Here, we require that (in every model $M$) for an adjective $a \in A$, $[a]$ must be a transitive relation.

Proof system. We adopt the same proof system as in Figure 8.3, but with one addition. This addition is the rule for transitivity:

$$\frac{a(t_1, t_2) \quad a(t_2, t_3)}{a(t_1, t_3)} \text{trans}$$

This rule is added for all $a \in A$.

Example 8.16 We have seen an informal example in (8.4) at the beginning of this chapter. At this point, we can check that our system does indeed have a derivation corresponding to this. We need to check that $\Gamma \vdash \varphi$, where $\Gamma$ contains

$$\forall(sw, \forall(ripe, bigger)), \forall(pineapple, \forall(kq, bigger)), \forall(pineapple, \forall(ripe, bigger)),$$

and $\varphi$ is

$$\forall(\exists(sw, bigger), \forall(kq, bigger)).$$

(We are going to use $kq$ as an abbreviation of kumquat for typographical convenience, and similarly for $sw$ and sweet.)

Example 8.17 The example at the beginning of this chapter cannot be formalized in this fragment because the correct reasoning uses the transitivity of is bigger than. However, we can prove a result which may itself be used in a formal proof of (8.4):

$$\begin{align*}
\text{Every sweet fruit is bigger than every ripe fruit} \\
\text{Every pineapple is bigger than every kumquat} \\
\text{Every non-pineapple is bigger than every unripe fruit} \\
\text{Every sweet fruit is bigger than every kumquat}
\end{align*}$$

(8.5)

To discuss this, we take the set $P$ of unary atoms to be

$$P = \{\text{sweet, ripe, pineapple, kumquat}\}.$$  

We also take $R = \{\text{bigger}\}$ and $K = \emptyset$. Figure 8.5 contains a derivation showing (8.5), done in the manner of Fitch [5]. The main way in which we have bent the English in the direction of our formalism is to use the bar notation on the nouns. The main reason for presenting the derivation as a Fitch diagram is that the derivation given as a tree (as demanded by our definitions) would not fit on a page. This is because the cases rule is not a first-class rule in the system, it is a derived rule (see Example 8.5 above). Our Fitch diagram pretends that the system has a rule of cases. Another reason to present the derivation as in Figure 8.5 is to
Every sweet fruit is bigger than every ripe fruit

Every pineapple is bigger than every kumquat

Every pineapple is bigger than every ripe fruit

x is a sweet fruit

x is bigger than every ripe fruit

x is a pineapple

x is bigger than every kumquat

x is a pineapple

x is bigger than every ripe fruit

y is a kumquat

y is a ripe fruit

x is bigger than y

y is a ripe fruit

x is bigger than y

x is bigger than y

x is bigger than every kumquat

x is bigger than every kumquat

Every sweet fruit is bigger than every kumquat

Figure 8.5. A derivation corresponding to the argument in (8.5).

8. LOGIC BEYOND THE ARISTOTLE BORDER

make the point that the treatment in this chapter is a beginning of a formalization of the work that Fitch was doing.

In Figure 8.5, we see that $\Gamma \vdash \forall(sweet, \forall(kq, bigger))$. (The a derivation was presented using a format which could be converted to our official format of natural deduction trees.) That work used $R = \{bigger\}$, but here we want $R = \emptyset$ and $A = \{bigger\}$. The same derivation works, of course. Transitivity enables us to obtain a derivation for (8.4):

\[
\begin{align*}
\exists(sw, bigger)(x) & \vdash \forall(kq, bigger)(x) \\
\forall(kq, bigger)(x) & \vdash \forall(\exists(sw, bigger), \forall(kq, bigger)) \\
\forall(\exists(sw, bigger), \forall(kq, bigger)) & \vdash \forall I^3
\end{align*}
\]

Adding the transitivity rule gives a sound and complete proof system for the semantic consequence relation $\Gamma \models \varphi$. The soundness is easy, and so we only sketch
the completeness. We must show that a set \( \Gamma \) which is consistent in the new logic has a transitive model. The canonical model \( M(\Gamma) \) as defined in Section 8.1.5 is automatically transitive; this is immediate from the transitivity rule. And as we know, it satisfies \( \Gamma \).

### 8.2.1. \( \mathcal{L}(adj) \) has the Finite Model Property.

Our final result is that \( \mathcal{L}(adj) \) has the finite model property. We extend the work in Section 8.1.6. The inspiration for our definitions comes from the technique of filtration in modal logic, but we shall not refer explicitly to this area.

We again assume that \( \Gamma \) is consistent, and \( \Gamma^* \) has the properties of Lemma 8.9.

**Definition 8.18** For \( a \in A \), we say that \( j \) reaches \( k \) (by a chain of \( \equiv \) and \( a \) statements) if there is a sequence

\[
\begin{align*}
   & j = j_0 \equiv k_0, \\
   & j_1 \equiv k_1, \ldots, \\
   & j_n \equiv k_n = k
\end{align*}
\]

such that \( n \geq 1 \), and \( \Gamma^* \) contains \( a(k_0, j_1), \ldots, a(k_{n-1}, j_n) \).

**Lemma 8.19** Assume that \( j \) reaches \( k \) by a chain of \( \equiv \) and \( a \) statements.

1. If \( c(k) \in \Gamma^* \), then \( \Gamma^* \) contains \( \exists(c, a)(j) \).
2. If \( j, k \in K(\mathcal{L}) \), then \( \Gamma^* \) contains \( a(j, k) \).

**Proof** By induction on \( n \geq 1 \) in (8.6). For \( n = 1 \), we have essentially seen the argument as a step in Lemma 8.13. Here it is again. Since \( c(k_1) \) and \( j_1 \equiv k_1 \), we see that \( c(j_1) \). Together with \( a(k_0, j_1) \), we have \( \exists(c, a)(k_0) \). And as \( j_0 \equiv k_0 \), we see that \( \exists(c, a)(j_0) \).

Assume our result for \( n \), and now consider a chain as in (8.6) of length \( n + 1 \). The induction hypothesis applies to

\[
\begin{align*}
   & j = j_1 \equiv k_1, \\
   & j_2 \equiv k_2, \ldots, \\
   & j_{n+1} \equiv k_{n+1} = k
\end{align*}
\]

and so we have \( \exists(c, a)(j_1) \). Since \( a(k_0, j_1) \), we easily have \( \exists(c, a)(k_0) \) by transitivity. And as \( j_0 \equiv k_0 \), we have \( \exists(c, a)(j_0) \).

The second assertion is also proved by induction on \( n \geq 1 \). For \( n \), we have \( j = j_0 \equiv k_0 \), \( \Gamma^* \) contains \( a(k_0, j_1) \); and \( j_1 \equiv k_1 \). Then since the \( \equiv \) is the identity on \( K(\mathcal{L}) \), \( j = j_0 = k_0 \), and \( j_1 = k_1 = k \). Hence \( \Gamma^* \) contains \( s(j, k) \). Assuming our result for \( n \), we again consider a chain as in (8.6) of length \( n + 1 \). Just as before, \( j = j_0 = k_0 \), and so \( \Gamma^* \) contains \( a(j, j_1) \). By induction hypothesis, \( \Gamma^* \) contains \( a(j_1, k) \). By transitivity, \( \Gamma^* \) contains \( a(j, k) \).

We endow \( N \) with an \( \mathcal{L} \)-structure as follows:

\[
[p] = \{ ([j], Q) : p(j) \in \Gamma^* \text{ and } Q \in \{\forall, \exists\} \}. \\
[s](([j], Q), ([k], Q')) \text{ iff one of the following two conditions holds:} \\
(1) \text{ There is a set term } c \text{ such that } \Gamma^* \text{ contains } c(k) \text{ and } \forall(c, s)(j). \\
(2) Q' = \exists, \text{ and for some } j_* \equiv j \text{ and } k_* \equiv k, \Gamma^* \text{ contains } s(j_*, k_*). \\
[a][([j], Q), ([k], Q')] \text{ iff} \\
(1) \text{ If } \forall(c, a)(k) \in \Gamma^*, \text{ then also } \forall(c, a)(j) \in \Gamma^*. \\
(2) \text{ In addition, either (a) or (b) below holds:} \\
   (a) \text{ There is a set term } c \text{ such that } \Gamma^* \text{ contains } c(k) \text{ and } \forall(c, a)(j). \\
   (b) Q' = \exists, \text{ and } j \text{ reaches } k \text{ by a chain of } \equiv \text{ and } a \text{ statements.} \\
\text{(Notice that this definition is independent of the representatives in } [j] \text{ and} \\
[k].) \\
\text{For a constant } j \text{ of } \mathcal{L}, [j] = ([j], \exists).
Once again, we suppress $\Gamma$ and $\Gamma^*$ and simply write $N$ for the resulting $L$-structure.

**Lemma 8.20** For $a \in \mathbf{A}$, each relation $[a]$ is transitive in $N$.

**Proof** In this proof and the next, we are going to use $l$ to stand for a constant symbol, even though earlier in the chapter we used it for a literal. Assume that

$$([j], Q) \models [a] ([k], Q') \models [a] ([l], Q'').$$

Clearly we have the first requirement concerning $[a]$; if $\forall(c, a)(l) \in \Gamma^*$, then also $\forall(c, a)(j) \in \Gamma^*$.

We have four cases, depending on the reasons for the two assertions in (8.7).

- **Case 1** There is a set term $b$ such that $\Gamma^*$ contains $b(k)$ and $\forall(b, a)(j)$, and there is also a set term $c$ such that $\Gamma^*$ contains $c(l)$ and $\forall(c, a)(k)$. By (1), $\Gamma^*$ contains $c(l)$ and $\forall(c, a)(j)$. And so we have requirement (2a) concerning $[a]$ for $([j], Q)$ and $([l], Q'')$.

- **Case 2** There is a set term $b$ such that $\Gamma^*$ contains $b(k)$ and $\forall(b, a)(j)$, and $k$ reaches $l$. Note that $a(j, k)$. So $j$ reaches $l$.

- **Case 3** $j$ reaches $k$ by a chain of $\equiv$ and $a$ statements, and there is a set term $c$ such that $\Gamma^*$ contains $c(l)$ and $\forall(c, a)(k)$. Then $a(k, l)$. And so $j$ reaches $l$.

- **Case 4** $j$ reaches $k$, and $k$ reaches $l$. Then concatenating the chains shows that $j$ reaches $l$.

**Lemma 8.21** For all $c \in \text{Sub}(\Gamma)$, $[c] = \{([j], Q) : c(j) \in \Gamma^* \text{ and } Q \in \{\forall, \exists\}\}$.

**Proof** We argue by induction on $c$. Much of the proof is as in Lemma 8.13. For $c$ a unary atom, the result is obvious. Also, assuming that $[c] = \{([j], Q) : c(j) \in \Gamma^*\}$ we easily have the same result for $\overline{c}$ using the maximal consistency of $\Gamma^*$:

$$([j], Q) \in [\overline{c}] \iff ([j], Q) \not\in [c] \iff c(j) \not\in \Gamma^* \iff \overline{\exists}(j) \in \Gamma^*.$$

Assume about $c$ that if $c \in \text{Sub}(\Gamma)$, then $[c] = \{([j], Q) : c(j) \in \Gamma^*\}$. In view of what we just saw, we only need to check the same result for $\forall(c, s), \exists(c, s), \forall(c, a), \text{and } \exists(c, \overline{s})$.

- **$\forall(c, s)$**. Suppose that $\forall(c, s) \in \text{Sub}(\Gamma)$, so that $c \in \text{Sub}(\Gamma)$ as well. We prove that

$$[\forall(c, s)] = \{([j], Q) : \forall(c, s)(j) \in \Gamma^*\}.$$

Let $([j], Q) \in [\forall(c, s)]$. We shall show that $\forall(c, s)(j) \in \Gamma^*$. If not, then by maximal consistency, $\exists(c, \overline{s})(j) \in \Gamma^*$. By the Henkin property, let $k$ be such that $\Gamma^*$ contains $c(k)$ and $\overline{\exists}(j, k)$. By induction hypothesis, $([k], \forall) \in [c]$. And so $([j], \forall)[s][[k], \forall]$. Thus there is a set term $b$ such that $\Gamma^*$ contains $b(k)$ and $\forall(b, s)(j)$. From these, $\Gamma^*$ contains $s(j, k)$. And thus $\Gamma^*$ is inconsistent. This contradiction shows that indeed $\forall(c, s)(j) \in \Gamma^*$.

In the other direction, suppose that $([j], Q)$ is such that $\forall(c, s)(j) \in \Gamma^*$. Let $([k], Q') \in [c]$, so by induction hypothesis, $c(k) \in \Gamma^*$. By the way we interpret binary relations in $N$, $[s](([j], Q), ([k], Q'))$. This for all $([k], Q') \in [c]$ shows that $([j], Q) \in [\forall(c, s)]$.

- **$\exists(c, s)$**. Suppose that $\exists(c, s) \in \text{Sub}(\Gamma)$, so that $c \in \text{Sub}(\Gamma)$ as well. Let $([j], Q) \in [\exists(c, s)]$. Let $k$ and $Q'$ be such that $[c]([k], Q')$ and $[s](([j], Q), ([k], Q'))$. By induction hypothesis, $c(k) \in \Gamma^*$. First, let us consider the case when $Q' = \forall$. Let $b$ be such that $\Gamma^*$ contains $b(k)$ and $\forall(b, s)(j)$. Using $(\forall E)$, we have $\Gamma^* \vdash$
8.2. Adding transitivity: $\mathcal{L}(\text{adj})$

$\exists(c,s)(j)$. And as $\Gamma^*$ is closed under deduction, $\exists(c,s)(j) \in \Gamma^*$ as desired. The more interesting case is when $Q' = \exists$, so that for some $j_*, \equiv j$ and $k_*, \equiv k$, $\Gamma^*$ contains $s(j_*, k_*).$ Since $c(k)$ and $k \equiv k_*$, we have $c(k_*) \in \Gamma^*$. Then using ($\exists I$), we see that $\exists(c,s)(j_*) \in \Gamma^*$. Since $j \equiv j_*$, once again we have $\exists(c,s)(j) \in \Gamma^*$.

Conversely, suppose that $\exists(c,s)(j) \in \Gamma^*$. By the Henkin property, let $k$ be such that $c(k)$ and $s(j,k)$ belong to $\Gamma^*$. Then $[s]([[j],Q],([k],\exists))$, and by induction hypothesis, $[c]((k)$. Hence $([j],Q) \in [\exists(c,s)]$.

$\forall(c,a)$. Suppose that $\forall(c,a) \in \text{Sub}(\Gamma)$, so that $c \in \text{Sub}(\Gamma)$ as well. We prove that $\forall[c,a] \equiv \{([j],Q) : \forall(c,a)(j) \in \Gamma^*)\}$. The first part argument is the left-to-right inclusion. It is exactly the same as what we saw above for the sentences of the form $\forall(c,s)$.

In the other direction, suppose that $\forall(c,a)(j) \in \Gamma^*$; we show that $([j],Q) \in [\forall(c,a)]$. For this, let $([k],Q') \in [c]$. By induction hypothesis, $c(k) \in \Gamma^*$. We must verify that if $\forall(b,a)(k) \in \Gamma^*$, then also $\forall(b,a)(j) \in \Gamma^*$. This is shown in the derivation below:

$$\begin{array}{l}
\forall(c,a)(j) \\
\forall E \quad [b(x)]^1 \forall E \quad \forall(b,a)(k) \\
\forall E \quad a(k,x) \\
\text{trans} \quad a(j,x) \\
\forall E \quad \forall(b,a)(j) \quad \forall I^1
\end{array}$$

Since $\Gamma^*$ is closed under deduction, we see that indeed $\forall(b,a)(j) \in \Gamma^*$. Going on, we see from the structure of $N$ that $[s]([[j],Q],([k],Q'))$. This for all $(k,Q') \in [c]$ shows that $([j],Q) \in [\forall(c,a)]$.

$\exists(c,a)$. Suppose that $\exists(c,a) \in \text{Sub}(\Gamma)$, so that $c \in \text{Sub}(\Gamma)$ as well.

Let $([j],Q) \in [\exists(c,a)]$. Let $k$ and $Q'$ be such that the following two assertions hold: $[a]([[k],Q')]$ and $[a]([[j],Q],([k],Q'))$. By induction hypothesis, $c(k) \in \Gamma^*$. There are two cases depending on whether $Q' = \forall$ or $Q' = \exists$. The argument for $Q' = \forall$ is the same as the one we saw in our work on sentences $\exists(c,s)$ above. The more interesting case is when $Q' = \exists$. This time, $j$ reaches $k$. By Lemma 8.19, $\exists(c,a)(j) \in \Gamma^*$.

Conversely, suppose that $\exists(c,a)(j) \in \Gamma^*$. By the Henkin property, let $k$ be such that $c(k)$ and $a(j,k)$ belong to $\Gamma^*$. The derivation below shows that if $\forall(d,a)(k) \in \Gamma^*$, then $\forall(d,a)(j) \in \Gamma^*$ as well:

$$\begin{array}{l}
\forall(c,a)(k) \\
\forall E \quad [d(x)]^1 \forall E \quad \forall(d,a)(k) \\
\forall E \quad a(k,x) \\
\text{trans} \quad a(j,x) \\
\forall E \quad \forall(d,a)(j) \quad \forall I^1
\end{array}$$

So $[a]([[j],Q],([k],\exists))$, and by induction hypothesis, $[c]((k)$. Hence $([j],Q) \in [\exists(c,a)]$.

This completes the induction. □ □

**Lemma 8.22** $N \models \Gamma$.

**Proof** We check the sentence types in turn, using Lemma 8.21 without mention.
First, let $\Gamma$ contain the sentence $\forall(b, c)$. Then $b$ and $c$ belong to $Sub(\Gamma)$. Let $([[j], Q)] = \Gamma$, so that $b(j) \in \Gamma^*$. We have $d(j) \in \Gamma^*$ using ($\forall E$). This for all $([[j], Q])$ shows that $N \models \forall(b, c)$.

Second, let $\exists(c, d) \in \Gamma$. By the Henkin property, let $j$ be such that both $c(j)$ and $d(j)$ belong to $\Gamma^*$. The element $([[j], \forall])$ shows that $[c] \cap [d] \neq \emptyset$. That is, $N \models \exists(c, d)$.

Continuing, consider a sentence $b(j) \in \Gamma$. As $b \in Sub(\Gamma)$, we have $([[j], \exists)] = \Gamma$, so that $N \models b(j)$.

The work for sentences of the forms $s(j, k)$ and $\pi(j, k)$ was done in Lemma 8.14. The most intricate part of this proof concerns sentences $a(j, k), \pi(j, k) \in \Gamma$. Recall that we are dealing in this result with sentences of $L$, and so $j$ and $k$ are constant symbols of that language. Also recall that $[[j]] = ([j], \exists)$, and similarly for $k$.

Consider sentences in $\Gamma$ of the form $a(j, k)$. It is easy to see that if $\exists(c, a)(k)$ belongs to $\Gamma$, then so does $\exists(c, a)(j)$. (See the $\exists(c, a)$ case in Lemma 8.21.) From this it follows easily that $[[a]](\Gamma, [[k]])$. And so $N \models a(j, k)$ in this case.

We conclude with the consideration of a sentence in $\Gamma$ of the form $\pi(j, k)$. We wish to show that $N \models \pi(j, k)$. Suppose towards a contradiction that $N \models a(j, k)$. Then we have $[a](\Gamma, [[k]])$. There are two possibilities, corresponding to the alternatives in the semantics of $a$. The first is when there is a set term $c$ such that $\Gamma^*$ contains $c(k)$ and $\forall(c, a)(j)$. Using ($\forall E$), $\Gamma^*$ then contains $a(j, k)$. But recall that $\Gamma$ contains $\pi(j, k)$. So in this alternative, $\Gamma^* \supset \Gamma$ is inconsistent. In the second alternative, $j$ reaches $k$ by a chain of $\equiv$ and $a$ statements. By Lemma 8.19, $a(j, k) \in \Gamma^*$. So $\Gamma^*$ is inconsistent, and we have our contradiction. □ □

Once again, this gives us the finite model property for $L(adj)$. The result is not interesting from a complexity-theoretic point of view, since we already could see from Lutz and Sattler [13] that the logic had an EXPTime satisfiability problem.

Conclusion.
9.1. Introduction

To introduce the topic of monotonicity and polarity, let’s consider a very simple sentence:

(9.1) Every dog barks.

9.1.1. History. One enterprise at the border of logic and linguistics is the crafting of logical systems which capture aspects of inference in natural language. The ultimate goal is a logical system which is strong enough to represent the most common inferential phenomena in language and yet is weak enough to be decidable. Another goal would be to use representations which are as close as possible to the surface forms. This second goal is likely to be unobtainable, and when one backs off to consider structured representations, it is natural to turn to those suggested by linguistic formalisms of one form or another.

The particular strand of this enterprise that concerns us in this chapter is the natural logic program initiated by Johan van Benthem [33, 34], and elaborated in Víctor Sánchez Valencia [32]. As mentioned in van Benthem [35], the “proposed ingredients” of a logical system to satisfy the goals would consist of several “modules”:

1. Monotonicity Reasoning, i.e., Predicate Replacement,
2. Conservativity, i.e., Predicate Restriction, and also
3. Algebraic Laws for inferential features of specific lexical items.

We are only concerned with (1) in this chapter, and we are mainly concerned with an alternative to van Benthem’s seminal proposal.

There are several reasons why monotonicity is worthy of study on its own: it is arguably tied up with linguistic phenomena, it is a pervasive phenomenon in actual reasoning, and finds its way into computational systems for linguistic processing (cf. Nairn, Condoravdi and Karttunen [23] and MacCartney and Manning [14]). The leading idea in van Benthem’s work is to develop a formal approach to monotonicity on top of categorial grammar (CG). This makes sense because the semantics of CG is given in terms of functions, and monotonicity in the semantic sense is all about functions. The success story of the natural logic program is given by the monotonicity calculus, a way of determining the polarity of words in a given sentence. Before turning to it, we should explain the difference between monotonicity and polarity. We follow the usage of Bernardi [1], Section 4.1:

The differences between monotonicity and polarity could be summarized in a few words by saying that monotonicity is a property of functions . . . . On the other hand, polarity is a static syntactic notion which can be computed for all positions in a given
formula. This connection between the semantic notion of monotonicity and the syntactic one of polarity is what one needs to reach a proof theoretical account of natural reasoning and build a natural logic.

This point might be clarified with an example of the monotonicity calculus. Consider the sentence *No dog chased every cat*. The sentence is to be derived in some categorial grammar, and to be brief we suppress all the details on this except to say that a derivation would look like the trees below, except without the + and − marks. Given the derivation, one would first add some notations to indicate that the interpretation of *every*, as a function from noun meanings, is monotone decreasing but results in a function that is monotone increasing. Also, the interpretation of *no*, as a function from noun meanings, is monotone decreasing and again results in a function that is monotone decreasing.

Monotonicity Marking Polarity Determination

On the left, we have monotonicity markings in the derivation tree itself. These markings were propagated from the information about *every* and *no* according to a rule which need not concern us. One then turns the Monotonicity Marking tree into the Polarity Determination tree. The rule is: for each node, count the number of − signs from it to the root of the tree (at the bottom). This changes the signs for *every*, *cat*, and *every cat*. The resulting signs are the polarities. For example, the + marking on *cat* indicates that this occurrence of *cat* might be replaced by a word with a “larger” value. For example, replacing *cat* with *animal* gives a valid inference: *No dog chased every cat* implies *No dog chased every animal*. On the other hand, *dog* is marked −, corresponding to the fact that replacement inferences with it go in the opposite direction. So replacing *dog* with *animal* would not give a validity; instead, replace with the “smaller” phrase *old dog*.

As Dowty [4] mentions, “The goal [in his paper] is to ‘collapse’ the independent steps of Monotonicity Marking and Polarity Determination into the syntactic derivation itself, so that words and constituents are generated with the markings already in place that they would receive in Sánchez’ polarity summaries.” His motivation is mainly linguistic rather than logical: “the format of Sánchez’ system … is no doubt highly appropriate for the logical studies he developed it for, it is unsuitable for the linguistics applications I am interested in.” For this reason, he gave an “alternative formulation.” This system is “almost certainly equivalent to Sánchez’,” but he lacked a proof. Bernardi [1] also notes that while her system and Sánchez’ “are proven to be sound, no analogous result is known for” the system proposed by Dowty. However her presentation makes it clear that there are advantages to the internalization of polarity markers. Others who use a system inspired by [4] include Christodouloupolous [3].

1 Incidentally, we are taking this sentence and others like it in the subject wide-scope reading: there is no dog with the property that it chases every cat.
The purpose of this chapter is to offer a soundness proof for internalized monotonicity markings, and to discuss related issues. The first of these issues has to do with a choice in the overall categorial architecture. We wish to take seriously the idea that syntactic categories might be interpreted in ordered settings. Given two types $\sigma$ and $\tau$, we want to consider two resulting types: the upward monotone functions and the downward monotone functions. The first alternative is to generate types as follows: begin with a set $T_0$ of basic types, and then build the smallest set $T_1 \supseteq T_0$ closed in the following way:

$$ (9.2) \text{ If } \sigma, \tau \in T_1, \text{ then also } (\sigma, \tau)^+ \in T_1 \text{ and } (\sigma, \tau)^- \in T_1. $$

This would seem to be the most straightforward way to proceed, and it actually seems to be close to the way that is presented in the literature\(^2\). Indeed, we worked out the theory this way before deciding to rewrite this chapter based on a second alternative. This second way is based on the observation that a downward monotone function from a preorder $P$ to a preorder $Q$ is the same thing as an upward monotone function from $P$ to the opposite order $-Q$. This order $-Q$ has the same points as $Q$, but the order is “upside down.” The suggestion is that we generate types a little differently. We take smallest set $T_1 \supseteq T_0$ closed in the following way:

$$ (9.3) \text{ If } \sigma, \tau \in T_1, \text{ then also } (\sigma, \tau) \in T_1. $$

$$ (9.4) \text{ If } \sigma \in T_1, \text{ then also } -\sigma \in T_1. $$

We shall call this the fully internalized scheme. This includes copies of the types done the first way, by viewing $(\sigma, \tau)^+$ as $(\sigma, \tau)$ and $(\sigma, \tau)^-$ as $(\sigma, -\tau)$. It is a strictly bigger set, since $-\sigma$ for the basic types enters into the fully internalized hierarchy but not the lesser one.

9.2. Background: Categorial Grammar

This section offers a review of the essential facts concerning categorial grammar. For further background, and much more discussion of the subject, see Moortgat [16].

Fix a set $C_0$ of basic categories. Let $C$ be the smallest superset of $C_0$ closed in the following way:

$$ (9.5) \text{ If } C, D \in C, \text{ then also } C \setminus D \text{ and } C/D \text{ belong to } C. $$

Let $W$ be a set of words. A lexicon is a subset $L \subseteq W \times C$; that is, a lexicon is a set of pairs consisting of a word and a category. There is no requirement that this lexicon is function, and indeed it usually is not a function.

Continuing, we extend $L$ to $L^* \subseteq W^* \times C$ by the following two rules:

$$ (9.6) \text{ If } t : C \setminus D \text{ and } s : C, \text{ then } st : D. $$

$$ (9.7) \text{ If } t : C/D \text{ and } s : D, \text{ then } ts : D. $$

Here the notation $s : C$ means that $(s, C) \in L^*$.

An $AB$-grammar is a finite lexicon $L$ together with a distinguished category $S$. The language of $L$ is the set of all $t \in W^*$ such that $t : S$. These are called $AB$-grammars after Kazimierz Ajdukiewicz, the originator of categorial grammar, and Yehoshua Bar-Hillel, the one who added directionality to Ajdukiewicz’ system.

It is useful to recast (9.6) and (9.7) in the style of proof rules, as follows:

\(^2\)For example, Dowty [4] states his category formation rules in this way, adding directional versions. But shortly thereafter he mentions that lexical items will “be entered in two categories.” Adding downward-marked versions of the base categories is tantamount to what we call the fully internalized scheme.
That is, the lexicon generates proof trees. We declare the category of a proof tree \( T \) as the category of its root node, and we write \( T : C \) for this.

**Example 9.1** Let \( \mathcal{C}_0 = \{S,T,U\} \), let \( W = \{a,b,c\} \), and let \( \mathcal{L} \) be the following lexicon:

- \( a : S/T \)
- \( b : U \)
- \( b : U/U \)
- \( c : T \)
- \( b : T/S \)
- \( b : T/U \)
- \( c : T/S \)

An example of a word in the language of this lexicon \( \mathcal{L} \) is \( abbac \), via the following derivation:

- \( s : C \)
- \( t : C\backslash D \)
- \( st : D \)
- \( t : C/D \)
- \( s : D \)
- \( ts : D \)

Another example of a word in this language is \( abbacabbbbbb \). (Indeed, for those familiar with the notation of regular languages, the language of this grammar is exactly \((a(b^+e))^+\). This fact is not intended to be obvious, and indeed it takes some work to establish it.)

**Example 9.2** This time we take \( \mathcal{C}_0 = \{X,S\} \) and \( W = \{a,b\} \). The lexicon is

- \( a : S/X \)
- \( b : S\backslash X \)
- \( b : X \)
- \( c : T \)

This language is famous in formal language theory because it is context-free but not regular. For an example of a derivation, here is a reason why \( aabb \) is in the language:

\[
\begin{array}{c}
\hline
a: S/X & b: X \\
\hline
ab: S & b: S\backslash X \\
\hline
a: S/X & \hline
\end{array}
\]

Another example of a word in this language is \( aabbacabbbbbb \). (Indeed, for those familiar with the notation of regular languages, the language of this grammar is exactly \((a(b^+e))^+\). This fact is not intended to be obvious, and indeed it takes some work to establish it.)

**9.2.1. The syntax-semantics interface in AB-grammars.** Fix a set \( \mathcal{T}_0 \) of basic types. Let \( \mathcal{T} \) be the smallest superset of \( \mathcal{T}_0 \) closed in the following way:

\[
(9.9) \quad \text{If } \sigma, \tau \in \mathcal{T}, \text{ then also } (\sigma, \tau) \in \mathcal{T}.
\]

We associate (semantic) types to (syntactic) categories, starting with a map \( a : \mathcal{C}_0 \to \mathcal{T} \), and then extending via

\[
(9.10) \quad a(C\backslash D) = a(C/D) = (a(C), a(D)).
\]

A *model* is an assignment \( X_\sigma \) for \( \sigma \in \mathcal{T}_0 \), and it extends to the full set \( \mathcal{T} \) by

\[
(9.11) \quad X_{(\sigma, \tau)} = [X_\sigma, X_\tau], \text{ the set of all functions } f : X_\sigma \to X_\tau.
\]

Each entry in \( w : C \) in the lexicon then is given a semantics \( [w] \in X_{a(C)} \). To each derivation tree \( T : C \) then has an associated \( [T] \in X_{a(C)} \). The definition is by induction on \( T \). For \( T \) a one-point tree, \( T \) is essentially a typed word \( w : C \), and so \( [T] \) is given by the semantics of the lexicon. If \( T \) comes from (9.8), say from the first rule and via subtrees \( T_0 : C \) and \( T_1 : C\backslash D \), then \( [T] = [T_1][T_0] \).
9.3. Background: Preorders and their Opposites

For the work in the rest of this chapter we need to use preorders. Our goal in this section is to state the facts which are needed in as compact a manner as possible, and also to mention a few examples that will be used in the sequel.

A preorder is a pair $\mathcal{P} = (P, \leq)$ consisting of a set $P$ together with a relation $\leq$ which is reflexive and transitive. This means that the following hold:

1. $p \leq p$ for all $p \in P$.
2. If $p \leq q$ and $q \leq r$, then $p \leq r$.

Frequently one assumes that the order has additional properties: it may be also anti-symmetric, or a lattice order, etc. For most of what we are doing, no extra assumptions are needed. Indeed, the specific results do not use the preorder properties very much, and some of the work in this section goes through for absolutely arbitrary relations. However, since the intended application is to a setting in which the relation represents some kind of inference, it makes sense to state the results for the weakest class of mathematical objects that has something like “inference,” and this seems to be preorders.

**Example 9.3** For any set $X$, we have a preorder $X = (X, \leq)$, where $x \leq y$ iff $x = y$. This is called the flat preorder on $X$. More interestingly, for any set $X$ we have a preorder $\mathcal{P}(X)$ whose set part is the set of subsets of $X$, and where $p \leq q$ iff $p$ is a subset of $q$. Another preorder is $2 = \{F, T\} = (\{F, T\}, \sqsubseteq)$ with $F \sqsubseteq T$.

The natural class of maps between preorders $\mathcal{P}$ and $\mathcal{Q}$ is the set of monotone functions: the functions $f : P \to Q$ with the property that if $p \leq q$ in $\mathcal{P}$, then $f(p) \leq f(q)$ in $\mathcal{Q}$. When we write $f : \mathcal{P} \to \mathcal{Q}$ in this chapter, we mean that $f$ is monotone. We write $[\mathcal{P}, \mathcal{Q}]$ for the set of monotone functions from $\mathcal{P}$ to $\mathcal{Q}$. $[\mathcal{P}, \mathcal{Q}]$ is itself a preorder, with the pointwise order:

$$f \leq g \text{ in } [\mathcal{P}, \mathcal{Q}] \iff \text{ for all } p \in P, f(p) \leq g(p) \text{ in } \mathcal{Q}$$

Let $\mathcal{P}$ and $\mathcal{Q}$ be preorders. The product preorder $\mathcal{P} \times \mathcal{Q}$ is the preorder whose universe is the cartesian product $P \times Q$, and with $(p, q) \leq (p', q')$ iff $p \leq p'$ in $\mathcal{P}$ and $q \leq q'$ in $\mathcal{Q}$.

Preorders $\mathcal{P}$ and $\mathcal{Q}$ are isomorphic if there is a function $f : P \to Q$ which is bijective (one-to-one and maps $P$ onto $Q$) and also has the property that $p \leq p'$ iff $f(p) \leq f(p')$. We write $\mathcal{P} \cong \mathcal{Q}$. We also identify isomorphic preorders, and so we even write $\mathcal{P} = \mathcal{Q}$ to make this point.

**Proposition 9.4** For all preorders $\mathcal{P}$, $\mathcal{Q}$, and $\mathbb{R}$, and all sets $X$:

1. For each $p \in P$, the function $\text{app}_p : [\mathcal{P}, \mathcal{Q}] \to \mathcal{Q}$ given by $\text{app}_p(f) = f(p)$ is an element of $[\mathcal{P}, \mathcal{Q}]$.
2. $[\mathcal{P} \times \mathcal{Q}, \mathbb{R}] \cong [\mathcal{P}, [\mathcal{Q}, \mathbb{R}]]$.
3. $[X, 2] \cong \mathcal{P}(X)$.

**Proof** For part (1), let $f \leq g$ in $[\mathcal{P}, \mathcal{Q}]$. Then $\text{app}_p(f) = f(p) \leq g(p) = \text{app}_p(g)$.

In the second part, let $f : [\mathcal{P} \times \mathcal{Q}, \mathbb{R}] \to [\mathcal{P}, [\mathcal{Q}, \mathbb{R}]]$ be $f(a)(p)(q) = a(p, q)$. It is routine to check that $f$ is an isomorphism.

In part (3), the isomorphism is $\varphi : [X, 2] \to \mathcal{P}(X)$ given by $\varphi(f) = \{p \in P : f(p) = \top\}$.

$\square$
Antitone functions and opposites of preorders. We are also going to be interested in antitone functions from a preorder $\mathbb{P}$ to a preorder $\mathbb{Q}$. These are the functions $f : \mathbb{P} \to \mathbb{Q}$ with the property that if $p \leq q$ in $\mathbb{P}$, then $f(q) \leq f(p)$ in $\mathbb{Q}$. We can express things in a more elegant way using the concept of the opposite preorder $-\mathbb{P}$ of a given preorder $\mathbb{P}$.

This is the preorder with the same set part, but with the opposite order. Formally, $-\mathbb{P} = \mathbb{P}$ with $p \leq q$ in $-\mathbb{P}$ iff $q \leq p$ in $\mathbb{P}$.

Exercise 34 This exercise and the next ones ask you to classify some functions as monotone or antitone. To start, look at or draw pictures of $2$, $2^{\text{op}}$, $2 \times 2$, $(2 \times 2)^{\text{op}}$, $2^{\text{op}} \times 2$, and $2 \times 2^{\text{op}}$.

1. Consider the negation function on $\{T, F\}$. Which of the following is true, and which false: $\neg : 2 \to -2$, or $\neg : 2 \to 2$, So would we say that $\neg$ is a monotone function on $2$, or an antitone function?

2. Is the conjunction operation $\land$ a monotone function on from $2 \times 2$ to $2$ or an antitone function, or neither?

3. Is the implication operation $\to$ a monotone function on from $2 \times 2$ to $2$ or an antitone function, or neither?

Exercise 35 Let $\mathbb{P}$ be any preorder, and let $\mathbb{Q}$ be $\mathcal{P}(\mathbb{P})$, the power set preorder on the underlying set $P$ of $\mathbb{P}$. Let $\uparrow : P \to -\mathbb{Q}$ be defined by

$$\uparrow (p) = \{ p' \in P : p \leq p' \text{ in } \mathbb{P} \}.$$
Show that \( \uparrow : P \to \neg Q \).

**Exercise 36** Can a function \( f : P \to Q \) be both monotone and antitone at the same time? Show that \( f \)

We collect the most important facts on this operation below.

**Proposition 9.5** For all preorders \( P, Q, \) and \( R, \) and all sets \( X, \)

(1) \(-(-P) = P.\)
(2) \([P, -Q] = [-P, Q].\)
(3) \([-P, -Q] = [-P, Q].\)
(4) If \( f : P \to Q \) and \( g : Q \to R, \) then \( g \cdot f : P \to R.\)
(5) If \( f : P \to -Q \) and \( g : Q \to -R, \) then \( g \cdot f : P \to R.\)
(6) \(-P \times Q \simeq -P \times -Q.\)
(7) \( X \simeq X.\)
(8) \([-X, P] \simeq [X, -P].\)

**Proof** For part (1), recall first that \( P \) and \(-P\) also have this same underlying set. Concerning the order: \( x \leq y \) in \( P \) iff \( y \leq x \) in \(-P\).

For (2), suppose that \( f : P \to -Q.\) Then \( f \) is a set function from \( P \) to \( Q.\) To see that \( f \) is a monotone function from \(-P\) to \( Q,\) let \( x \leq y \) in \(-P.\) Then \( y \leq x \) in \( P.\) So \( f(y) \leq f(x) \) in \(-Q.\) And thus \( f(x) \leq f(y) \) in \( Q,\) as desired. This verifies that as sets, \([P, -Q] = [-P, Q].\)

To show that as preorders \([P, -Q] = [-P, Q],\) we must check that if \( f \leq g \) in the order on \([P, -Q],\) then \( g \leq f \) in the order on \([-P, Q] \) as well. For this, let \( p \in -P; \) so \( p \in P.\) Then \( f(p) \leq g(p) \) in \(-Q.\) Thus \( g(p) \leq f(p) \) in \( Q.\) This for all \( p \in P \) shows that \( g \leq f \) in \([-P, Q].\)

Part (3) follows from parts (1) and (2).

For part (4), assume \( x \leq y.\) Then \( f(x) \leq f(y), \) and so \( g(f(x)) \leq g(f(y)).\)

For part (5), note that by part (2), \( g : -Q \to R.\) So by part (4), \( g \cdot f : P \to R.\)

Continuing with part (6), note first that \(-P \times Q \) and \(-P \times -Q \) have the same elements. To check that the orders are appropriately related, note that the following are equivalent: \((p,q) \leq (p',q') \) in \(-P \times Q; \) \((p',q') \leq (p,q) \) in \( P \times Q; \)

\(p' \leq p \) in \( P \) and \( q' \leq q \) in \( Q; \) \( p \leq p' \) in \(-P \) and \( q \leq q' \) in \(-Q; \)

\(p,q) \leq (p',q') \) in \(-P \times -Q.\)

For part (7), we verify that the identity map on \( P \) is an isomorphism from \( P \) to \(-P.\) If \( p \leq q \) in \( P, \) then clearly \( q \leq p \) in \(-P.\) And if \( q \leq p \) in \(-P, \) then since \( P \)

is flat, \( p = q.\) Thus \( p \leq q \) in \( P.\)

The last part comes from parts (3) and (8).

**Example 9.6** Part (3) in Proposition 9.5 is illustrated by considering \([2, 2] \) and \([-2, -2].\) Let \( c \) be the constant function \( T, \) let \( d \) be the constant function \( F, \)

and let \( i \) be the identity. Then \([2, 2] \) is \( d \) \(< i \) \(< c, \) and \([-2, -2] \) is \( c \) \(< i \) \(< d.\)

Part (2) is illustrated by \([-2, 2] \) and \([2, -2].\) Let us write \( \neg \) for the negation function on truth values which we saw in Example ???. Then \([-2, 2] \) is \( d \) \(< \neg \) \(< c, \) and \([2, -2] \) is \( c \) \(< \neg \) \(< d.\)

**Example 9.7** The classical conditional \( \to \) gives a monotone function from \(-2 \times 2 \) to \( 2.\) Since \([-2 \times 2, 2] \simeq [-2, [2, 2]],\) this gives

\[\text{if } \in [-2, [2, 2]].\]
This same function belongs to the opposite preorder, and so we could just as well write
\[ \text{if } \in \sim \sim \quad \sim \sim \sim \]

**Example 9.8** Let \( X \) be any set. We use letters like \( p \) and \( q \) to denote elements of \([X, 2]\). Please keep in mind that the set of (monotone) functions from \( X \) to 2 is in one-to-one correspondence with the set of subsets of \( X \) (see Part (3) of Proposition 9.4). Define

\[
\begin{align*}
\text{every} & \in \sim \sim \sim \sim \\
\text{some} & \in \sim \sim \sim \\
\text{no} & \in \sim \sim \sim
\end{align*}
\]

in the standard way:

\[
\begin{align*}
\text{every}(p)(q) &= \begin{cases} T & \text{if } p \leq q \\ F & \text{otherwise} \end{cases} \\
\text{some}(p)(q) &= \neg \text{every}(\neg \cdot q) \\
\text{no}(p)(q) &= \neg \text{some}(p)(q)
\end{align*}
\]

It is routine to verify that these functions really belong to the sets mentioned above. And as in Example 9.7, each of these functions belongs to the opposite preorder as well, and we therefore have

\[
\begin{align*}
\text{every} & \in \sim \sim \sim \sim \\
\text{some} & \in \sim \sim \sim \\
\text{no} & \in \sim \sim \sim
\end{align*}
\]

Summary. The main point of this section is the fact mentioned in Proposition 9.5, part (3). In words, the monotone functions from \( -P \) to \( -Q \) are exactly the same as the monotone functions from \( P \) to \( Q \), but as preordered sets themselves, \([P, Q]\) and \([P, Q] \sim \sim \sim \) are opposite orders. A simple illustration of the opposite orders was presented in Example 9.6. This fact, and also part (2) of Proposition 9.5, is important in the linguistic examples because it justifies why lexical items are typically assigned two categories. For example, if we interpret a common noun by an element \( f \) of a preorder of the form \([X, 2]\), then the same \( f \) is an element of \( -[X, 2] \), so we might as declare our noun to also have some typing which calls for an element of \(-[X, 2] \). And in general, if we type a lexical item \( w \) with a type \((\sigma, \tau) \) corresponding to an element of some preorder \([P, Q] \), then we might as well endow \( w \) with the type \((-\sigma, -\tau) \) with the very same interpretation function.

### 9.4. Higher-order Terms over Preorders

Fix a set \( \mathcal{J}_0 \) of basic types. Let \( \mathcal{J}_1 \) be the smallest superset of \( \mathcal{J}_0 \) closed in the following way:

\[
\begin{align*}
(9.12) & \quad \text{If } \sigma, \tau \in \mathcal{J}_1, \text{ then also } (\sigma, \tau) \in \mathcal{J}_1. \\
(9.13) & \quad \text{If } \sigma \in \mathcal{J}_1, \text{ then also } -\sigma \in \mathcal{J}_1.
\end{align*}
\]

Let \( \equiv \) be the smallest equivalence relation on \( \mathcal{J}_1 \) such that the following hold:

\[
(1) \quad -(-\sigma) \equiv \sigma.
\]
9.4. HIGHER-ORDER TERMS OVER PREORDERS

(2) $- (\sigma, \tau) \equiv (-\sigma, -\tau)$.

(3) If $\sigma \equiv \sigma'$, then also $-\sigma \equiv -\sigma'$.

(4) If $\sigma \equiv \sigma'$ and $\tau \equiv \tau'$, then $(\sigma, \tau) \equiv (\sigma', \tau')$.

**Definition 9.9** \( \mathcal{T} = \mathcal{T}_1 / \equiv \). This is the set of types over \( \mathcal{T}_0 \).

The operations $\sigma \mapsto -\sigma$ and $\sigma, \tau \mapsto (\sigma, \tau)$ are well-defined on \( \mathcal{T} \). We always use letters like $\sigma$ and $\tau$ to denote elements of \( \mathcal{T} \), as opposed to writing \([\sigma]\) and \([\tau]\). That is, we simply work with the elements of \( \mathcal{T}_1 \), but identify equivalent types.

**Definition 9.10** Let \( \mathcal{T}_0 \) be a set of basic types. A typed language over \( \mathcal{T}_0 \) is a collection of typed variables $v : \sigma$ and typed constants $c : \sigma$, where $\sigma$ in each of these is an element of \( \mathcal{T} \). We generally assume that the set of typed variables includes infinitely many of each type. But there might be no constants whatsoever. We use \( \mathcal{L} \) to denote a typed language in this sense.

Let \( \mathcal{L} \) be a typed language. We form typed terms $t : \sigma$ as follows:

1. If $v : \sigma$ (as a typed variable), then $v : \sigma$ (as a typed term).
2. If $c : \sigma$ (as a typed constant), then $c : \sigma$ (as a typed term).
3. If $t : (\sigma, \tau)$ and $u : \sigma$, then $t(u) : \tau$.

Frequently we do not display the types of our terms.

**9.4.1. Semantics.** For the semantics of our higher-order language \( \mathcal{L} \) we use models \( \mathcal{M} \) of the following form. \( \mathcal{M} \) consists of an assignment of preorders $\sigma \mapsto P_\sigma$ on \( \mathcal{T}_0 \), together with some data which we shall mention shortly. Before this, extend the assignment $\sigma \mapsto P_\sigma$ to \( \mathcal{T}_1 \) by

\[
\begin{align*}
P_{(\sigma, \tau)} &= [P_\sigma, P_\tau] \\
P_{-\sigma} &= \neg P_\sigma
\end{align*}
\]

An easy induction shows that if $\sigma \equiv \tau$, then $P_\sigma = P_\tau$. So we have $P_\sigma$ for $\sigma \in \mathcal{T}$. We use $P_\sigma$ to denote the set underlying the preorder $P_\sigma$.

The rest of the structure of a model \( \mathcal{M} \) consists of an assignment $[c] \in P_\sigma$ for each constant $c : \sigma$, and also a typed map $f$; this is just a map which to a typed variable $v : \sigma$ gives some $f(v) \in P_\sigma$.

**9.4.2. Ground terms and contexts.** A ground term is a term with no free variables. Each ground term $t : \sigma$ has a denotation $[t] \in P_\sigma$ defined in the obvious way:

\[
\begin{align*}
[c] &= \text{is given at the outset for constants } c : \sigma \\
[t(u)] &= [t][u]
\end{align*}
\]

A context is a typed term with exactly one variable, $x$. (This variable may be of any type.) We write $t$ for a context. We’ll be interested in contexts of the form $t(u)$. Note that if $t(u)$ is a context and if $x$ appears in $u$, then $t$ is a ground term; and vice-versa.

In the definition below, we remind you that subterms are not necessarily proper. That is, a variable $x$ is a subterm of itself.

**Definition 9.11** Fix a model \( \mathcal{M} \) for \( \mathcal{L} \). Let $x : \rho$, and let $t : \sigma$ be a context. We associate to $t$ a set function

\[ f_t : P_\rho \to P_\sigma \]

in the following way:
(1) If \( t = x \), so that \( \sigma = \rho \), then \( f_x : P_\sigma \to P_\sigma \) is the identity.

(2) If \( t \) is \( u(\nu) \) with \( u : (\tau,\sigma) \) and \( \nu : \tau \), and if \( x \) is a subterm of \( u \), then \( f_t \) is

\[
\text{app}_{\nu} : f_u. \quad \text{That is, } f_t(\nu) = a \in P_\rho \mapsto f_u(a)(\nu)
\]

(3) If \( t \) is \( u(\nu) \) with \( u : (\tau,\sigma) \) and \( \nu : \tau \), and if \( x \) is a subterm of \( v \), then \( f_t \) is

\[
\text{app} : f_v. \quad \text{That is, } f_t = a \in P_\rho \mapsto [u](f_v(a)).
\]

The idea of \( f_t \) is that as \( a \) ranges over its interpretation space \( P_\rho \), \( f_t(a) \) would be the result of substituting various values of this space in for the variable, and then evaluating the result.

Notice that we defined \( f_t \) as a set function and wrote \( f_t : P_\rho \to P_\sigma \) instead of \( f_t : P_\rho \to P_\sigma \). The reason why we did this is that it is not immediately clear that \( f_t \) is monotone. This is the content of the following result.

**Lemma 9.12 (Context Lemma)** Let \( t \) be a context, where \( t : \sigma \) and \( x : \rho \). Then \( f_t \) is element of \( [\rho,\rho,\sigma] \).

**Proof** By induction on \( t \). In the case that \( t \) is a type variable \( x : \sigma \), then \( f_x \) is the identity on \( P_\sigma \), and indeed the identity is a monotone function on any preorder.

Next, assume that \( x \) occurs in \( u \). By induction hypothesis, \( f_u \) belongs to \( [\rho,\rho,\tau,\sigma] = [\rho,\rho,\rho,\rho] \). The function \( \text{app}_{\nu} \) belongs to \( [[\rho,\rho,\tau,\sigma],\rho,\rho,\rho,\rho] \) by Proposition 9.4, part (1), and so \( f_t \) belongs to \( [\rho,\rho,\rho,\rho,\rho] \) by Proposition 9.5, part (4).

Finally, assume that \( x \) occurs in \( v \). By induction hypothesis, \( f_v : P_\rho \to P_\tau \) and \( [u] \in P_{(\tau,\sigma)} \), so that \( [u] : P_\tau \to P_\sigma \). By Proposition 9.5, part (4), \( [u] \cdot f_v \) is an element of \( [\rho,\rho,\rho,\rho,\rho] \).

This Context Lemma is exactly the soundness fact underlying internalized polarity markers. We’ll see some examples in the next section. Notice also that it is a very general result: the only facts about preorders were the very general results in Section 9.3.

The Context Lemma also allows one to generalize our work from the applicative (AB) grammars to the setting of CG using the Lambek Calculus. In detail, one generalizes the notion of a context to one which allows more than one free variable but requires that all variables which occur free have only one free occurrence. Suppose that \( x_1 : \rho_1, \ldots, x_n : \rho_n \) are the free variables in such a generalized context \( t(x_1, \ldots, x_n) : \sigma \). Then \( t \) defines a set function a set function \( f_t : \prod_i P_{\rho_i} \to P_\sigma \). A generalization of the Context Lemma then shows that \( f_t \) is monotone as a function from the product preorder \( \prod_i P_\rho \). So functions given by lambda abstraction on each of the variables are also monotone, and this amounts to the soundness of the introduction rules of the Lambek Calculus in the internalized setting.

This point an advantage of the internalized system in comparison to the monotonicity calculus: in the latter approach, the results of polarity determination become side conditions on the introduction rules for the lambda operator.

**9.4.3. Logic.** Figure 9.2 several sound logical principles. These are implicit in Fyodorov, Winter, and Francez [6]; see also Zamansky, Francez, and Winter [39] for a more developed proposal. The rules the reflexive and transitive properties of the preorder, the fact that all function types are interpreted by sets of monotone
functions, and the pointwise definition of the order on function sets. The rules in Figure 9.2 define a logical system whose assertions order statements such as \( u : \sigma \leq v : \sigma \); then the statements such as \( u : \sigma \) become side conditions on the rules. (For simplicity, we are only dealing with ground terms \( u \) and \( v \), but it is not hard to generalize the treatment to allow variables.) The logic thus defines a relation \( \Gamma \vdash \varphi \) on order statements. Given a model \( M \) and an order statement \( \psi \) of the form \( u : \sigma \leq v : \sigma \), we say that \( M \) satisfies \( \psi \) iff \([u]\leq [v]\) in \( P_{\sigma} \). The soundness of the logic is the statement that every model \( M \) satisfying all of the sentences in \( \Gamma \) also satisfies \( \varphi \).

The Context Lemma then gives another sound principle:

\[
\begin{align*}
& x : \sigma \text{ in the context } t \\
& u : \sigma \leq v : \sigma
\end{align*}
\]

\[
\frac{t(u) : \tau \leq t(v) : \tau}{t(u) : \tau \leq t(v) : \tau}
\]

However, all instances of (9.14) are already provable in the logic as we have defined it. This is an easy inductive argument on the context \( t \).

### 9.5. Examples and Discussion

We present a small example to illustrate the ideas. It is a version of the language \( \mathcal{RE} \) which we saw in Chapter 6. We also take the opportunity to discuss internalized marking in a general way. We have a few comments on the challenges of building a logic based on internalized markings. And we also have a discussion of the word any and how it might be treated.

First, we describe a language \( \mathcal{L} \) corresponding to this vocabulary. Let’s take our set \( T_0 \) of basic types to be \( \{t, pr\} \). (These stand for truth value and property. In more traditional presentations, the type \( pr \) might be \( (e, t) \), where \( e \) is a type of entities.)

Here are the constants of the language \( \mathcal{L} \) and their types:

1. We have typed constants

\[
\begin{align*}
\text{every}^+ & : (-pr, (pr, t)) & \text{every}^- & : (pr, (-pr, -t)) \\
\text{some}^+ & : (pr, (pr, t)) & \text{some}^- & : (-pr, (-pr, -t)) \\
\text{no}^+ & : (-pr, (-pr, t)) & \text{no}^- & : (pr, (pr, -t)) \\
\text{any}^+ & : (-pr, (pr, t)) & \text{any}^- & : (-pr, (-pr, -t))
\end{align*}
\]

2. We fix a set of unary atoms corresponding to some plural nouns and lexical verb phrases in English. For definiteness, we take \text{cat, dog, animal,}

(3) We also fix a set of binary atoms corresponding to some transitive verbs in English. To be definite, we take chase, see. Every binary atom $r$ gives four type constants:

- $r^+_1 : ((pr,t),pr)$
- $r^+_2 : ((-pr,t),-pr)$
- $r^-_1 : ((-pr,-t),-pr)$
- $r^-_2 : ((pr,-t),-pr)$

This completes the definition of our typed language $L$.

We might mention that the superscripts + and − on the constants $c : \sigma$ of $L$ are exactly the polarities $\text{pol}(\sigma)$ of the associated types as we defined them in the previous section. These notations + and − are mnemonic; we could do without them.

As always with categorial grammars, a lexicon is a set of pairs consisting of words in a natural language together with terms. We have been writing the words in the target language in italics, and then terms for them are written in sans serif. It is very important that the lexicon allows a given word to appear with many terms. As we have seen, we need every to appear with every$^+$ and every$^-$, for example. We still are only concerned with the syntax at this point, and the semantics will enter once we have seen some examples.

Examples of typed terms and contexts. Here are a few examples of typed terms along with their derivations. First, here is a derivation of the term corresponding to the English sentence No dog chases every cat:

\[
\begin{aligned}
\text{no}^+ & : (\neg pr,(\neg pr,t)) & \text{dog}^- & : \neg pr \\
\text{chase}^-_1 & : ((\neg pr,-t),\neg pr) & \text{every}^- & : (pr,(\neg pr,-t)) \\
\text{no}^+(\text{dog}^-) & : (\neg pr,t) & \text{chase}^-_1(\text{every}^-(\text{cat}^+)) & : \neg pr \\
\text{no}^+(\text{dog}^-)(\text{chase}^-_1(\text{every}^-(\text{cat}^+))) & : t
\end{aligned}
\]

One should compare this treatment with what we saw of the same sentence in the Introduction. The internalized setting does not have steps of Monotonicity Marking and Polarity Determination. It only involves a derivation in the applicative CG at hand. Please also keep in mind that the superscripts + and − above are from the lexicon, not from any additional process, and that the lexicon could have been formulated without those superscripts. They are in fact the polarities of the attendant categories. (For example, every$^-$ is of type $pr,(\neg pr,-t)$, and the polarity of its occurrence above is $\neg$.)

We similarly have the following terms:

- $\text{some}^+(\text{dog}^+)(\text{chase}^+_1(\text{every}^+(\text{cat}^-))) : t$
- $\text{some}^+(\text{dog}^+)(\text{chase}^+_2(\text{no}^+(\text{cat}^-))) : t$
- $\text{no}^+(\text{dog}^-)(\text{chase}^-_2(\text{no}^+(\text{cat}^-))) : t$

The point here is that all four different typings of the transitive verbs are needed to represent sentences of English. I do not believe that this point has been noticed about the internalized scheme before. It raises an issue that we shall take up in the next subsection: how would a grammar writer come up with all of the needed categories?

Here is an example of a context: $\text{no}^+(x : \neg pr)(\text{chase}^-_1(\text{every}^-(\text{cat}^+))) : t$. So $x$ is a variable of type $\neg pr$. In any model, this context gives a function from interpretations of type $\neg pr$ to those of type $\neg t$. The Context Lemma would tell us
that this function is a monotone function. Turning things around, it would be an antitone function from interpretations of type \(-pr\) to those of type \(-t\).

Any. The internalized approach enables a treatment of any that has it any mean the same thing as every when it has positive polarity, and the same thing as some when it has negative polarity. For example, here is a sentence intended to mean everything which sees any cat runs:

\[
\begin{align*}
\text{any}^- : \langle -pr, (pr, -t) \rangle & \quad \text{cat}^- : -pr \\
\text{see}_2^- : \langle (pr, -t), -pr \rangle & \quad \text{any}^-(\text{cat}^-) : \langle -pr, -t \rangle \\
\text{every}^+ : \langle -pr, (pr, t) \rangle & \quad \text{see}_2^- (\text{any}^-(\text{cat}^-)) : -pr \\
\text{every}^+ (\text{see}_2^- (\text{any}^-(\text{cat}^-))) : (pr, t) & \quad \text{runs}^+ : pr \\
\end{align*}
\]

The natural reading is for any to have an existential reading. Another context for existential readings of any is in the antecedent of a conditional. In the other direction, consider

\[
\text{any}^+(\text{cat}^-) (\text{see}_1^- (\text{any}^+(\text{dog}^-))) : t.
\]

The natural reading of both occurrences is universal.

9.5.1. Standard models. Up until now, we have only given one example of a typed language \(\mathcal{L}\). Now we describe a family of models for this language. The family is based sets with interpretations of the unary and binary atoms. To make this precise, let us call a pre-model a structure \(M_0\) consisting of a set \(M\) together with subsets \([p] \subseteq M\) for unary atoms \(p\) and relations \([r] \subseteq M \times M\) for binary atoms \(r\).

Every pre-model \(M_0\) now gives a bona fide model \(M\) of \(\mathcal{L}\) in the following way. The underlying universe \(M\) gives a flat preorder \(\mathcal{M}\). We take \(\mathbb{P}_{pr} = [\mathcal{M}, 2] \cong \mathcal{P}(\mathcal{M})\). We also take \(\mathbb{P}_t = 2\).

We interpret the typed constants \(p^+ : pr\) corresponding to unary atoms by

\[
[p^+] (m) = T \quad \text{iff} \quad m \in [p].
\]

(On the right we use the interpretation of \(p\) in the model \(M\).) Usually we write \(p^+\) instead of \([p]\).

The constants \(p^- : -pr\) are interpreted by the same functions.

The interpretations of every\(^+\), some\(^+\), and no\(^+\) are given in Example 9.8 (taking the set \(X\) to be the universe \(M\) of \(M\)). And the interpretations of every\(^-\), some\(^-\), and no\(^-\) are the same functions.

We interpret any\(^+\) the same way as every\(^+\), and any\(^-\) the same way as some\(^+\).

Recall that the binary atom \(r\) gives four typed constants \(r_1^+, r_1^-, r_2^+, r_2^-\). These are all interpreted in model in the same way, by

\[
r_\ast (q) (m) = q (\{ m' \in M : [r] (m, m') \})
\]

It is clear that for all \(q \in \mathbb{P}_{(pr, t)} = [[\mathcal{M}, 2], 2]\), \(r_\ast (q)\) is a function from \(M\) to \(\{T, F\}\). The monotonicity of this function is trivial, since \(\mathcal{M}\) is flat.

We do need to check that all of the interpretations of the constants are correctly typed. This is important, and it raises a general issue.
Issue 1: What are the sources of the multiple typings of the lexical items? As this chapter draws to a close, we wish to identify some issues with the internalized scheme of polarity marking. Here is the first: The scheme requires many entries for each lexical item, usually with the same semantics in every model. These amount to multiple typings of the same function. What is the source of these multiple typings, and can they be automatically inferred?

The principal source for multiple typings is the fact noted in Proposition 9.5, part (3): \([-\mathcal{P}, -\mathcal{Q}] = -[\mathcal{P}, \mathcal{Q}]\). That is, we have two (opposite) order relations on the same set, and so it makes sense that we should type an element of that set in two ways. This accounts for the multiple typings of nouns and determiners in our example, but not for transitive verbs. For them, and for all categories in more complicated languages, we suspect that there is no simple answer. Perhaps the best one could hope for would be a formal system in which to derive the various typings. In the same way that one could propose logical systems which process inferences in parallel with syntactic derivations, one could hope to do the same thing for order statements. Here is our vision of what a derivation might look like in such a system. Let us write \(\varphi(m, r)\) for \(\{\{m', \mathcal{M} : [r] (m, m')\}\}\). A justification for the typing \(r^{-2} : [[\mathcal{M}, 2], -2], [\mathcal{M}, -2]]\) could be the derivation below:

\[
\begin{align*}
q & \le q' \text{ in } [[\mathcal{M}, 2], -2], \varphi(m, r) \in [\mathcal{M}, 2] \\
r_2^{-2}(q) &= q(\varphi(m, r)) \le q'(\varphi(m, r)) = r_2^{-2}(q') \text{ in } -2 \\
r_2^{-2}(q) & \le r_2^{-2}(q') \text{ in } [\mathcal{M}, -2] \\
r_2^{-2} : [[\mathcal{M}, 2], -2], [\mathcal{M}, -2]]
\end{align*}
\]

The challenge would be to propose a system which is small enough to be decidable, and yet big enough to allow us to use the notation \(\varphi(m, r)\) as we have done it.

Here is a related point. As Dowty [4] notes, the parsing problem for internalized grammars is no harder than that of CG in the first place. In fact, the work for internalized grammars is a special case, since they are CGs with nothing extra. However, this could be a misleading comparison: if one is given a CG \(\mathcal{G}\) and some additional information (for example, the intended meanings of the constants corresponding to determiners), one would need to convert this to an internalized grammar \(\mathcal{G}^*\). It may be that the size of the lexicon of \(\mathcal{G}^*\) is exponential in the size of the lexicon of \(\mathcal{G}\). The fact that in our small grammar we need four typings for the verbs makes one worry. And in the worst case, the internalized scheme would be less attractive for computational purposes.

Issue 2: what is the relation of polarity and monotonicity? It is “well-known” in this area that negative polarity items are those which must occur, or usually occur in “downward monotone positions” in a given sentence. But without saying what the orders on the semantic spaces, this point could be confusing. For example, suppose one has a reason to assume that the order on truth values is \(T < F\) rather than \(F < T\). In this case, the notions of upward monotone positions and downward monotone ones would switch. More seriously, given a complicated function type \((\sigma, -\tau)\), it is not so clear whether this should be recast as \(- (\sigma, \tau)\). In this case, it would not be so clear what the upward and downward monotone positions are in a sentence.

This matter might be clarified by looking at no vs. at most one in a sentence such as No dog chases every cat. As we have seen, our treatment suggests that the occurrence of no is in a positive position, and so we expect it to be monotone.
increasing. We would like to say that no \( \leq \) at most one, since this is the verdict from generalized quantifier theory. Indeed, the interpretations functions no and at most one are taken to be functions of type \(((M, 2), (M, 2), 2))\), and then the order on this space is taken to be ultimately determined by the last “2”. However, we do not have a single function no but rather two functions, no\(^+\) and no\(^−\). These are not quite restrictions of no and at most one. For that matter, why exactly do we say that no\(^+\) is monotone increasing? It’s type was \((-pr, (-pr, t))\), after all. What seems to be going on here is that in a curried form, we have \((-pr \times -pr, t)\). When curried like this, no\(^+\) and no\(^−\) are restrictions of no and at most one. Moreover, the order on on a product space is determined by the codomain order, forgetting the domain order. Also, the “meaning space” in a model for types pr and \(-pr\) are the same. Taken together, this means that no\(^+\) indeed corresponds to a monotone function and no\(^−\) to an antitone function.

Here is a general definition of the polarity of a type. Define \(\text{pol}(\sigma) : T_1 \rightarrow \{+, −\}\) as follows:

1. If \(\sigma \in T_0\), then \(\text{pol}(\sigma) = +\).
2. \(\text{pol}((\sigma, \tau)) = \text{pol}(\tau)\).
3. \(\text{pol}(−\sigma) = −\text{pol}(\sigma)\).

It is easy to check that if \(\sigma \equiv \tau\), then \(\text{pol}(\sigma) = \text{pol}(\tau)\). (It is enough to show that \(\text{pol}(−(−\sigma)) = \text{pol}(\sigma)\), and also that \(\text{pol}(−(\sigma, \tau)) = \text{pol}((−\sigma, −\tau))\).) This allows us define \(\text{pol}\) on the quotient set, \(T_1/\equiv\), our set \(T\) of types.

The definition of the \(\text{pol}\) function gives an algorithm to calculate \(\text{pol}(\sigma)\). Another way to do this would be to: (1) put \(\sigma\) in “negation normal form”, by driving the - signs through function spaces so that there are no function types inside the scope of any - sign; (2) remove an even number of - signs from basic types, so that the remaining basic types have either zero or one - sign in front; (3), finally, starting at the top, proceed “to the right” through the term until one reaches either a basic type or its negation.

Let \(u\) be a subterm occurrence in the term \(t\). Then there is a unique context \(t′(x)\) such that \(t = t′[u \leftarrow x]\). Let \(\sigma\) be the type of \(x\) in \(t′\). We say that \(u\) has positive polarity if \(\text{pol}(x) = +\), and that \(u\) has negative polarity if \(\text{pol}(x) = −\).

We naturally would like to say that if an occurrence \(u\) has positive polarity in \(t\), then that occurrence is in a monotone increasing position, and if the occurrence has negative polarity, then it is in a monotone decreasing position.

Issue 3: how can we build a logic to reason about classes of standard models? Our last issue concerns the contribution of the types in an internalized CG to a natural logic. The matter is illustrated by a discussion concerning the example language \(L\) presented at the beginning of Section 9.5. Suppose we are given entailment relations concerning the unary and binary atoms of \(L\). For unary atoms, these take the form \(p \Rightarrow q\), where \(p\) and \(q\) are unary atoms. For binary atoms, entailment relations look similarly: \(r \Rightarrow s\). (Incidentally, one is also interested in exclusion relations: see MacCartney and Manning [14] for applications, and Icard [10] for theory behind this.)

These entailment relations correspond to semantic restrictions on standard models. The first corresponds to the requirement that \(\llbracket p \rrbracket \subseteq \llbracket q \rrbracket\), and the second to the parallel requirement that \(\llbracket r \rrbracket \subseteq \llbracket s \rrbracket\). Working with entailment relations in this way means restricting attention to standard models which conform to the restrictions. It is natural to then look for additional axioms and/or rules of inference...
in the logical system that would give a sound and complete logic for these models. The entailment relations correspond to axioms of the sentential kind, and also on the orders. The sentential axioms can be stated only for the unary atoms, and they would be sentences like \((\text{every}^+(p^-))q^-\). For the binary atoms, the statements are more involved and we leave them as exercises. Turning to axioms on the orders, our entailment relations give us the following:

\[
\begin{align*}
    p^+ & \leq q^+ \text{ in } pr \\
    q^- & \leq p^- \text{ in } -pr \\
    r_1^+ & \leq s_1^+ \text{ in } ((-pr, t), -pr) \\
    r_2^- & \leq s_2^- \text{ in } ((-pr, t), pr)
\end{align*}
\]

However, even adopting both kinds of axioms on top of the logical principles we saw in Section 9.4.3 would not result in a complete logical system, so it would fall short of what we want in a natural logic. One source of problems was noted in Zamansky, Francez, and Winter [39]: the system would not have any way to derive principles corresponding to de Morgan’s laws. Another problem for us would be even more basic. The resulting logic does not appear to be strong enough to derive the sound sentences of the form every \(X\) is an \(X\). To be sure, there are unsound sentences of this form in the logic, due to the typing. For example, the way we are doing things, every dog who chases any cat chases any cat is not a logical validity at all, since the first any is to be interpreted existentially and the second universally. (I first heard an example of this type in a lecture of Barbara Partee.) But without any, we would like to derive, for example

\[\text{every}^+(\text{see}^-((\text{every}^-(\text{cat}^+))))(\text{see}^+(\text{every}^-((\text{cat}^-))))\]

At this point, I can merely speculate and offer a suggestion. My feeling is that one should look to logics with individual variables, and then the properties of the individual words become rules of inference in the logic. Specifically, it should be the case that if \(x\) is an individual or NP-level variable, then one should be able to infer \(\text{cat}^+(x)\) from \(\text{cat}^-(x)\), and vice-versa. For more on logics of this type, see [19]. However, that paper does not build on the framework of categorial grammar and has nothing to say about polarity and monotonicity. Integrating the work there with the proposal in this chapter is the most important open issue in work on this topic.

Conclusion. This chapter explores what happens to CG if we adopt the idea that interpretation needs to happen in ordered domains, especially if we add to the type formation rules a construction for the opposite order. Doing this gives a Context Lemma in a straightforward way, and the Context Lemma is tantamount to the soundness of the internalized polarity marking scheme. The types in the internalized scheme are more plentiful than in ordinary CG, and this is a source of issues as well as possibilities. The issues concern how the new typings are to be generated, and how they are to interact with each other in proof-theoretic settings. There certainly is more to do on this matter, and this chapter has only provided the groundwork. With some additional work, one could easily imagine that the internalized scheme will be an important contribution to the research agenda of natural logic.
Bibliography


[37] Vardi, M. Y., and Wolper, P. Automata-theoretic techniques for modal


[40] Ziegler, S. Some completeness results in syllogistic logic. Student Reports, Research Experience for Undergraduates Program, Indiana University, 2008.

Index

atom
  binary, 49
  complemented, 30
  unary, 5, 49

boolean algebra, 24
  power set, 24
  syntactic, 25
Boolean modal logic, 51
  bottom ($\bot$), 53

categorical grammar, 109
  complete
    logical system, 7
    set of sentences, 55
    strongly, 22
    weakly, 22
  consistent, 16, 55

ex falso quodlibet, 16

filter, 25

inconsistent, 16, 55

literal, 13, 30, 49

logic
  Boolean modal, 51
  sentential, 21–27
logical system
  sentential, 23

model, 5, 50, 114

negation
  semantic, 52

orthoposet, 33
  morphism of, 34
  point of, 34

polarity
  negative, 121
  positive, 121

poset, 33

preorder, 8, 33
  flat, 111

function space, 111
  opposite, 112

proof tree, 6, 31, 53

RAA, 52

set terms, 49

sound, 7

syllogistic proof system, 59

syllogistic rule, 59

tautology, 22

ultrafilter, 25

valuation, 21