
LOGIC

A Modern Approach

# Beginning Deductive Logic, Advanced

via HyperSlate™ and HyperGrader™

Larry likes Lucy. Everyone likes anyone who likes at least one entity. Does everyone like Lucy? Prove it!

Selmer Bringsjord

Naveen Sundar Govindarajulu

Joshua Taylor

and camelid art by … KB Foushée

 Published in the United States by Motalen LLC.
www.motalen.com

ISBN 978-0-692-05943-2 (e-book, edition 011519)
ISBN 978-0-692-05944-9 (Slate™ software system, edition 012516)

# Preface

## 0.1 Student/Instructor: This is the Advanced Version!

Please note this immediately: The book you're currently reading, as indicated by its title, is an *advanced* version of beginning deductive logic (BDL) in the LOGIC: A MODERN APPROACH (LAMA; rhymes with "llama") paradigm.[1] College or university students lacking a demonstrable understanding of high-school mathematics in its full breadth (including therefore *discrete* mathematics) are advised to pursue their initial learning of deductive logic using the standard, non-advanced cousin of the present textbook. (Of course, alternatively, those without such understanding can secure it by suitable remedial coverage of high-school mathematics, and can then return here after remediation is complete and understanding is in place.) Needless to say, college-level logic instructors of classes with undergraduate students whose high-school training in math is insufficient are urged to use the standard, non-advanced relative of the present textbook. The standard textbook[2] presupposes exceedingly little on the part of learners who take it up, unfolds at a slower pace, and covers less content. The Slate software system, and Hypergrader (an online AI system that among other things automatically grades files created in Slate), can both be used for the teaching of beginning deductive logic in *both* advanced and non-advanced modes.

The phrase 'demonstrable understanding of high-school mathematics in its full breadth' is rather a mouthful, and stands in need of clarification. What do we mean? Let's first divide the phrase into two parts: *demonstrable understanding* and *high-school mathematics in its full breadth.* Now we can explain each part separately.

As to what's meant by the second part of the phrase, 'high-school mathematics in its full breadth,' that's easy: we simply refer to the branches of mathematics traditionally introduced in high school: viz., algebra, geometry, and trigonometry. Hopefully high school afforded coverage of elementary differential and integral calculus, but this isn't stricly required.[3] In addition, it would be a good thing for

[1] Abbreviating, the present book is affectionately known as LAMA–BDL**A**.

[2] LAMA-BDL

[3] It would nonetheless be a very good thing if the student embarking upon stufy of formal logic via LAMA-BDLA has digested elementary differential and integral calculus. The formal definition of a limit, which 99 times out of 100 is an early foundational concept in this calculus, turns out to involved sophisticated quantification, and thus any student that truly grasps this concept has a leg up.

students to have been taught, as is commendably done in some high schools,[4] some mathematical statistics and computer programming,[5] but neither of these is required either. In fact, the present book includes a brief introduction to logic programming.[6]

Now we come to the first part of our phrase: What is it to have, as we say, *demonstrable understanding* of high-school mathematics? By this is meant that the student is able to *prove* relevant theorems, for while the LAMA™ paradigm is based upon a number of pedagogical principles, first and foremost among them is what can be labelled the Driving Dictum:

> If you can't prove it, you don't get it.

In our experience, some, upon being urged to abide by the Driving Dictum, viscerally see it as unreasonably demanding. Given this, it's important to to know that if the cognitive science of reasoning, decision-making, and rationality/irrationality has disclosed anything, it has disclosed that life lived in accordance with the Driving Dictum wards off error. This can be immeidately seen when we note the intellectual travails of humans who shun the Driving Dictum. For example, consider this short arithmetic problem:[7]

> A bat and a ball cost $1.10 in total. The bat costs $1 more than the ball. How much does the ball cost?

The vast majority of those not living by the Driving Dictum respond confidently with: "10 cents." Alas, this is incorrect. And that this response is wrong can be quickly revealed if the following request is issued, and the attempt to accede to it is made': Prove it! The reason why this request is so instructive is that when one attemps to prove that under the supposition that "10 cents" is the cost of the ball, here's what happens when one attempts to articulate a valid proof:

> **Proof**: Suppose that the cost of the ball is 10¢, as suggested. The objective is to prove that under this supposition, the cost of the bat and ball together is

[4] To its great detriment as a nation, and to the detriment of many young minds who deserve better, the notion that high-school graduates needn't have more than a smidgeon of mathematics (proposed e.g. in Baker 2013) is now a cancer that has taken firm root in many U.S. schools and households. Perhaps equally pedagogically pernicious is the idea that some college-bound students in the States should be encouraged to follow a route to earning college/university mathematics credit that allows them to dodge algebra, geometry, trigonometry, and calculus. The most prominent such route is one in which college-bound students simply get some elementary coverage of statistics. Unfortunately, one noteworthy consequence of this route is that it produces college graduates constitutionally unable to explain why it's not impossible that an arrow shot by an archer moves (!); see (Bringsjord & Bringsjord 2014).

[5] Notice that we refer here to *mathematical* statistics. Plenty of classes and books devoted to teaching statistics are unfortunately devoid of any mathematical rigor to speak of. As to computer science, we say only that having had some of it can't hurt. Unfortunately, K–12 coverage of computer science in the U.S. is almost invariably informal, and as such doesn't request of students that they prove anything. Given the Driving Dictum presented just below, this means that computer science in the U.S. is taught in a way that fails to guarantee genuine understanding on the part of students. This situation is made all the worse by the fact that the Advanced Placement computer science curriculum and exam is not based on either of the programming paradigms most naturally associated with rigorous proof (functional programming and logic programming), but rather on the object-oriented paradigm.

[6] The first part of this introduction is provided in §2.8.

[7] This unassuming but pregnant problem is due to Kahneman & Frederick (2002).

> $1.10. Now, the ball must cost $1 more than the ball, and one dollar more than 10 cents is $1.10, so that is the cost of the ball. To get the cost of both bat and ball together we add $1.10 + 10¢, which yields $1.20. OOPS!!!

Returning to the realm of high-school mathematics, if we follow the Driving Dictum, it's easy to ascertain whether or not a student (or, for that matter, a machine/AI does[8]) really understands core concepts in, for instance, algebra. Here's a simple example, a problem early in the textbook *Algebra 2* by Charles et al. (2012) that calls for a proof: Prove that the sum of two consecutive positive integers is invariably an odd integer. We assume that our readers can quickly accomplish this.[9] There are many, many requests for proofs in this Algebra-2 textbook (and for that matter in all of its counterparts from other publishers), after this initial, easy one. Here are a few examples, each of which you should be able to establish after some work:

- Prove that a triangle with vertices (3, 5), (-2, 6), and (1, 3) is a right triangle. (page 198)
- Prove that $n^3 - n$ is divisible by 3 for all positive integer values of $n$. (page 516)
- Prove this factoring formula for the sum of cubes: $a^3 + b^3 = (a + b)(a^2 - ab + b^2)$. (page 704)

The truth of the matter is that high-school mathematics, if one takes it seriously and excels, places the student in a position from which they can in turn excel in beginning deductive logic at the college level, even when introduced in the advanced and comprehensive manner of the present book. The fundamental reason, put starkly, is that this level of mathematics is filled with, indeed is directly based upon, deductive logic, and includes plenty of problems that are solved only by providing valid proofs. This can be seen not only by visiting high-school textbooks for Algebra 2, but also by perusing first-rate textbooks for high-school geometry, a very nice example being *Geometry: Common Core* by Bass & Johnson (2012), whose second chapter is entitled "Reasoning and Proof," with sub-sections mastery of which are ideal for learning deductive logic via the present textbook and its associated software.[10]

---

[8] These days there's much veneration of so-called "machine learning," or just "ML," on the strength of the claim that computing machines who engage in ML learn sophisticated things (e.g. games of perfect information) with amazing speed. From the perspective of the Driving Dictum, such exuberance about ML is suspicious. For an account of what is called 'real learning,' and an argument for the proposition that ML-powered machines don't really learn anything at all, see (Bringsjord, Govindarajulu, Banerjee & Hummel 2018).

[9] Pedantic, but in the clear, crisp style that will become our bread and butter as we progress:

> **Proof**: Let $n$ and $n+1 = m$ be two arbitrary consecutive positive integers (i.e., $n, n+1 \in \mathbb{Z}^+$). We know from basic arithmetic that $n + (n+1) = 2n+1$. Obviously $2n$ is even, since it can be divided without remainder by 2 (to leave $n$). But then $2n+1$ must be odd. **QED**

For a somewhat more meaningful diagnostic challenge along the same lines: Prove that the sum of the cubes of any two consecutive positive integers is odd.

[10] The sub-sections are, and we reproduce verbatim: 2.1 Patterns and Inductive Reasoning; 2.2 Conditional Statements, 2.3 Biconditionals and Definitions, 2.4 Deductive Definitions, 2.5 Reasoning in Algebra and Geometry, and finally 2.6 Proving Angles Congruent. Any high-school student who has mastered this chapter already knows a fair amount of what is covered in Chapter 2 in the present book.

We end this section by using a convenient deductive inference schema almost always covered in high-school mathematics. The schema is known as **transposition** or **contraposition**, and allows one to deduce from a conditional statement `if` $\phi$ `then` $\psi$, where $\phi$ and $\psi$ are themselves statements, this new conditional statement, the **contrapositive** of the original one: `if not` $\psi$ `then not` $\phi$. With this schema, then, we deduce from the Driving Dictum this statement:

If you do get it, you can prove it.

## 0.2 So, What's Our "Modern" Approach?

It's likely you're able to read this sentence because you (or a benefactor who operated on your behalf) parted with some hard-earned money. Courtesy of this transaction, you now have in your possession a book-software combination (LAMA-BDLA and HyperSlate™) that, used in conjunction with the AI grading system HyperGrader™, will introduce you to modern, formal, deductive logic in what seems to us to be a particularly, if not a singularly, effective way. This trio has been well over a decade in the making, and the construction — since new logics and applications thereof, in both cases from us and our collaborators, continue to arrive on the scene — is still steadfastly underway. Since we stand on the shoulders of many who came before us, there is a real sense in which the trio in question has been in the making for well over two millennia. Ancient thinkers may have been greatly mistaken about the *physical* world,[11] but many of them were spot-on right about many aspects of deductive logic. Aristotle, a "mathematical spirit" who worked his wonders over 20 centuries ago, is very much alive in the pages of the present book, and its corresponding software.[12] Of course, as we soon enough explain, the *specifics* of Aristotle's logic are certainly outmoded, and have been superseded by logics that do everything Aristotle's could, and a whole lot more.[13] The "whole lot more" first arrived on the scene, as far as we can tell, in the mind of the last great polymathic genius: Leibniz. We view Leibniz as the inventor of modern formal logic, and as the primogenitor of our approach to reasoning, which is to see this reasoning as graphical and diagrammatic in nature.[14]

[11] Case in point, Aristotle's attempt to solve Zeno's paradoxes of physical motion is entirely deficient. If interested, see (Bringsjord & Bringsjord 2014).

[12] The phrase 'mathematical spirits' is due to mathematician David Hilbert, who used it in the dramatic opening of his famous 1900 Paris lecture, delivered to direct future mathematical minds to such problems as the 23 grand ones he then proceeded to list. Hilbert was addressing the International Congress of Mathematicians. An online English version of Hilbert's address is available at http://aleph0.clarku.edu/~djoyce/hilbert/problems.html. The logician Kurt Gödel, some of whose seminal work we shall have occasion to study later, was soon able to make stunning progress on some of Hilbert's challenges.

[13] We thus for good reason regard study of Aristotle's logic to be otiose for non-historians, and hence refrain from covering Aristotle's logic in the present textbook. And excellent, short overview is provided in (Smith 2017). This overview includes presentation of a roadmap that will enable interested readers to study all the parts of Aristotle's writings that deal with logic. More specifically, all that Smith (2017) deals with can be found in (McKeon 1941).

[14] The classic starting point for the study of Leibniz and his contribution to modern formal logic is (Lewis 1960). However, this starting point, we now know, massively understates the degree to which

But while our modern approach to logic is indebted to the past, it in no way asks you, a student of logic in the new millennium, to return to the past. Like you, we find it empowering to use a modern mobile smartphone, and though such a device owes a debt to $19^{th}$-century telegraph technology, it makes little sense to tap out a message to your friend in Morse code. Those days are gone. And likewise gone are the days of memorizing Aristotle's syllogisms, since they are swallowed up and exceeded by even a tiny part of modern (formal) deductive logic.

Why didn't we just take the rather less strenuous route of circulating and posting a recommendation on the Internet to the effect that all beginning students of formal deductive logic (you included) ought to simply purchase one of the many *other* logic textbook/courseware pairs on the market? We cheerfully own up to the answer, one that you you will no doubt have anticipated: Because we believe you are better off learning beginning deductive **l**ogic under **a m**odern **a**pproach (i.e., abbreviating, under LAMA™; pronounced to rhyme with the name of the fascinating sure-footed animal), with the software system known as 'HyperSlate™,' which you now (or soon will) have, and with the third member of the trio: the online AI system able to diagnose and guide student work: HyperGrader™. We firmly believe that this approach is pedagogically superior to all the alternatives. In the book you're viewing, our approach, and HyperSlate™ and HyperGrader™, are specifically harnessed to teach you beginning (formal) deductive logic. There is plenty of important logic that isn't deductive, and of course plenty of deductive logic that's well beyond the introductory level (and plenty of informal stuff that's called "logic"; in fact the phrase 'informal logic' is sometimes used), but basic (formal) deductive logic is the absolute cornerstone of all rational thought on our planet (and for that matter on any others that might have intelligent life!) — yesterday, today, and tomorrow. Take it away, and in one fell swoop you sweep away the formal (= exact) sciences, computer science (composed, so the not-inaccurate slogan goes, of logic + electrical engineering), the physical sciences (which to the extent that they are sound, are based in the exact sciences), rigorous philosophy, and more.[15] Some of the additional things that would be destroyed by the absence of deductive formal logic are exceedingly practical. For instance, since the only way we have, and will ever have, to establish that software is correct and will behave in the future as we wish it to, is to deduce that the software in question is correct using formal logic, the ubiquity and centrality of software in human life confirms the importance of what you are about to learn. While at present our daily lives are only sprinkled with the appearance of autonomous AIs/robots, the time is fast coming when our existence will be intertwined with, and increasingly dependent upon, such artificial agents, and we will want to be quite certain that these agents will behave correctly.

Leibniz invented key aspects of modern formal logic. As detailed more recently by Lenzen (2004), Leibniz anticipated by approximately one-and-a-half centuries the propositional calculus (!), the inventor of which had been been presumed to be Boole (a presumption made courtesy of Boole 2010). (Boole's classic book is available free from http://www.gutenberg.org.) As Lenzen also points out, Leibniz also anticipated another part of modern formal deductive logic: *modal* logic. Modal logic is introduced in the present book (Chapter 5).

[15] You would also sweep away rigorous economics, since without formal logic we lose real analysis, game theory, decision theory, and so on, but we have found that our readers are profoundly mixed on whether or not it would be bad to lose economics.

But since the behavior of these agents will hinge in turn on the correctness of the computer programs that regulate them, and since, again, such correctness can be established only via the use of formal, deductive logic, our lives will in a very real sense rest in the hands of logic. If no one has told you this before, that is most unfortunate, but now you know. Good thing you're reading what you're reading.[16]

We do gladly concede that there is much value in *in*formal logic (and indeed Slate has been invented, refined, and implemented to support humans in doing informal reasoning), but the power of informal logic is in our view enjoyed in earnest only by one who *first* understands the formal, rigorous machinery that gives the informal level the matchless power it has. Learning informal logic before learning the basics of formal logic is by our lights like learning to count before learning such arithmetical facts as that (a) 3 is greater than 2 and that (b) 2 is greater than 1: Perhaps a child can *in some sense* count by mouthing "1, 2, 3" without a prior understanding of and implicit assent to (a) and (b), but on the face of it such a pedagogical progression is backwards. The child here is just making sounds without real understanding. Likewise, in the realm of logic, the student of informal logic shouldn't be taught to mouth assent to such things as that from "All men are mortal" and "Matt is man" it follows that "Matt is mortal" without understanding that this is true because the *underlying formal pattern*

$$\frac{\begin{array}{c}\text{All } As \text{ are } Bs.\\ e \text{ is an } A.\end{array}}{e \text{ is a } B.}$$

is valid. This particular formal structure was seen to be at the heart of human reasoning and rationality by Aristotle over (to say it again) two millennia ago. What makes the pattern a *formal* one, and therefore serves to classify the pattern as part of formal logic? You doubtless know the answer: because no matter what you substitute in for the collections $A$ and $B$ and the entity $e$, the conclusion will indeed follow from the two premises. Unless the student comes to understand the domain-independence of formal logic, no real learning of formal logic has happened. This implies that those educational games and activities that seek to foster learning of logic and mathematics by "concretizing" problems and problem-solving, in the absence of cultivating abstract, domain-independent capability, are at best profoundly incomplete and misleading, and profoundly destructive at worst. A five-year-old who is told such things as that the small yellow wooden circle can go either on this square or that square (where e.g. the squares are cells in a grid that can either be filled or vacant), and then dutifully places the circle in one of the two allowed locations, doesn't need to have a genuine and general understanding of Boolean disjunction, which applies as nicely to simple games as it does to advanced mathematical physics. Without such understanding, our five-year-old really doesn't know what he's doing. And worse, if the human avoids sustained training in formal logic, he or she can fail to do any better at 55 than at five.[17]

[16] For further reading, see (Bringsjord 2015*b*, Govindarajulu & Bringsjord 2015, Arkoudas & Bringsjord 2007).

[17] Beware, then, of educational games to *purport* to cultivate, in those that play them, skill at formal reasoning.

Summing up: Both because of the nature of our approach and courseware, and the importance of what you will be studying, you (and perhaps your benefactor) have made a wise investment.

But why *specifically* is the LAMA™ paradigm, and specifically HyperSlate™ and HyperGrader™, better? We have already hinted at our answer above, and in the title of the present book: because this pair, falling squarely in the pardigm in question, takes a *modern* approach to teaching logic. But what does *that* mean? The hallmarks of what we call a "modern" approach, in addition to the Driving Dictum introduced above (again: "If you can't prove it, you don't get it"), include the following ten properties:

1. *Use and Learning of Automated Reasoning Technology.*
2. *Coverage of Large Space of Modern Logics, and Three Families Thereof.*
3. *Use of Flexible, Graphical Workspaces for Human-Computer Collaboration.*
4. *Learning of Logic via Playing **Truly** Challenging Games on Planet Earth — Rather Than The Sort of Simple Games Traditionally Targeted by AI.*
5. *A Grounding in the Cognitive Science of Human Reasoning and Decision-Making.*
6. *LAMA™, HyperSlate™, and HyperGrader™ Are Based on Deep Mathematical Theories of Cognition and Computation.*
7. *Extending Formal Deductive Logic Beyond the Merely Symbolic/Linguistic to the Pictorial/Diagrammatic.*
8. *Extending Formal Deductive Logic to Bridge to Uncertainty/Probability in **Inductive** Logic.*
9. *Seamless Innteroperability and Integration with a Corresponding Type of Logic-based Computer Programming.*[18]
10. *Logic as the Emerging Foundation for Rigorous Science.*

We now unpack each of these ten properties in turn, albeit very briefly in each case.

### 0.2.1 The Hallmarks of Our Modern Approach, Unpacked

1. *Use and Learning of Automated Reasoning Technology.* Today, humanity has computer programs able to do things which in the past were the sole province of human beings. It has probably not escaped your notice, for example, that the best checkers player, chess players, and Go players on our planet are no longer human but rather are computers; and perhaps you know that recently even in a game like *Jeopardy!*, which puts a premium on the use of human language, computers can perform at an impressive level.[19] One of the things that computing machines can now do that not all that long

[18] The type of computer programming is a generaliztion of what is called *logic programming*: viz. **pure general logic programming**, or just 'PGLP' for short.

[19] IBM's Watson system beat the best human *Jeopardy!* players; the system is explained at the "Scientific-American" level in (Ferrucci et al. 2010). For commentary on Watson in connection with the AI HAL 9000 in Kubrick's *2001*, commentary offered *before* the actual competition that Watson won, see (Bringsjord, Clark & Taylor 2010).

ago used to require rarefied human cognition is proving theorems. Machines can't yet match humans in the proving of *robust* theorems (in fact, they aren't even close; we'll get to this issue later, when you know a bit more formal logic), but nonetheless it's undeniable that machines can now routinely prove theorems that not all that long ago required significant human ingenuity.[20] HyperSlate™ and HyperGrader™ have been engineered, and this textbook written, to leverage this progress in support of student learning.[21] In short, and put bluntly: No student these days should learn logic without specifically learning how to collaborate directly and deeply with intelligent machines able *themselves* to reason deductively. Put in analogical terms: the point is little different than the fact that no pilot should learn how to fly a plane without learning specifically how to collaborate with smart software in doing so (e.g. the software that controls autopilot functionality), or that no engineer should be trained without learning how to use a calculator and beyond. One of the most amazing (and by our lights, distressing) facts about American post-secondary education is that, even today, as the field of artificial intelligence (AI) is thriving and marching steadily upwards, many students at high-schools, colleges, and universities in the U.S. are learning logic using only paper and pencil.

2. *Focus on Modern Logics, and Three Families Thereof.* Lots of logic textbooks spend considerable time on logics like Aristotle's **Theory of the Syllogism**, which we alluded to earlier. This simple logic debuted, as has been noted, rather long ago. It's always good to know one's history, but the fact is that logic has advanced rather far in the 2000-plus years since Aristotle's small (but seminal) system was presented. In fact, as Glymour (1992) has elegantly explained, Aristotle's primitive logic was insufficiently expressive to rigorously explain why even ancient Euclid's reasoning (in showing such propositions as that the sum of the interior angles of a standard triangle is 180 degrees) is so compelling. To hammer home the point again: The fact is, when students start learning physics at the college level today, they don't go back and learn what the ancients were doing. The reason is simple: the work of the ancients has been forever superseded by a host of innovations. This book includes coverage of modern logical systems that have superseded old logics like those presented by Aristotle and others. More specifically, we cover modern **classical extensional logic**, invented in the 20th century to represent and rigorize mathematical reasoning (and now used to do lots of real-world things; we'll get to that); and we also cover **basic intensional logic**, which also entered the picture in the 20th century, but for different purposes: to systematize the concepts of possibility, necessity, knowledge, belief, obligation, and so on, and to systematically represent what is meant by sentences in **natural languages** like English.[22] These two families of logical systems are unified by the use of HyperSlate™, which allows the user to select a particular logical system when starting a new workspace. As far as we know, HyperSlate™ is the only software in existence that allows a human to pilot an intelligent computer which provides, at once, access to many different logics at the same time.[23]

[20] Luger (2008) makes the point that automated theorem proving is a crucial component of the sort of human-level AI that many are after, and provides a new overview of the earliest days of automated theorem proving.

[21] LAMA™ is also attuned to progress in what is known as *model finding*, which we see as included within automated reasoning, at least for the family of extensional logics.

[22] Natural languages stand in contrast to **formal languages**. The latter, examples of which you will soon get to know quite well, have a fully specified alphabet and grammar. A programming language is for instance a formal language.

[23] The deductive logics you will be learning each fall into one of two main families just mentioned:

We said there are *three* families of logics introduced herein. What's the third? It's **inductive logic**, and while the the present version of this textbook goes very light on this third family, the some key information is provided to the student. The reason for the light coverage is that inductive logic is non-deductive, and this textbook, as its title suggests, is intended to serve mostly as an introduction to beginning deductive logic. Nonetheless, it's important for the student of elementary deductive logic to understand some inductive logic, if for no other reason than to see that beginning inductive logic is quite a different beast.[24]

3. *Use of Flexible, Graphical Workspaces for Human-Computer Collaboration.* Real humans doing real deductive reasoning in the real world don't simply list out, line-by-line, sequences of formulas when doing so. Yet if you look at standard approaches to teaching deductive logic, and the software touted in these approaches, you would think that real-world deductive reasoning is all and always nothing more than a one-dimensional list of formulas, with a bit of indentation. No; the real world is richer than this, beyond belief. If you look at the notebooks, scratchpads, blackboards, whiteboards, and even computer screens of highly effective human reasoners, you rarely find the kind of rigid text-only linear lists of formulas that have been, and unaccountably continue to be, the hallmark of most if not all other logic books and courseware. Slate is a step in the direction of our vision for a cognitively plausible workspace that is tremendously open-ended and genuinely intelligent and collaborative. In Slate, formal proofs are directed hypergraphs whose vertices contain sentences and, often, other relevant information. (Don't worry in the least if you don't presently know what a hypergraph is. All will be gradually explained in due course!) At the same time, Slate can be used to constuct trees. This means for instance that instead of the traditional harsh division seen in standard approaches between a linear step-by-step proof on the one hand, and entirely different formats for semantics (e.g., truth tables) on the other, Slate's graphs can be used to express proofs *and* express semantics in the form of trees. In addition, in Slate, formal interpretations or models can also be constructed in graphical form.

4. *Learning of Logic via Playing **Truly** Challenging Games — Rather Than the Sort of Simple Games Traditionally Targeted by AI.* The LAMA™ paradigm includes a commitment to learning formal logic in part by playing games; but not easy (or, for that matter, so-called "serious") games, and not games on which mindless trial-and-error and plug-and-chug can secure success. Educational games under the LAMA™ approach must be sufficiently hard and, in keeping with the Driving Dictum, success in LAMA™ games must require the gamer to demonstrate mastery by justifying the

**extensional** or **intensional**. Within the first family, coverage is provided first of logics that have no quantification (viz., **propositional calculus** and **pure predicate calculus**), and then of **full first-order logic** (with subsequent coverage of **second-order logic** and beyond: see Chap. 4). Within the family of intensional logics, this book introduces **propositional modal logic**, **propositional deontic logic**, **propositional epistemic logic**, and finally **quantified modal logic**. All of these logics are finitary in nature: they don't permit expressions that are infinitely long. Yet some of the most interesting and powerful logics are infinitary ones. E.g., the logic $\mathscr{L}_{\omega_1\omega}$ allows conjunctions (such as '1 is greater than zero and 2 is greater than zero and ...'), and also allows proofs that are infinitely long. Summing up, then, we can picture the Universe of Logics as shown in Figure 2.

[24] Very briefly, deductive reasoning, when valid, can't possibly be called into question, because it partakes of inference schemata, such as *modus ponens*, that are indubitable and invulnerable.[25] This schema says that if we know that $\phi$ implies $\psi$, and we have $\phi$, it absolutely, positively *must* be the case that $\psi$. Inductive reasoning, in stark contrast, includes such inference schemata as that if it's overwhelmingly likely that if $\phi$ holds so does $\psi$, and $\phi$ does in fact hold, it follows (inductively!) that $\psi$ holds. Chapter 9 is where a glimpse of inductive logic is given.

correct answers given, or moves made.[26] LAMA™ games are provided separately from the present LAMA-BDLA textbook, and separate as well from HyperSlate™ and online intelligent software that complements HyperSlate™ (e.g., Hypergrader™). It is recommended that the first game students play in the LAMA™ series of games is Catabot Rescue 1, Beginner.[27] For more on genuinely challenging reasoning games, see (Govindarajulu 2013).

5. *A Grounding in the Cognitive Science of Human Reasoning and Decision-Making.* LAMA™ is based on what decades of research in cognitive science and artificial intelligence has informed us about human reasoning and decision-making. Piaget believed that neurobiologically normal human beings provided with a general and standard K-through-12 education (and with sufficient nutrition, shelter, and so on) would effortlessly become competent deductive reasoners and problem-solvers at a level beyond full first-order logic (e.g., see Inhelder & Piaget 1958).[28] Unfortunately, Piaget was a little too sanguine. While he carried out numerous experiments showing that many young people do indeed develop an impressive capacity for *some* deductive reasoning, even Piaget himself conceded toward the end of his career that his subjects were not representative of the general population. LAMA™ is based on the *neo*-Piagetian view that *if* students are appropriately trained (courtesy of books and software of the sort that you now own and/or have access to!), they can indeed become first-rate deductive reasoners. There is a lot of evidence supporting this view, including recent neuroscientific evidence.[29] As to theories of reasoning directly at odds with the LAMA™ paradigm, there are some. The best-known of these is so-called *mental models theory* (MMT), originally introduced by Johnson-Laird (1983). A core tenet of MMT is that the human mind isn't fundamentally equipped with the capacity to reason deductively in a manner aligned with classical deduction as seen in formal logic and mathematics. Johnson-Laird and colleagues are quite aggressive and open about their disdain for formal logic (e.g. see Khemlani, Byrne & Johnson-Laird 2018). For work in the cognitive science of reasoning that accords with the LAMA™ paradigm, and hence is at odds

[26] Some readers may be familiar with logic puzzles in which a story that frames the problem is first given, then a set of clues, and the challenge is to e.g. pin down who did what when with what. Before the advent of apps on smartphones and tablets, these puzzles were seen almost exclusively in magazine-style hard-copy form. These sorts of puzzles have also been a mainstay in certain standardized exams, e.g. the GRE and LSAT. Those who learn sufficient formal logic thereby cultivate a capacity to excel on these problems (Bringsjord & Bringsjord n.d., Bringsjord 2001), but since the customary presentation of these problems lacks a demand that the supplied answer be justified by a proof, the problems in question, in the LAMA™ paradigm, are far from a pedagogically perfect fit. Readers/students who enjoy such puzzles are therefore encouraged to try to prove that their moves and answers are correct, using HyperSlate™.

[27] For more information, visit http://www.catabotrescue.com. Human prowess at Catabot Rescue, which can become quite remarkable, has a number of interesting implications for education and cognitive science; for more on this, see (Govindarajulu & Bringsjord 2016).

[28] Piaget held that humans develop cognitively through a serious of stages. In the fourth stage, human cognition is distinguished by a use of formal deductive logic in the third logical system we will be studying herein: namely, full first-order logic. See note 23.

[29] For an account of, and evidence in favor of, neo-Piagetianism, see (Bringsjord, Bringsjord & Noel 1998, Rinella, Bringsjord & Yang 2001, Bringsjord & Yang 2003). For neuroscientific evidence, see (Mukherjee 2009). On the other hand, there is *also* a mountain of evidence that if students *don't* receive suitable training in formal logic, they often rely on forms of inference that are invalid. This is of course entirely unsurprising. Without effective mathematical education of the right sort, ostensibly bright humans naturally believe such things as that when an arrow is in motion after being shot, it occupies a series of discrete places at each of the discrete moments composing the interval of time consumed by its trajectory. But such a belief implies that motion is impossible. Given the differential and integral calculus, we know this view is silly, and indeed self-contradictory (for a full discussion, see Bringsjord & Bringsjord 2014).

with MMT, the reader can consult the truly excellent work of Lance Rips (1989, 1994).[30] The LAMA™ paradigm in addition reflects the theoretical position, in the cognitive science of comparative reasoning and decision-making, that reasoning and decision-making at the human level is qualitatively distinct from such activity that is merely at the animal level. The essence of the mastery of formal logic is command of abstract structures (e.g., the inference pattern alluded to in the previous paragraph) that, as the empirical research in animal cognition shows, no animals can grasp. A clever canine can come to know that if the ice on a pond looks a certain way, walking on that ice is a bad idea, and so if Rover comes upon such a pond, it clicks for him that walking on it is a bad idea, and he remains on the firm ground of shore.[31] But the dog will never grasp the abstract inference schema operative here, which is none other than *modus ponens*. The LAMA™ paradigm, put simply, is one suitable for humans, not nonhuman animals. For more on the cognitive science of human versus animal reasoning and decision-making, see (Penn, Holyoak & Povinelli 2008), which is written in opposition to Darwin's (1997) exuberant but misplaced praise for the reasoning power of dogs.

6. *LAMA™, HyperSlate™, and HyperGrader™ Are Based on Deep Mathematical Theories of Cognition and Computation.* This isn't the place to explain the mathematics in question; we convey only the core idea here in rapid, intuitive fashion; sedulous readers are invited to carry out further investigation.[32] Our quick, intuitive explanation is based on Figure 3.

   This figure takes the agent and HyperSlate™ workspace, together, as capable of solving problems supplied from some outside source; we can refer to this source as the **environment**. The environment supplies problems to be solved by the duo of human and HyperSlate™; this collaboration, note is indicated in Figure 3. In the abstract, challenges are in the form of a question; this is also shown in the figure. The question is usually whether some given statement $p$, expressed as a formula $\phi_p$, can be proved from some given set of statements, expressed as a set $\Phi$ of formulae. When the given information is absent, we have a case where $\Phi$ is the empty (= null) set $\emptyset$. In that case the challenge is to prove $\phi_p$ without the benefit of some starting premises or axioms.

   Sometimes the human agent in this scheme can operate mechanically, with minimum ingenuity. This is the case when the human operating with HyperSlate™ can follow an algorithm. In the course of this book, students will acquire a number of algorithms

[30] The inquisitive reader may wonder why, specifically, Rips's work is in line with LAMA™, whereas MMT is inconsistent with this paradigm. The answer can be barbarically (but hopefully informatively!) encapsulated by way of reference to the proof pattern known as **indirect proof** (also known as **proof by contradition** and **reductio ad absurdum**), as follows: Rips's position is that human beings are capable of for instance solving reasoning problems that call for inference patterns such as this, which is at the very heart of the LAMA™ paradigm, while MMT maintains that the natural or "logically untrained" human mind can make no sense of this pattern whatsoever.

[31] This is an example given by Darwin himself, in an attempt, a failed one, alas, to show that the canine mind isn't qualitatively different that yours and ours; see (Darwin 1997).

[32] Where should they start? They will need to learn formal deductive logic one "notch" beyond a standard introductory logic course, but full assimilation of the present book, which as we have noted is advanced, will suffice. Assimilation of only the *non*-advanced introductory content under the LAMA™ paradigm would require a second course in formal logic, often called "intermediate" formal logic. Autodidacts can e.g. consult either or (better) both of (Boolos, Burgess & Jeffrey 2003) and (Ebbinghaus, Flum & Thomas 1994). Second, they will need a solid upper-undergraduate course in the theory of computation; a candidate book is (Lewis & Papadimitriou 1981). Third, they will need to become familiar with what are today called **Kolmogorov-Uspenskii machines**; one should begin with the *locus classicus*: (Kolmogorov & Uspenskii 1958).

which, when followed (or executed), allow significant progress to be made toward a solution to the problem confronting the student. This said, even beginning deductive logic includes challenges that cannot be met by *any* algorithm. (The discovery of this fact was momentous in the history of formal logic and computation, and is discussed later.) We'll get to such challenges soon enough; they are part of the fun of studying logic!

7. *Extending Formal Deductive Logic Beyond the Merely Symbolic/Linguistic to the Pictorial/Diagrammatic.* One of the most astounding things about the field of logic is that while it has been going strong for well over two millennia, only the tiniest speck of the discipline during that rather long stretch has been devoted to the study of reasoning over *pictorial* information. Everything other than this speck has been taken up in reasoning over representations that are *symbolic* (or, as it's sometimes said, *sentential* or *linguistic*). In LAMA™, logic is based on representations that are symbolic *and* pictorial, and in Chapter 8, such **heterogeneous logic** is introduced, and students are taught how to construct proofs that employ both diagrams and formulae.

8. *Extending Formal Deductive Logic to Intersect with Uncertainty/Probability.* Standard practice in the coverage of high-school mathematics is to try to distinguish between deductive and inductive reasoning.[33] Deduction is said to be a process of inferring "facts" from "facts" in such a way that the inferences are certain. In constrast, to inductively reason to some proposition from some information means that that proposition is only for instance *likely* given the information, even if the information is 100% true. This distinction is invariably made in such a way that an insuperable divide is formed between the two modes of reasoning, with logic pushed completely over to the deductive side of the house, and things like statistics and probability are left to tend to the other side. It's amazing but true that this unwise and inaccurate division is made as well by seemingly competent scientists and engineers. The problem with the division is simply that logic covers *both* deductive and inductive reasoning; it's just that the latter type of reasoning is covered not by standard deductive logic, but rather by *inductive* logic. LAMA™ is based on a treatment of logic that unifies deductive and inductive reasoning, and the kernel of the unification is explained in Chapter 9, in which some simple inductive logics in the LAMA™ style are presented. (Naturally enough, since the present book is an introduction to deductive logic, the treatment of inductive logic is minimal.) It is these inductive logics that allow the deductive-inductive unification to be accomplished.

9. *Seamless Innteroperability and Integration with a Corresponding Type of Logic-based Computer Programming.* Some of our readers will have some familiarity with computer programming; and some of these readers will specifically be acquainted with the fact that computer programming can be pursued within fundamentally different (indeed, in many ways radically different) paradigms. For example, in the **imperative** or **procedural** paradigm, which is unfortunately almost without exception what is taught to youth in the pre-secondary and secondary grades in technologized nations, the basic idea, put simply, is that a computer program consists in a series of instructions to do this, then that, then this, with the possibility for branching and iteration included in this step-by-step approach. For instance, a procedural program to compute the exponentiation function $exp(n,m) = n^m$ on positive integers might codify

[33] Note e.g. the first section heading '2.1 Patterns and Inductive Reasoning' given in the progression reproduced in footnote 10. In addition, online mini-lessons available from The Khan Academy attempt to distinguish between inductive versus deductive reasoning (e.g., see this tutorial).

the following basic do-this-then-that algorithm:[34]

A: Set the value of `result` to $n$.

B: If $m = 1$, PRINT `result` and STOP.

C: Set `counter` to 2.

D: Multiply `result` $\times n$; and set `result` to this product.

E: If `counter` $= m$, PRINT `result` and STOP; otherwise, increment `counter` by 1, and go to line D.

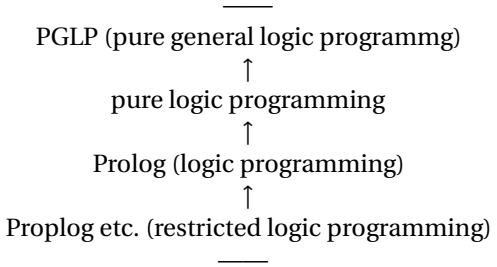
Let's call this algorithm '$A$,' and let's assume that some procedural program $P_A$ that regiments the algorithm $A$ is available. Now suppose that Johnny is wondering whether $2^5$ equals 32; perhaps he has received this question from his teacher: "Is $2^5$ equal to 32?". One possible move in response is for Johnny to follow $A$ by hand and see what he gets, and then give his answer to the teacher. Another possible move is to give as input 2 and 5 to $P_A$ and see what he gets as output. If he doesn't make a mistake in his manual implementation of $A$ on this input, and if the program $P_A$ doesn't malfunction (and he has made no mistake when suppplying the input to the program), Johnny will reply with a Yes to his teacher. However, what if the teacher is just a bit more demanding? Specifically, what if she says to Johnny: "Are you sure? How do you *know* that the correct answer is Yes?" Now Johnny has a problem. He will need to move beyond the procedural paradigm in order to answer the teacher's follow-up question. He will need to somehow *prove* that the affirmative answer is correct, and the only way he can do that is to turn to logic. The beauty and power of the approach to programming that LAM™ fits is that when the computer provides an answer on the strength of executing a **pure general logic program**, a proof that that answer is correct is automatically provided as well. This is so because the answer is produced via automated deduction, and the deduction to produced the answer is in fact a proof, one that is retained, checked, and verified. Such programs are at the heart of what Bringsjord calls **pure general logic programming**, or just PGLP for short.

PGLP is an extension of the programming paradigm known simply as 'logic programming.' Logic programming is a programming paradigm is one in which a program that captures the exponentiation function includes two components: namely, one consisting of a set $\Phi$ of declarative formulae that formally define the positive integers and the relevant arithmetic (addition, multiplication, exponentiation) on them; and, secondly, a process that searches for a proof of an given individual formula $\phi$ from $\Phi$. (In the case of Johnny, $\phi$ would of course express the informal statement that two raised to the fifth power is 32.) In the **pure logic programming** paradigm for computer programming, only the first of these components is something humans need to bother themselves with. Unfortunately, pure logic programming is something that's not achieved by the most popular "logic progrmmaing" language, Prolog. The reason is that in using Prolog, humans must in fact concern themselves with the second component. Beyond pure logic programming is what we have alluded to already, just above: again, **pure general logic programming**, or again, simply the acroynm 'PGLP.'. PGLP includes the purity that inheres in being able to blithely ignore the second component, *and* includes the formalisms that allow the human programming to generalize away from standard first-order logic and fragments thereof to *any* formal logic obeying certain conditions. Given that there are logic-programming languages that are less

[34] This is a very naïve algorithm, but for expository purposes is adequate.

expressive and hence less powerful even than Prolog (e.g., Proplog, which is has no provision for quantifiers and identify), we have the following basic progression $\mathscr{P}$ as what summarizes our discussion:

Figure 1: The Logicist Programming Progression $\mathscr{P}$




Now, what is the relationship between the LAMA™ paradigm, specifically including the present book, and the progression shown in Figure 1? The present book, LAMA-BDLA, will position those readers who assimilate it to move on directly to a deep understanding of the entire progression $\mathscr{P}$, and to corresponding skill at using logic-programming languages like Prolog. However, notice that we say that the reader will be in *position* to acquire deep understanding. We don't say that assimilation of LAMA-BDLA gives the student all that is needed. The reason is that the key theorems the underlie logic programming and Prolog aren't covered in the current version of LAMA-BDLA.[35]

10. *Logic as the Emerging Foundation for Rigorous Science.* Finally, our modern approach reflects the fact that formal logic is the *lingua franca* of all intellectual pursuits that strive for true rigor. As an example, physics, at the start of our new millennium, is being gradually expressed in formal logic; this is true, for instance, for quantum mechanics and special and general relativity (e.g. see Andréka, Madarász & Németi 2007, Andréka, Madarász, Németi & Székely 2011). A second example is one we have already alluded to above: The verification that software and software-based beings (e.g., robots) will behave as we desire them to behave is advancing on the strength of the only effective resource we have to meet such a challenge: formal logic. We believe that students of beginning deductive logic must be given a sense of the connection between formal logic and rigorous science and engineering. This textbook reflects this belief.

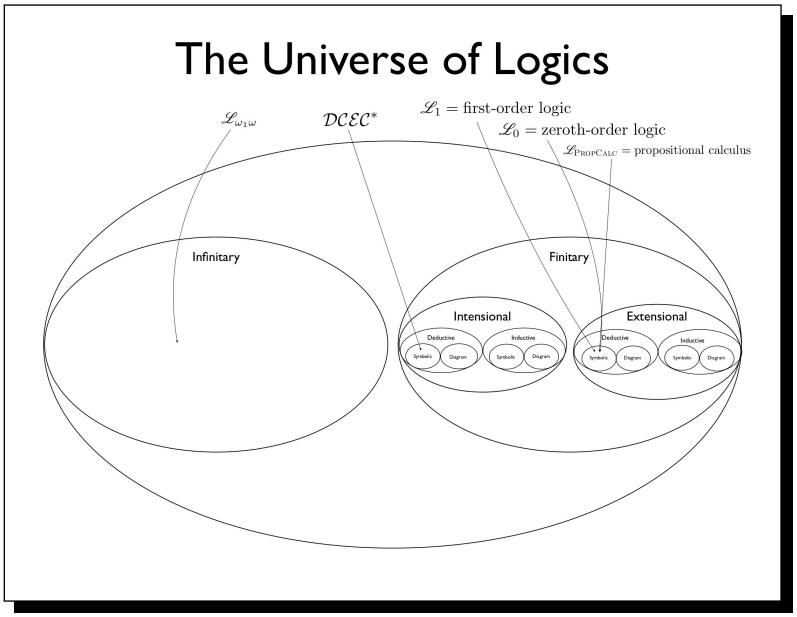
We present now a simple, anticipatory example that briefly shows the first five of the properties just enumerated, in action.

## 0.3 The Hallmarks in Action in an Example

To begin, take a look at Figure 4. This figure shows an argument, built in a **graphical workspace**, where that argument is expressed as a hypergraph, for a conclusion

[35] E.g., Herbrand's Theorem is fundamental to logic programming, and is not covered in the current version of LAMA-BDLA.

Figure 2: The Universe of Logics



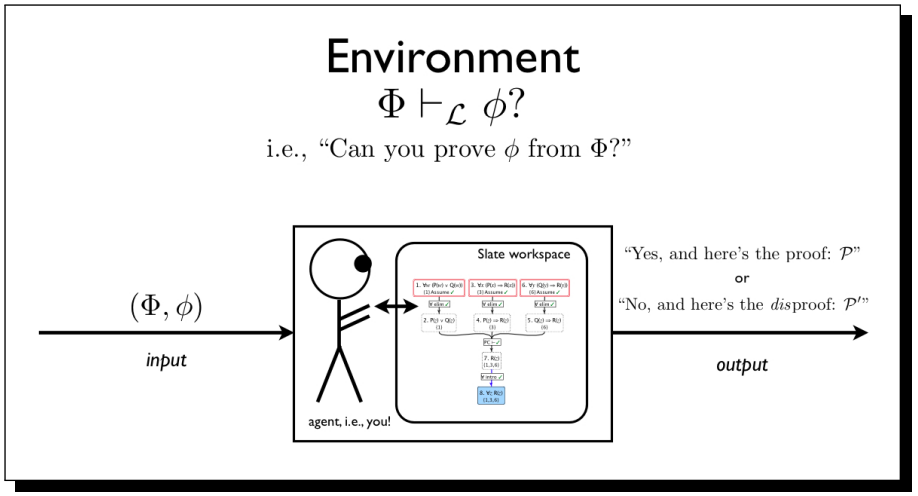

that we claim follows conclusively from the three shown premises. The graph has five nodes; three are labelled `PREMISE1`, `PREMISE2`, and `PREMISE3`; each of these, as you might guess, contains a premise. A fourth node is labelled `CONCLUSION`, and naturally enough contains the conclusion of the argument. A fifth node, rather more mysterious, contains this: `FOL` $\vdash$ X . We'll return to the meaning of this information momentarily. Study the graph a bit more. Do you agree that the conclusion, which expresses the proposition that Larry and Lucy like each other, can be correctly deduced from these premises?

Perhaps you aren't sure how to answer, because the phrase 'can be correctly deduced' is a bit unclear to you. (It *should* be at least *somewhat* unclear to you, since making it clear is one of the reasons for the remainder of the present book!) We can begin to fix that immediately. What we mean is this: Is it true that if the three premises are assumed to be true, then it *must* be true that Larry and Lucy like each other? This is just another way of asking whether mutual liking between Larry and Lucy can be correctly deduced from the three premises, because any time a statement can be correctly deduced from a set of premises, that statement absolutely, positively *must* be true if the premises are true.

We assume that your answer to this question is in the affirmative. (If you disagree, you haven't done enough thinking about the figure. So you might wish to return to it now before proceeding.[36])

[36] If, after sustained thinking about whether, given the premises here, Larry and Lucy must like each other, you fail to apprehend that indeed they must, you should probably find a fellow human fluent in

Figure 3: Pictorial Encapsulation of the Underlying Mathematics of LAMA and Slate




Now, let's see how the first five properties above, all hallmarks of LAMA, are put into action in connection with this example. Giving each of these five properties mnemonic labels, and keeping them in the order in which they were presented above, we can list them now economically as follows. After the shorthand list, we deploy the label in our discussion, bolding each occurrence of a label.

1. **automated reasoning**
2. **modern logics**
3. **graphical workspaces**
4. **cognitive science and AI**
5. **underlying mathematical theory**

First, it should come as no surprise to you that despite whatever other powers Slate has, it doesn't have the power to understand all of English as easily as a human can. Slate does understand many formal languages, and advanced versions of Slate do understand parts of English, but here in this introductory book we will be working with a version of Slate that doesn't automatically convert English into formal logic. More generally, we will be working with a version of Slate that doesn't automatically convert sentences in — what we have already called — **natural languages** (a class that includes not only English, but German, Norwegian, Japanese, etc.) to formulae in formal logic. Given this, for us in this book, part of the point of studying formal logic is to gain an ability to cast important parts of a natural language (in our case: English) into formal logic. And specific to the case at hand, we have therefore taken it upon ourselves to convert the English you see in Figure 4 to formal logic.

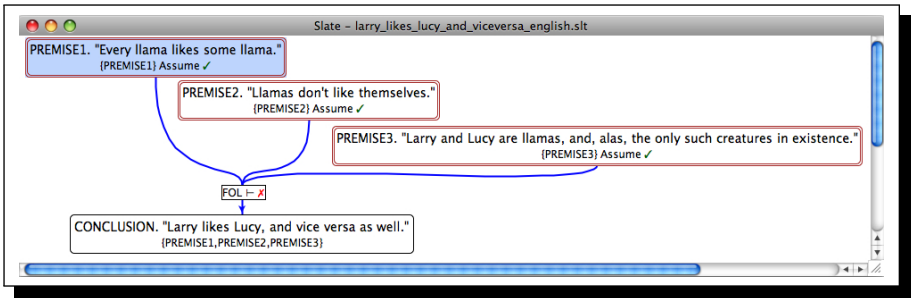
English, and in conversation with him or her make sure you understand in this case what the English means.

It turns out that though Aristotle was a smart chap, his logic (which you'll recall is The Theory of the Syllogism) is not up to the challenge of representing the argument about Larry and Lucy seen in Figure 4. But one of the **modern logics** that Slate has facility with, aforementioned **full first-order logic** (or, for short, **FOL**), has little problem with our specimen involving Larry and Lucy. If you now look again at Figure 4, you'll see that three arcs flow into that mysterious node that contains a red X immediately after this:

$$\text{FOL} \vdash$$

What on earth does this mean? Well, we just introduced you to the abbreviation 'FOL.' What you know at this point is that this is an abbreviated name for a particular logic (first-order logic). (Of course, you don't know anything significant *about* FOL, but that's fine; this is after all just the Preface.) What Slate as an **automated reasoner** is saying here via its red cross after FOL $\vdash$ is this: "Dear User, the argument that you have given me is not a valid deduction in the logic FOL. Sorry."

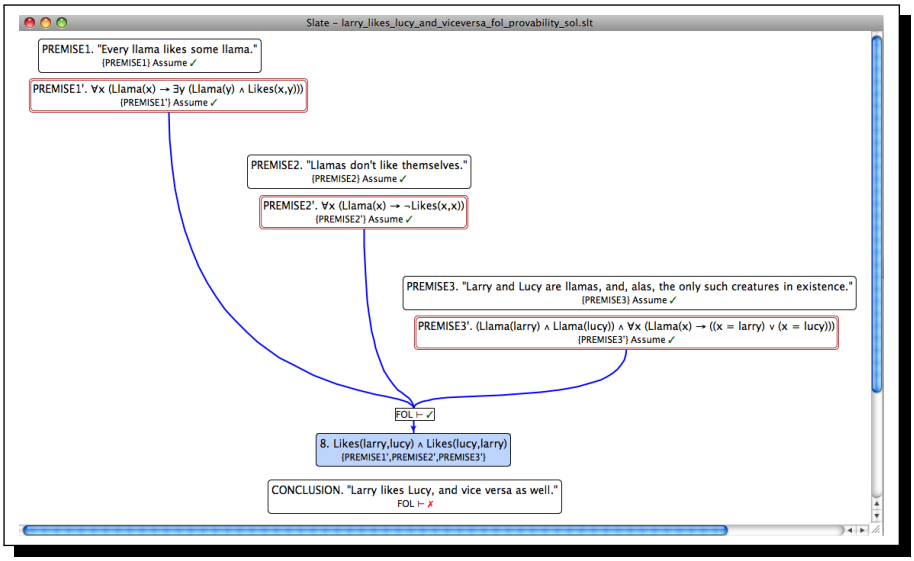
Figure 4: An Argument for Mutual Liking Between Larry and Lucy




However, when we translate the English into FOL for Slate, the system is smart enough to instantly declare that the conclusion can indeed be deduced from the three premises. This declaration is shown in Figure 5. Notice specifically that now Slate has presented an encouraging ✓; this signals that the proposed deduction is indeed valid in FOL. Of course, you don't yet know exactly what the funny symbols that you now see in the figure mean; that understanding will come. You will simply have to trust us at this early point that Slate is indeed right that the deduction is valid, on the strength of its **automated reasoning** power. Later, after you've learned some formal logic, you will yourself be able to prove that Slate is indeed correct. And after gaining a facility to verify that Slate is right when it issues an oracular declaration, you will come to see that Slate is pretty close to infallible. Slate has considerable intelligence, and as we said above, in our modern approach to learning logic, the student learns by interacting with a machine "being" that itself has sub-human but nonetheless considerable — and helpful — intelligence.

Notice that in the course of our discussion, the third property on the list above, that is **graphical workspaces**, has been used. Well, even if you haven't had any formal logic before, it's a good bet that you have been exposed to arguments and proofs that are both textual and linear in form, rather than what you have seen in

Figure 5: Slate Confirms Mutual Liking Between Larry and Lucy




Figures 4 and 5. In this old, linear, purely textual style, one sees things like this:

| | | | |
|---|---|---|---|
| | (1) | Llamas don't like themselves. | |
| | (2) | Larry is a llama. | |
| ∴ | (3) | Larry doesn't like himself. | (from (1) & (2) |

We have already said that when you inspect the notebooks of sophisticated human problem solvers, you rarely find anything as stark and one-dimensional as this kind of sterile progression. In general, line-by-line textual proofs reflect a slavish conformity to formats sometimes seen in computer science, where the old and venerable schemes for understanding the essence of computation do include such rigid ones at the **Turing machine**,[37] and where a finished computer program often does take exactly the form of a line-by-line progression of text. In addition, there is a longstanding tradition in the teaching of elementary, formal logic to ask students to proceed in rigid, line-by-line form, using indentation when necessary to indicate sub-proofs. But in learning logic in this day and age we shouldn't be limited by such straight-jackets. And fortunately, with Slate to help us, we aren't. In fact, the great news is that it can be proved that Slate is powerful enough to express any of the old-fashioned line-by-line proofs, should anyone want to work in the antiquated style. (It can also be proved that other graph-based formats for describing deduction can *also* be expressed in Slate.[38]) In the chapters to come, a

[37] Introduced by Turing (1937), and thereafter named in his honor. These machines are briefly described in Chapter 1.

[38] E.g., in §2.7.2.1 it's shown that so-called **resolution graphs**, which use resolution inference schemata, can be easily and accurately expressed as Slate hypergraphical proofs.

lot more will be said about, and done in, Slate's workspaces.

What about the remaining attributes in the quintet? Well, in our little example we have in fact seen in action, at least implicitly, the fourth and fifth attributes. The ability to translate the English statements contained within the nodes of the graph shown in Figure 4 into the formulas shown in the nodes of the graph shown in Figure 5 was of course required. You don't yet have this ability, but **cognitive science** tells us that that ability can indeed be cultivated, and of course the adventure you are about to take is designed to achieve exactly that cultivation. And what of the fifth attribute? Here you will have to trust us for the time being that anyone able to move from Figure 4 to Figure 5 in Slate has indeed conformed to our **deep underlying mathematical theory** of problem solving (which, as you'll recall, is at least intuitively encapsulated in Figure 3). That said, we do now briefly describe one algorithm that we executed in order to convert the English version of `PREMISE3` to some of the symbolization in `PREMISE3'`. We conceive of the algorithm as receiving **arguments** as input, and producing **values** as outputs. The particular algorithm in question receives an English statement $s$ of the form "*Name* is an *R*"; and it runs as follows.

1. Convert *Name* to a constant c in the logic in question. (In this case, the logic is FOL.)
2. Convert *R* to a relation symbol R in the logic in question. (In this case, the logic, again, is FOL.)
3. Assemble the formula R(c) from the results of 1. and 2.; and return this formula.

Of course, this is only one "building-block" algorithm that we ran in order to produce the symbolization shown in Figure 5 from the raw English shown in Figure 4, but this is after all only the Preface, and we thus rest content with your having but an intuitive sense of what's involved in moving from all of the English shown in the former figure to all of the logical notation shown in the latter one.

We anticipate this from the alert reader: "Ok, but what about the *ninth* property of a modern approach to deductive logic?" In what sense and in what way is the LAMA paradigm reflective of your claim that formal logic is the basis of even rigorous physical and natural science? Good question. At this point you will need to take us at our word, and trust us that logic is the steadily emerging universal language of scientific rigor. (Hence, if you perceive scientific rigor where no logic is used, you are misperceiving.) We will provide the evidence in due course. We will also provide in due course the material you must master to speak this universal language.

Finally, we humbly acknowledge in gratitude those devoted and brilliant intelligence analysts who, in the embryonic days of the LAMA paradigm and Slate, provided helpful comments and insights; and also the many RPI students who have likewise provided then, and indeed still provide — in connection with a more mature version of the LAMA paradigm and its technologies — now, valuable feedback.

Selmer Bringsjord
Naveen Sundar G.
Joshua Taylor

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Preliminaries

## 1.1 What is Logic?

We see two general ways to answer this question: One can be adventurous, and at least a bit opinionated, and provide a set of replies that far exceed the traditional, academic one; or one can stick with tradition and supply the customary slogan, and then flesh it out at least to some degree. We pursue both ways, in sequence: We first give the risky replies (§1.1.1, and immediately after that the traditional slogan (§**??**), which we unpack.

### 1.1.1 Adventurous Replies to the Question

As we give each adventurous reply in sequence to "What is logic?" we provide in each case a brief explanation of the reply:

1. "The science and engineering of reasoning." — so the not-unreasonable slogan (discussed in the next section, 1.1.2) goes.

   > *Explanation*: This is the traditional academic reply to the question; the reply is fleshed out in the next section (§1.1.2).

2. The key to becoming, and being, rational.

   > *Explanation*: It's reasonable to hold that rationality consists in the capacity to behave in accordance with the dictates of formal logic. This is in fact the position of the lead author of the present book, whose long-taught course Are Humans Rational? presents and defends this position over the course of a semester.[1] The only rigorous test of rationality that we are acquainted with, Stanovich, West & Tolpak's (2016) Comprehensive Assessment of Rational Thinking (CART), gives problems that specifically yield only to thinking that is based on formal logic. As an example of an problem like those on the CART used to assess facility with syllogistic reasoning, consider the following from (Johnson-Laird & Savary 1995):

[1] An opposing view, one that explicitly rejects formal logic as the basis for rational thinking and decision-making, is given in (Sosa 1999).

(i) All the Frenchmen in the restaurant are gourmets.
(ii) Some of the wine drinkers are gourmets.
(iii) Some of the Frenchmen are wine drinkers.

The challenge here is to answer this question: Does (iii) from the combination of (i) and (ii)? The irrational answer is "Yes"; the rational one is "No." The only way to get the rational answer in principled fashion (e.g., without simply guessing) is to use formal logic (in terms of the present book, if one understands Chapter 3 one will give the rational answer, and will indeed be able to *prove* that this is the right answer).

3. The only invincible subject there is.

   *Explanation*: Suppose someone proclaims "I have a refutation of logic!" If this is a rational person, and his/her hearers are rational, back will come the request to provide substantiation of the claim — but if this request is met, logic must have been used to do so. The proclamation is thus self-refuting.

4. The basis for the formal sciences (from mathematics to game theory to decision theory to probability calculi to axiomatic physics ...) — and hence the basis for disciplines based on the formal sciences (e.g., engineering, computer science).

   *Explanation*: Any number of ways can be taken to justify this answer to the question on a discipline-by-discipline basis, but all such ways would be suffocatingly laborious. Let's rest content here with simply pointing out that the formal sciences would evaporate without the persistent use of deductive inference patterns that are the province of (and indeed originated in) formal logic. These patterns are covered in the present book. One example is the inference rule/schema known as *modus tollens*, used throughout the formal sciences: viz.,

   $$\frac{\phi \rightarrow \psi, \quad \neg\psi}{\neg\phi}$$

   This inference schema tells us that whenever we have a statement that if $\phi$ then $\psi$, and we know that it's not the case that $\psi$, we can infer to what is below the horizontal line in this simple table, that is to $\neg\phi$.

5. The way of escape from shallow content and context to pure, immaterial, and immortal form and structure (which is why the exotic, imaginary, and seemingly non-sensical is so pedagogically useful).

   *Explanation*: To give just the kernel of the basic idea behind this answer, note that the inference rule of *modus tollens*, given immediately above, appears to be an immaterial, immortal (Who or what could possibly destroy it?) thing that can be apprehended by rational minds.

6. The most challenging subject there is.

   *Explanation*: Don't worry in the least! This is true only for *advanced* topics in logic, not for beginning deductive logic learned the way you'll be learning it.[2]

[2] Brave readers wanting an example they can study after the present book-software system is mastered are directed to the technique of forcing, introduced by Cohen (1963).

7. One of the chief differentiators between dogs and monkeys versus you (let alone bears and you); and mindless machines (like Deep Blue & Watson) versus you.

   *Explanation*: Nonhuman animals can't understand, let alone use, the abstract inference schemata such as *modus tollens* — and this is just the start of their logical limitations.

8. A key to riches.

   *Explanation*: We are here referring to *material* riches, and we assume that you find yourself in need of capital in order to generate riches. In that case, in order to get the capital (by legal means), you will need to persuade one or more persons by rational argument that you should be given the capital in question. Such argumentation is invariably and necessarily based upon the use of logic.

9. The key to divining the meaning of life (and other such big questions, such as whether God exists).

   *Explanation*: The only way to ascertain the answers to such weighty problems is surely to engage in systematic thought based upon formal logic — or so your authors hold.

10. The better way to program computers; and fundamentally the only way to reliably program computers.

    *Explanation*: Computer programming is pursued within certain paradigms, for instance *procedural*/*imperative*, *functional*, and *logic programming*. We recommend a pure, general form of the last of these.

11. One of two fundamental approaches to studying minds, and replicating/simulating minds in machines.

    *Explanation*: The field of artificial intelligence, AI, can accurately be partitioned, methodologically, into a camp that pursues the capture of thought by way of statistical/arithmetical processing (as e.g. in artificial neural networks), versus those who seek to capture thinking in formal logic. An overview of AI that makes this plain is (Bringsjord & Govindarajulu 2018).

12. The thing many creatures of fiction have mastered — have you?

    *Explanation*: This is a reference to fictional detectives, such as the first to appear on Earth, Edgar Allen Poe's C. Auguste Dupin, who cracks his first case in "The Murders in the Rue Morgue."

### 1.1.2 The Stuff, Standard Reply to the Question

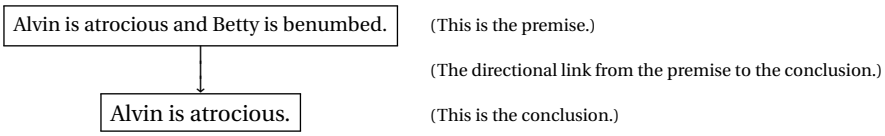
That is rather a deep and tricky question, as of course are relatives like: "What is physics?" and "What is economics?" and "What is art?". Perhaps one has to be either brave or foolhardy (or both!) to venture an answer to such a question, unless perhaps one has set aside enough time and space to craft at least a sustained essay,

or perhaps even a book. Well, we fancy ourselves brave; here goes; all done in but one sentence, and a short one at that:

Logic is the science and engineering of reasoning.

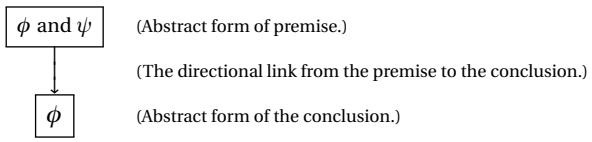
Maybe you don't find this answer enlightening. If so, that could be because you're not entirely sure what reasoning is. Well, by 'reasoning' we mean to refer to the process of linking statements or formulae (the latter are customarily used to represent the former; more on that momentarily) together in a chain that flows from premises to a conclusion, where it's at least hoped that the links are in accord with underlying, abstract, formal patterns of inference. It's common usage to label these chains 'arguments' — and here, as you might suspect, the term 'argument' means nothing like an emotionally charged dispute. Rather, an argument is a certain sort of structured object, no different in that regard than, say, a computer program. Here's a simple argument of the relevant kind, annotated in accordance with what we've just said:

**Argument 1**




Sometimes the directionally linked chain in reasoning is strong, and sometimes it's weak; there are also shades of strength (e.g., *indubitably* strong), but let's leave such gradations aside for now. In fact, we shall content ourselves here with but a rough-and-ready notion of what strength and weakness amounts to in connection with reasoning, by saying simply this: The strong case holds when, if the premises are believed by a thoroughly rational agent, then therefore so should the conclusion be believed by that agent.[3] Usually the better part of why, assuming a rational agent accepts the premises, she should accept the conclusion is that the links in the chain are indeed in accord with the right sort of patterns of inference. For example, **Argument 1** above is indubitably strong, since any rational agent who believes a statement having the underlying form of the premise in this case must believe a conclusion having the form in question. This can be made clear instantly, by looking at the underlying form of **Argument 1**, which is this:

**Abstract, Underlying Form of Argument 1**

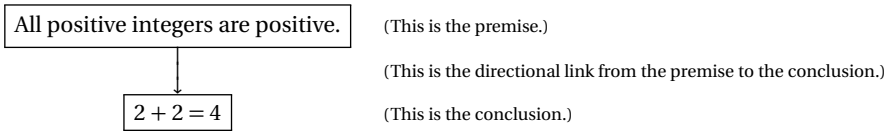



[3] Prudent is the author in the realm of logic and computation who follows the great Leibniz, who wrote in his immortal *Theodicy*: "Right reason is a linking together of truths, corrupt reason is mixed with prejudices and passions." *Theodicy* is available in electronic form for free, thanks to Project Gutenberg.

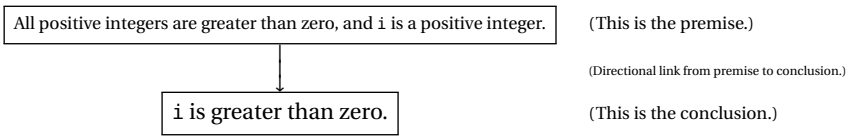
Here, the Greek letters are variables that can be instantiated to any particular statement whatsoever. Any argument that abides by this abstract form is thereby a very strong argument.

On the other hand, an argument is weak just in case a rational agent who believes the premises is not obligated to assent to the conclusion. The concept of obligation here used is obviously an intellectual one, not an ethical one. One if obligated as a student of (ordinary base-ten) arithmetic to believe that $0\neg = 1$, but this isn't an *ethical* obligation; it's an intellectual or cognitive one.

For example, the one-premise argument

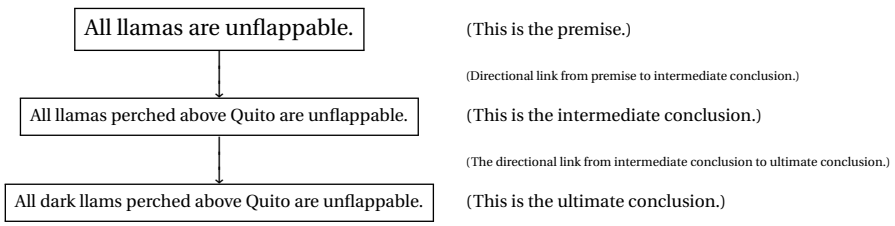



*is* an argument (since the premise is a statement linked to the conclusion), but it's a poor one, since a rational agent who believes the premise (and all such rational agents in fact do believe this) isn't thereby intellectually obligated to believe the conclusion. A perfectly rational agent will of course believe both the premise and the conclusion here, since both are not only true, but theorem, but this agent isn't obligated to believe the conclusion *because* she believes the premise. In contrast, the argument




is good argument, indeed a *very* good one. This time, if a rational agent ought to believe the premise, that agent is absolutely compelled, intellectually speaking, to believe that the integer `i` is greater than zero. We assume that you agree. In fact, since you are quite rational (you're reading our book, after all), we are confident that *you* find yourself obliged to believe the conclusion, on the assumption that the premise is true.

Sometimes reasoning involves multiple linked steps, not just single-link arguments. For instance, here's a piece of reasoning that involves two links and three statements, the first statement a premise, the second an intermediate conclusion, and the third and final statement the ultimate conclusion of the chain.



We said at the outset that logic is the science and engineering of reasoning. What is that science and engineering like? What does it involve? Well, note that so far in this chapter we have simply looked at examples of reasoning. Our specimens display that reasoning is the chaining from premise to conclusion. We have also allowed ourselves to speak of good and bad such chains, for the perspective of a thoroughly rational agent. All of this had been pretty vague. Getting rid of the vagueness in favor of rigor is what the science and engineering is all about, and the process of banishing the obscurity and putting precision in its place is going to take the pages that follow. It will suffice in the present chapter, which is devoted to setting out preliminaries before our investigation commences in earnest, to say just a bit more about the kind of reasoning we are especially concerned with.

As we've already said, we're specifically concerned in this introductory book with *deductive* reasoning. That means that the best of our chains will have the following rather amazing property, first mentioned in the Preface: if the premises are true, then the conclusion absolutely, positively *must* be true as well. In short, we shall be concerned, first and foremost, with **deductive arguments**, and especially with a particular sub-class of deductive arguments; namely, **proofs**. Logicians have a special fondness for *valid* arguments and proofs, and indeed much of our time together in these pages (and much of your time spent working with the accompanying Slate system) will be devoted to the hunt for such things.[4]

Please don't think for even a second that a focus on deductive arguments limits our investigation and your education. The reason is that a multitude of topics spring from our focus, and each of these topics will need to be addressed as we proceed. For example, what are the things linked together in deductive arguments? The answer, in a word, as we have already indicated, is: either **statements** or **formulas**. But what is a statement and what is a formula? The answers in turn, in short: A formula is a symbolic representation of a statement; and a statement is either a **declarative sentence** in some language like English, or it's a **declarative statement** like '$2+2=4$,' which is expressed in a formal language. But what is ... well, you get the idea: The concept of a deductive argument is a remarkably deep and fertile one, and we have our work cut out for us. We turn now to the start of that work in the following section, where we briefly introduce you to the core concepts of deductive logic in a bit more detail.

## 1.2 The Core Concepts of Logic

### 1.2.1 Sentences and Statements

English (and for that matter German, French, and so on) allows a number of different kinds of sentences, including ones that are inquisitive, imperative, and declarative. The latter type are of great importance to logic, and indeed declarative sentences in a natural language like English are very often the "raw material" given to the logician, from which she must create corresponding formulas in order to make any

[4] If you are new to logic, your learning will be enhanced and accelerated is you now go back a bit and try to figure out which of the three specimen chains given above in the present chapter, if any, are proofs.

progress. (If you read the Preface, and we certainly hope you did, you saw there that in the example involving Larry and Lucy we took English as raw material and generated corresponding formulas.) The same basic structure applies not only to natural languages, but to mathematical languages, where declarative statements (like Goldbach's Conjecture, visited soon below) are again the raw material the logician takes in as input. [We are going to have to wait rather a long while (Chapter 6) before we can see how Goldbach's Conjecture, upon being given as raw material, can be converted by the logician into an unambiguous formula with a certain measurable complexity.]

As you well know, we are currently communicating with you through written sentences in the English language. In this language, it is a declarative sentence which can take on semantic values like 'true' or 'false.' For example, the first two of the following three sentences is true, the third false.[5]

Henry Knox was one of George Washington's remarkably resourceful soldiers in the Revolutionary War. (1.1)

There are two prime numbers whose sum equals 24. (1.2)

There is a prime number divisible by six. (1.3)

Again, such sentences are said to be **declarative** in form. The mark of a declarative sentence is that it must have a **semantic value**, where 'true,' 'false,' and 'indeterminate' are three possible such values with which you are quite intuitively familiar. Many other semantic values are possible; for example: 'probable,' 'certain,' and 'beyond reasonable doubt.' So-called **truth-values**, the specific semantic values upon which we mostly focus in the present volume, are the semantic values 'true' and 'false.' A logic or logical system whose semantic values are restricted to just this pair is an example of a **bivalent** logic/logical system. Alternatively, such a logic or logical system could be classified as **two-valued**. One kind of **trivalent** or **three-valued** logic or logical system would be one whose semantic values are the aforementioned trio of 'true,' 'false,' and 'indeterminate.'

At least on an intuitive basis, you are well-acquainted with the two truth-values: You know that it's true that we know a thing or two about logic; you know that it's false that there is a maximally large positive integer; you know that it's true that there is at least one true statement; and so on. In addition, you have intuitive familiarity with the semantic value of 'indeterminate' as well. For example, at least currently, you are in all likelihood agnostic on whether "There's intelligent life on another planet" is true, and whether it's false; so it's value for you is indeterminate. In classical deductive logic, only truth-values are employed; and most (but not all) of our coverage will be classical. We can't do everything in formal deductive logic with this diad, or even with the triad that includes 'indeterminate,' but we can certainly do a lot with these three semantic values, as will be seen as we progress. The final chapter (9) in our book briefly covers introductory *inductive* logic, and therefore

[5] Recall that a prime number is one that is divisible only by itself and 1. As to Henry Knox and his resourcefulness, you would be inspired, we claim, to learn of "The Knox Trail," if you're not already familiar with it. A masterful recounting of the exploits of Knox and friends along this trail is provided by McCullough (2005) in his *1776*.

introduces semantic values associated with shades of uncertainty or probability (e.g., 'probable').

It's important to note that there's a substantial difference between truth and falsity in, as we might say, the real world, versus the corresponding concepts in formal logic. In the formal logics to come, we will use so-called "small caps" to make it clear when we're talking about truth and falsity in a logic, versus truth and falsity in real life. For instance, one of us (Bringsjord) is strongly inclined to hold it to be true that the best place to live in all the world is the Capital District of New York.[6] This isn't a universally affirmed proposition. For instance, there are certainly some denizens of sunny San Diego who hold it to be true that *that* place is the very best place to live; and hence that Bringsjord is quite mistaken. Formal logic is not an arbiter, in and of itself, of truth-value clashes like this. But, in formal logic, we can easily enough build models in which a formula representing Bringsjord's claim is TRUE, and in ones in which the California "opposition" is on the side of the angels as well.

Please throw off the impression that somehow logic makes everything relative, and trivializes real truth and falsity, in case you have that impression now. If everything works according to plan, the opposite will happen. The plan, in broad strokes, is that once "unsettled" things expressed in intuitive, informal terms are formalized in a logic or logics, matters can be settled. When this happens, it's usually because some statement, expressed as a formula, has been proved, and hence shown to be TRUE, and hence in turn shown to be true.

Note that not all sentences can have semantic values; the following two, for instance, are neither true nor false, nor anything else of the sort, and yet are of a type commonplace in English and other so-called **natural languages** (such as Norwegian and Japanese).

In order to be smarter, work your way through this book as soon as possible! (1.4)

Is your assignment finished? (1.5)

The second of these is inquisitive in form, the first imperative. Sometimes inquisitives can be of great interest to mathematicians and logicians. Indeed, the following inquisitive sentence, if answered, would put to rest a longstanding unresolved question in number theory.

Is every even number greater than or equal to four the sum of two prime numbers? (1.6)

Christian Goldbach famously conjectured that the answer is "Yes," and this conjecture is known far and wide in mathematics as **Goldbach's Conjecture**. We'll return to this conjecture later in the book. You may also remember our mention, in the Preface, of a famous presentation by Hilbert in 1900, in which he posed 23 difficult mathematical questions; hence his talk can be viewed as a sequence of sentences in the inquisitive mode.

[6] See his "The Best Place to Live."

In English, every declarative sentence ends with a period. In the formal sciences (formal logic, mathematics, game theory, decision theory, etc.), there are expressions that have the same general type of true-or-false content as the English statements seen above convey, and yet don't exactly abide by the grammar of English. For example, consider:

$$12 = 7 + 5 \tag{1.7}$$

We classify these expressions and those of the same type in English (i.e., declarative sentences) as **declarative statements**. The combined category of these two subcategories will for us just be **statements**.

### 1.2.2 Propositions

Suppose that we wish to convey to you that there the llama is standing on the lawn. We have a lot of options. We could utter the statement: 'The llama is standing on the lawn'; this would be an economical way to proceed. But we could also say: 'On the lawn there is a llama, and it's standing.' This wouldn't be a particularly elegant declarative sentence to utter (which you can perhaps testify to, since you may have just mouthed it quietly to yourself), but we're sure you'd get the idea. If we were feeling exceptionally long-winded, we could go with this: 'The lawn currently has upon it one animal, a member of the species *Lama glama*, and that animal is standing." What do these three statements have in common? The most prominent thing they have in common is that, as we shall say, they express the same **proposition**.

Notice that you can't touch a proposition; nor can you see it, or hear it. A proposition isn't a physical thing. But that doesn't mean that they're not important. Quite the contrary, actually. Humans for the most part quickly forget the exact language used in at least everyday communication — but retain an actionable recollection of the propositions expressed in this communication. You likely know that after the attack on Pearl Harbor, FDR used some memorable English; but you may not remember what he said, verbatim. Yet we have little doubt that you know that he expressed at least two propositions: namely, that an unjustified and very violent attack had been perpetrated against the United State, and two, that the U.S. was now formally at war with Japan.

We should point out that many other incorporeal things are important in logic and logic-based disciplines. A number of these non-physical things have already been mentioned to this point. For instance, we have spoken about algorithms. An algorithm isn't something you can touch in any way, needless to say.

When it comes to education, our view is that it's deep understanding of the underlying non-physical things that is what you should be aiming at, and it's certainly what we're aiming at helping to give you. Long after you put this book down, if you truly understand the chief propositions and algorithms defined, proved, and used herein, you will be very much the better for that understanding.

### 1.2.3 Arguments and Proofs

Of course, in mathematics and logic and related fields (e.g., computer science) it isn't enough to simply answer a question like (1.6); it isn't even enough to give an answer to such a question on the strength of some genuine evidence (such as that $6 = 5+1$, $8 = 7+1$, $10 = 5+5$). We must find *proofs* that certain answers are correct, and that certain others are wrong. As we have said, the study of proof is at the very heart of the adventure that you have embarked upon.

But of course you already know that proofs can't be constructed without ways of linking the statements involved. For example, suppose we wanted to prove the negation of (1.3); then we might reason as follows.

> **Proof**: Let $n$ be some arbitrary prime number. We know that if some number is prime, then by definition it's divisible only by itself and 1. Hence $n$ is divisible only by itself and 1. So it's not the case that $n$ is divisible by 6, for if it were, we would have a contradiction: $n$ would be divisible by itself and by 1 only, and yet at the same time by other numbers (like 3). It follows that no prime number is divisible by 6. **QED**

In this little unassuming proof (which as we shall soon enough see is an **informal proof**, since it cannot be mechanically verified by a computing machine) are found buried nearly all the essential logic required to initiate our project. For example, notice first that in the proof of (1.3) there is an inference from

$$\text{If } n \text{ is prime, then } n \text{ is divisible only by itself and 1.} \tag{1.8}$$

and

$$n \text{ is prime.} \tag{1.9}$$

to

$$n \text{ is divisible only by itself and 1.} \tag{1.10}$$

The general form of this **inference** is known as *modus ponens*. *Modus ponens* is an **inference rule** that permits us to derive from any instances of the pair of statements

$$\text{if } \phi \text{ then } \psi$$

and $\phi$, the conclusion $\phi$.

## 1.3 Summative List of Core Concepts

We now finish this section with a sequence of brief definitions of all of our core concepts. Each entry in the list will be relevant from this page to LAMA-BDL's final one.

**Declarative Sentence** A **declarative sentence** (in a language like English) is one that makes a claim that some state-of-affairs has a semantic value; this claim

can for us at any one time usually have one and only one of the core triad of semantic values, corresponding to whether the claim holds, fails to hold, or has an indeterminate semantic value (see next). A **declarative statement** is exactly like a declarative sentence, save for the fact that the former can be made in contexts like mathematics, game theory, physics, and so on, where the background language used isn't straight English (or some other natural language), but rather a mixture of English and formal notation. We lump both declarative sentences and declarative statements together to form the category of **statements**.

**The Two Core Truth-Values; Semantic Values** Statements can have one of two core **truth-values**: TRUE, FALSE. A third semantic value, INDETERMINATE, may sometimes be added. (Formal logic needs other semantic values, but this triad will take us quite far into what is after all *beginning* deductive logic.) A logic with two semantic values is **bivalent** or **two-valued**; one with three semantic values is **trivalent** or **trivalent**.

**Formula** A **formula** is simply a symbolic representation of a statement. For example, we might use the formula $L_f$ to represent that declarative sentence 'Llamas are fascinating animals.' As we shall see, formulae will need to be defined in very precise ways, so that for instance a computer can check whether a given formula is grammatically correct or not.

**Deductive Inference; Premises and Conclusions; Validity and Invalidity** A **deductive inference** is a link formed between one or more statements or formulas (the **premise(s)**), and a particular statement formula (the **conclusion**). The deductive inference is **deductively valid** (or simply **valid**) provided that it's impossible that the premises have the truth-value TRUE while the conclusion has the value FALSE. A deductive inference is **invalid** when it's not valid.

**Deductive Argument** A **deductive argument** is simply a set of one or more linked deductive inferences. A deductive argument is **valid** when all the inferences in it are valid; such an argument is **invalid** when one or more of the inferences in it are invalid.

**Proof** A **proof** is a species of deductive argument in which all the inferences therein link only statements or formulas. It's important to note that just as deductive arguments can be invalid, so can proofs. In fact, this follows deductively, since proofs are a type of deductive argument.[7] Later, we shall be able to partition proofs into the sub-category of **formal proofs** versus the sub-category of **informal proofs**. To anticipate the distinction, in a formal proof all the inferences link only formulae, and the inferences themselves conform to an exact specification of allowable rules of inference. Informal proofs will then be all those proofs that fail to qualify as formal.

[7] Alert readers will be right to observe that there is very little difference between a deductive argument and a proof.

## 1.4 "Background" Logic

Despite the fact that you are likely reading this sentence because you want to learn logic, the fact of the matter is that you *already* know a lot of logic: you have command over a large portion of **background logic**. What we have been talking about in the previous section, technically speaking, are the rudiments of *formal* logic, and *that* is probably something you have less familiarity with. Fortunately for you (and us too), your command of background logic will help you understand what is to come, for the simple reason that formal logic, at least of the standard variety, which will be our focus herein, is a sharpening of background logic. But what is background logic? At least as a first cut, this question is best answered by immediately giving an example.

Suppose that you are unsure as to where your electronic tablet is, and you need to fetch it as soon as possible. You remember, for sure, that you plugged it in last night to charge its battery, but for the life of you you can't remember where you did that — but fortunately you know that you plugged it in at one of two places: your office at work, or your home office. If you knew where it was at the moment, you would drive straight there briskly, and quickly assimilate a news story in the digital version of the *Wall Street Journal* that your boss has just texted you about: `J: Txt me *asap* your take on WSJ front-page story about our firm's new financial derivative!`. So which location is it? You dial your assistant at work:

"Alice, is my tablet by any chance plugged in in my office?"

She replies: "Hold on, John." And then a minute later she says: "Sorry, no. Nothing charging here."

Now, in order to deliver an opinion to your boss, what do you do now? Background logic dictates that you make a bead for your home, of course. But how do you know to do that? You know what to do because you instinctively know and routinely apply a rule of inference that is part of background logic; it's the rule that we can represent as

$$\frac{\text{Either } \texttt{Statement1} \text{ or } \texttt{Statement2}.;\ \texttt{Statement1} \text{ is false.}}{\texttt{Statement2} \text{ is true.}}$$

where, as you doubtless suspect, one is allowed to deduce what's below the line from what's above. We claim: Every neurobiologically normal and reasonably nurtured human being, if educated through high school, has perfect command over the deductive schema given immediately above — and indeed enjoys such command over a number of additional deductive schemas. Furthermore, we claim that such background-logic schemas are routinely used in everyday life by such humans in order to thrive; in fact, these humans probably need to use such schemas to just survive.

What other schemas are part of background logic? There are many, but we'll rest content with giving but two here and now. For the first, consider another narrative, one involving a child, Milly, informed for the first time about the concept of a prime number. Specifically, Milly learns that a prime number is one that is divisible only by itself and 1 without remainder; and she is told as well that 3 is a prime number,

and that 5 is as well. Milly then diligently verifies that both 3 and 5 can be divided without remainder only by 1 and themselves. Milly's teacher then asks: "Milly, is 11 a prime number?" Milly raises her eyebrows and then sets to work, by trying out all the relevant cases of attempted division. She writes that $11 \div 1$ is 11, that $11 \div 2$ "doesn't work," that $11 \div 3$ doesn't either ... but that (of course) $11 \div 11$ equals 1. Milly declares: "Yes! 11 is a prime number!"

In this example, even young Milly has skillfully used another deductive schema that is part of background logic, and which therefore you know and have long used; namely:

$$\frac{\texttt{All As are Bs.; Thing a is an A.}}{\texttt{Thing a is a B.}}$$

In the instance of the schema that Milly has employed, the collection of `As` are those numbers divisible without remainder only by 1 and themselves, `Bs` are the collection of prime numbers, and thing `a` is the number 11, which as she mechanically verifies is indeed a member of `A`. She then deduces that 11 is a `B`; that is, is a prime number.

We don't mean to suggest that background logic comprises all the rules of valid deductive inference seen in classical formal logic. Such as suggestion would be demonstrably false, in light of the empirical facts. For instance, consider the following rule $R$ of deductive inference.

$$\frac{\texttt{It's not the case that: if P, then Q.}}{\texttt{P is true.}}$$

Is this a deductively valid rule? Even if you see that it is, we are pretty sure that you're a *lot* less confident in the case of this rule, than in the two presented and discussed before it. Psychologists have learned that many human beings, even college-educated ones, fail to see that $R$ is indeed valid — and yet as we shall soon see, the rule, at least in beginning deductive logic, which is a key part of formal logic, is provably valid. As an exercise now, see if you can give an informal proof of $R$.

We could proceed to enumerate many additional aspects of background logic, but instead simply conclude this section by pointing out that nearly all of the rudiments of formal logic, enumerated in the previous section (§1.3), have direct correlates in background logic, and therefore you are in a most favorable position with respect to learning the content in the sequel. For example, you will have noticed that both of the two deductive schemas presented recently above have as their core building blocks *statements*, a type of object that we have classified as a core concept in formal logic.

## 1.5 Sets and Such

We could easily have extended the discussion of background logic to include many principles about collections, or sets. Instead, we have elected to plunge directly into giving you the concepts and principles about sets that formal logic presupposes. Essentially, what beginning (deductive) formal logic presupposes is what is

often called **naïve set theory**, to which we now turn. Later on (§6.4), we shall have occasion to study *non*-naïve set theory, in the form of a fixed set of axioms for set theory.

### 1.5.1 Naïve Set Theory

Another fundamental concept indispensable to our coming investigation is that of a **set**, which we regard to be simply a collection of objects.[8] Sets, as you doubtless know, can have members, or **elements**. For example, the set of all natural numbers $\{0,1,2,\ldots\}$, denoted hereafter as $\mathscr{N}$, has the number 7 as an element; in symbols:

$$7 \in \mathscr{N}.$$

We may say in such cases that **contains** 7, or that 7 **is an element of** $\mathscr{N}$, or simply that 7 is **in** $\mathscr{N}$. When some object is *not* a member of some set, as is true of the object $\pi = 3.14\ldots$ and the set $\mathscr{N}$, we use the notation

$$\pi \notin \mathscr{N}.$$

The number $\pi$, though not a natural number, is a **real number**; so, with the real numbers (= reals) denoted by $\mathscr{R}$, it is true that $\pi \in \mathscr{R}$.

So far all our examples of sets have been both number-related and infinite. These are not properties all of the sets of interest to us must have. For example, the set of all characters in an **alphabet** that anchors a given logical system (such as the alphabet of the propositional calculus, which will occupy us in the next chapter) is often naturally of interest to logicians, computer scientists, mathematicians, philosophers, and engineers, etc. To return to the realm of the numerical, the set of all natural numbers less than 5000 is a finite set; it can be denoted by

$$\{0,1,2,3,\ldots,5000\}.$$

Alert readers will have noticed that we have invoked the concepts of infinite and finite set without providing definitions. Precise definitions of these concepts are outside our purview, but we can rest on the following characterizations. A set that can be displayed in the form of a list from first element to last element — as in the example $\{0,1,2,3,\ldots,5000\}$ above — is finite; a set that cannot be so displayed — as in the case of $\mathscr{N}$, which has no last or final element — is infinite. The set containing no objects, the **empty set** or **null set**, denoted by $\emptyset$, will also be considered finite. We often use the elipsis, ..., to indicate that we are dealing with a set that cannot be listed from first element to final element. You may remember that we used the elipsis for this effect in the very first paragraph of this section, when indicating the

[8] This is the so-called "naïve" conception of a set — naïve because, as Bertrand Russell proved to Gotleib Frege's horror, this conception quickly leads to outright contradiction. (Russell's proof has come to be known as Russell's Paradox; we shall cover it in due course.) There are ways to circumvent this contradiction (via, e.g., axiomatic set theory, one example of which is the Zermelo-Fraenkel approach), but these routes are outside our purposes in the present, introductory chapter. Russell's Paradox and axiomatic set theory are discussed in the section "Axiomatic Set Theory" (§6.4) in Chapter 6 of this book.

elements of $\mathscr{N}$. We also used these small but powerful "three dots" when displaying $\pi$, in order to remind you that this number's decimal expansion is infinitely long.

It will be convenient to refer to sets by way of other sets and certain properties. For example, the finite set $\{1,2,3,4,5,6,7\}$ could be denoted by the expression 'the set of all elements in $\mathscr{N}$ which are less than or equal to seven and not equal to zero.' Such expressions, as you may suspect, can quickly get rather unwieldy. Accordingly, some notation (sometimes called **set-builder notation**) is in order. With respect to the previous example, this notation looks like this:

$$\{n \in \mathscr{N} : n \leq 7 \,\&\, n \neq 0\}.$$

In general, given some property $P$ and some set $A$, we may define the set consisting of all those members of $A$ having $P$. For example, the set $\{0,1,2,3,\ldots,5000\}$, in this scheme, becomes

$$\{n \in \mathscr{N} : n \leq 5000\}.$$

And, as a final exemplar of set-builder notation, the (positive) even numbers, $\mathscr{E} = \{2,4,6,8,\ldots\}$, are

$$\{n \in \mathscr{E} : n \in \mathscr{Z}^+ \text{ and divisible by } 2\}.$$

Notice that every element of $\mathscr{E}$ is also an element of $\mathscr{N}$. When such a situation holds, we say that the first set is a **subset** of the second. In symbols, the present case is represented as $\mathscr{E} \subseteq \mathscr{N}$. Notice that the converse is not true; that is, $\mathscr{N} \nsubseteq \mathscr{E}$. In this case we say that $\mathscr{E}$ is a **proper subset** of $\mathscr{N}$, written $\mathscr{E} \subset \mathscr{N}$. Two sets $A$ and $B$ are subsets of each other if and only if $A = B$. It will expedite matters if we agree, from this point on, that the empty set is a subset of all sets, and that all sets are subsets of themselves. Two sets that have no elements in common are said to be **disjoint**.[9]

There are three well-known and useful operations on sets worth isolating at this point, namely

- **union**, denoted by $\cup$
- **intersection**, denoted by $\cap$
- **difference**, denoted by $-$

These operations are easily defined using our set-builder notation:

- **union**: $A \cup B = \{x : x \in A \text{ or } x \in B\}$
- **intersection**: $A \cap B = \{x : x \in A \text{ and } x \in B\}$
- **difference**: $A - B = \{x : x \in A \text{ and } x \notin B\}$

The three operations obey certain "laws," all of which are rather easily proved. One such law is "Commutativity," which consists in the following two identities (where $A$ and $B$ are arbitrary sets).

$$A \cup B = B \cup A$$

[9] That is, two sets $A$ and $B$ are disjoint if and only if $A \cap B = \emptyset$.

$$A \cap B = B \cap A$$

Another is “DeMorgan Laws" (for sets; the counterparts for formulas in the propositional calculus, covered in the next chapter, are somewhat trickier, as we'll see), which consists of these two identitites:

$$A-(B \cup C)=(A-B)\cup(A-C)$$

$$A-(B \cap C)=(A-B)\cap(A-C)$$

Another operator is the **power set** operator; it generates the set of all subsets of a given set. Let $A=\{1,2,3\}$; then the power set of $A$, denoted by $2^A$ (or $\wp(A)$), is

$$\{A,\{1\},\{2\},\{3\},\{1,2\},\{1,3\},\{2,3\},\emptyset\}.$$

More generally, and expressed in our notation, given that $A$ is any set, $2^A=\{B : B \subseteq A\}$.

Notice that two sets with the same elements, regardless of how those elements are ordered, are one and the same. Hence, for instance, $\{1,2,3\}=\{3,2,1\}$. However, in the case of an **n-tuple**, order *is* important. We denote $n$-tuples by flanking them not with the curly brackets we have used to signify sets, but rather with parentheses. So, for example, (a, b, c) is an ordered 3-tuple, or simply a **triple** (a **pair** is an ordered 2-tuple). Note that while $(a,b,c)\neq(a,c,b)$, $\{a,b,c\}=\{a,c,b\}$.

#### 1.5.1.1 Exercises

- Provide informal proofs of the following:

  $A \cup B = B \cup A$

  $A \cap B = B \cap A$

  $A-(B \cup C)=(A-B)\cup(A-C)$

  $A-(B \cap C)=(A-B)\cap(A-C)$

### 1.5.2 Relations and Functions

Suppose that Selmer is taller than Josh. Then Selmer stands in a certain relation to Josh, and *vice versa* as well. What relation? The relation ‘taller than.’ But how can the meaning of such a relation be captured in austere, mathematical terms? The traditional answer, one that we will explicitly affirm and explain in Chapter 3: First-Order Logic, is to say that relations are to be defined in terms of set membership. For example, the relation ‘taller than,’ given that we're talking about humans, can be defined as the set of all pairs of humans $(x, y)$ such that $x$ is taller than $y$. As another example, consider the relation ‘greater than,’ where the domain in question is the set of natural numbers {0, 1, 2, 3, 4, }. In this case, the relation ‘greater than’ can be identified with the set

$$\{(x, y) \mid x > y\}$$

.

Given the concept of ordered pairs, we can define the **Cartesian product** of two sets $A$ and $B$, denoted $A \times B$, to be the set of all ordered pairs $(x, y)$ such that $x \in A$ and $y \in B$. For example, if $A$ is {2, 4, 6}, and $B$ is {3, 5, 7}, then $\{2,4,6\} \times \{3,5,7\}$ is

$$\{(2,3),(2,5),(2,7),(4,3),(4,5),(4,7),(6,3),(6,5),(6,7)\}.$$

Cartesian product can be extended to any number of sets. The $n$-**fold** Cartesian product $A_1 \times A_2 \times \ldots \times A_n$ is the set of all $n$-tuples $(a_1, \ldots, a_n)$, where $a_i \in A_i$, for $i$ from 1 to $n$. The $n$-fold Cartesian product of $A$ with itself, $A \times \ldots \times A$, is written $A^n$. For example, $\mathbf{N}^3$ is the set of all 3-tuples (or **triples**) of natural numbers.

A **binary relation** on two sets $A$ and $B$ is a subset of $A \times B$. For example, the ordinary less-than relation $<$ is a binary relation on **N**; more formally, $<$ is

$$\{(x, y) \mid x < y \text{ and } x, y \in \mathbf{N}\}.$$

An $n$-**ary** relation on sets $A_1, A_2, \ldots, A_n$ is a subset of $A_1 \times A_2 \times \ldots \times A_n$.

Chances are that since you're reading this book you've already been asked to deal with functions. If you were ever asked to do simple arithmetic (and of course you were, long ago), you were asked to deal with functions. Consider ordinary addition. Ordinary addition is a function from $\mathcal{N} \times \mathcal{N}$ to $\mathcal{N}$ (written $f : \mathcal{N} \times \mathcal{N} \longrightarrow \mathcal{N}$): you "plug into" this function pairs of numbers, for example (3,4), and receive back individual numbers, in this case 7. Of course, we usually write this fact as $3 + 4 = 7$.

Talking of "plugging in" things to a function, as you can probably guess, is too informal. What, precisely, *is* a function? A function $f : A \rightarrow B$ associates each member of $A$ with a unique member of $B$. To spell it out explicitly: A function $f : A \times A \rightarrow B$ is a subset of $A \times B$ such that if $(a, b) \in f$ and $(a, c) \in f$ as well, then $b = c$.

In a function $f$ from $A$ to $B$, $A$ is the **domain** and $B$ the **range**. For such a function, we follow usual notational practice and write $f(a)$, where $a \in A$, to refer to the $b \in B$ such that $(a, b) \in f$.

There are certain kinds of functions worth isolating even at this early stage in our investigations. A function $f : A \rightarrow B$ is

- **one-to-one** or **injective** if and only if for every two elements $a$ and $a'$ in $A$ where $a \neq a'$, $f(a) \neq f(a')$.
- **onto** or **surjective** if and only if for every $b \in B$ there is an $a \in A$ such that $f(a) = b$.
- **bijective** iff (short for 'if and only if') $f$ is both injective and surjective.

#### 1.5.2.1 Exercises

- Give an informal proof that if there is a surjective function $f$ from $\mathcal{N}$ to some set $A$, there is an injective function $g$ from $A$ to $\mathcal{N}$.

### 1.5.3 Finite and Infinite Sets and Alphabets

A finite set is one that has $n$ members, for some natural number $n$; this definition seems to do just fine. But what about *infinite* sets? How can *they* be defined? This is not an easy question. One possible answer, given to us by the late, great mathematician Dedekind, is to say that an infinite set is one that can be "paired off" with a proper subset of itself. For example, consider yet again $\mathcal{N}$. A proper subset of this set is the set $\mathcal{E}$ of even natural numbers $\{0,2,4,6,\ldots\}$. $\mathcal{N}$ can be paired off with $\mathcal{E}$:

$$0\ 0, 1\ 2, 2\ 4, 3\ 6, \ldots$$

Though this is a promising way to characterize infinite sets, it is nonetheless somewhat controversial, and we will not officially affirm it. We will instead say simply that infinite sets are those sets that are not finite. However, the general notion that the "size" of set can be based on the notion of pairing off is one that we *will* affirm — but we will make this notion a bit more precise. We will formalize the concept of pairing off through the term **equinumerous**; two sets $A$ and $B$ will be said to be equinumerous (abbreviated $A \sim B$) provided that there is a bijective function (or, as we can say, a **bijection**) $f : A \longrightarrow B$. A set that is equinumerous with $\mathcal{N}$ is said to be **countably infinite**; a set that is either countably infinite or finite is said to be **countable**.

Finally, we say explicitly what was left implicit above: **alphabets** are simply sets of **symbols**. For example, {0, 1} is an alphabet, as is {a, b, c, …, z}, which is an alphabet speakers and readers of English will be rather familiar with. The **length** of the alphabet {0, 1} is 2; the length of the second example just given is of course 26. Where a string is just a sequence of symbols from some alphabet $\mathcal{A}$, we denote the set of all strings that can be built from some alphabet $\mathcal{A}$ by $\mathcal{A}^*$.

### 1.5.4 Mathematical Induction

With not only finite sets introduced, but infinite ones as well, it behooves us to introduce the powerful proof technique of **mathematical induction**, for this technique will enable us to come to know things about infinite sets. Given use of what we have said above about so-called background logic (§1.4), mathematical induction, while initially forbidding for some students, becomes even common-sensical.

To see mathematical induction in action, we now prove that the set $\mathcal{E}$ of even numbers (defined rather pedantically above with help from set-builder notation) is equinumerous with the positive integers ($\mathcal{Z}^+$):

**Theorem**: $\mathcal{E} \sim \mathcal{Z}^+$.

**Proof**:

#### 1.5.4.1 Exercises

- Prove that if a set $A$ is countable, there is a surjective function $f$ from $\mathcal{N}$ to $A$.
- Prove that $\mathcal{N} \sim \mathcal{Z}^+$. ($\mathcal{Z}^+$ is the set $\{1,2,3,4,\ldots\}$ of positive integers.)

- Define a function from $\mathscr{E}$ to $\mathscr{N}$ and prove that your function is bijective (thus proving $\mathscr{E} \sim \mathscr{N}$).
- Define a function from $\mathscr{N}$ to $\mathscr{Q}^+$ (the positive rational numbers; i.e., numbers that can be defined as ratios of positive integers, like $\frac{2}{34}$ and $\frac{89}{22}$) and prove that your function is bijective (thus proving $\mathscr{N} \sim \mathscr{Q}^+$). (There are many routes that yield success on this challenge. A hint helpful for one route includes consideration of the following array.

$$\begin{array}{ccccccc}
\frac{1}{1} & \frac{2}{1} & \frac{3}{1} & \frac{4}{1} & \frac{5}{1} & \cdots \\
\frac{1}{2} & \frac{2}{2} & \frac{3}{2} & \frac{4}{2} & \frac{5}{2} & \cdots \\
\frac{1}{3} & \frac{2}{3} & \frac{3}{3} & \frac{4}{3} & \frac{5}{3} & \cdots \\
\frac{1}{4} & \frac{2}{4} & \frac{3}{4} & \frac{4}{4} & \frac{5}{4} & \cdots \\
\frac{1}{5} & \frac{2}{5} & \frac{3}{5} & \frac{4}{5} & \frac{5}{5} & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots &
\end{array}$$

  We encourage you to specifically ponder how you could algorithmically list all the elements of this array, starting with the first entry in the list, then moving to the second, then the third, then the fourth $= 4^{th}$, the $5^{th}, \ldots, 6^{th}, 7^{th}$, etc. If you succeed here, would you not have managed to essentially define a function $f$ from $\mathscr{Z}+$ to $\mathscr{Q}^+$? Assuming you agree, is $f$ surjective? Injective? If not injective, would you somehow modify the situation in order to yield a bijective function?)
- Where $\mathscr{A}$ is some finite alphabet, prove that $\mathscr{A}^*$ is countable.
- Let $A_1$ and $A_3$ be countably infinite sets, and suppose that for some set $A_3$ we have $A_1 \subset A_2 \subset A_3$. Prove that $A_2$ is countably infinite as well.
- Prove that $[0,1] \subset \mathscr{R} \not\sim \mathscr{Z}^+$. (It's safe to say that attempting to prove this theorem, without any advice or hints wouldn't be a cakewalk. Those who wish to tackle the naked, formidable challenge shouldn't read beyond the next semi-colon; for the others: Attempt a *reductio* proof, by starting with the supposition that in fact $[0,1] \sim \mathscr{N}$. Next, consider what this supposition immediately implies. E.g., the supposition entails that there is progression like the following.

$$\begin{array}{ccccccc}
1 & 2 & 3 & \ldots & n & n+1 & \ldots \\
\downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \ldots \\
r_1 & r_2 & r_3 & \ldots & r_n & r_{n+1} & \ldots
\end{array}$$

  The core idea here is simply that positive integer 1 gets mapped to real number $r_1$ in the interval $[0,1]$, 2 gets mapped to real $r_2$ in this interval, and so on *ad infinitum*, in such a way that every real number in the interval appears sooner or later in this progression. A more economical way to denote the progression is to give only the last row in it, and simply remember that each subscript $i$ in a label '$r_i$' indicates the positive integer $i$ that was mapped to the real in question. Given this setup, and given that each of the reals on this list can be represented in decimal notation, we suggest that you consider a tabular depiction of the list, where each row corresponds to a real $r_i$. To give you this tabular depiction, we first remind you that every decimal place in a decimal representation of a real in the interval $[0,1]$ is simply occupied by one of the digits from the set $\{0,1,2,\ldots,9\}$. In the following table, a digit selected from this set for

row $k$ and column $m$ is denoted by $d_{km}$. Now here's the table that is implied by the supposition:

| | | | | | | |
|---|---|---|---|---|---|---|
| $r_1$ | $d_{11}$ | $d_{12}$ | $d_{13}$ | $d_{14}$ | $d_{15}$ | $\ldots$ |
| $r_2$ | $d_{21}$ | $d_{22}$ | $d_{23}$ | $d_{24}$ | $d_{25}$ | $\ldots$ |
| $r_3$ | $d_{31}$ | $d_{32}$ | $d_{33}$ | $d_{34}$ | $d_{35}$ | $\ldots$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | |

In other words, the supposition is that every real number in the interval in question appears as a row in this table. Do you see a way to define a real in decimal form that is guaranteed *not* to appear in this table?)

## 1.6 Computers and Computation

Our coming investigation of deductive logic requires that we have a rudimentary but nonetheless fairly rigorous understanding of computers and computation. This requirement is in line with the intertwined history of logic, mathematics, and computation; for in this history, these three areas are inseparable, and the intimate interconnectivity between them has been in place and under refinement since at least three centuries before Christ. For example, Aristotle sought to systematically explain why it was that the compelling proofs of Euclid, which made liberal (and by today's standards, rather fuzzy) use of algorithms (and therefore of computation), *were* compelling; and his explanation came in the form of his inventing and specifying a rather limited formal logic composed of what we now call **syllogisms**. One of the valid syllogisms in Aristotle's logic is the following one, which is without doubt a part of background logic, and only a slight variant of a deductive schema you saw above.

$$\frac{\texttt{All As are Bs.;All Bs are Cs.}}{\texttt{All As are Cs.}}$$

No doubt you agree that this is a valid deductive schema.[10] The "tied togetherness" of the trio — of logic, math, and computation — remains in full force today. You may have noticed a symptom of this togetherness in the courses available at colleges and universities in computer science, since curricula for computer science invariably include at least some coverage of formal logic.[11] And the togetherness is very concrete and vibrant in the area of **program verification**, in which practitioners attempt to prove that even complex computer programs will behave as they are intended to. We will have occasion to revisit in earnest the problem of program verification later.

[10] For the full story of Euclid and Aristotle and beyond, and readable coverage of many of the technical details in the drama featuring these two as early performers, see (Glymour 1992).

[11] This coverage may be offered in courses whose titles don't contain the word 'logic.' E.g., a course called 'Discrete Mathematics' will invariably cover some basic formal logic, and any course providing rigorous, theoretical coverage of computation and complexity measures thereof will include a fair dose of formal logic.

Because computers and computation are woven into the fabric of daily life in the civilized world, it may seem odd to you that we can't simply assume that you already have sufficient understanding of this sphere, and move on. It's certainly *possible* that you have a deep understanding of computers and computation. For example, you may have been previously exposed to **Turing machines** (which were mentioned in passing in the Preface), and/or frameworks equivalent to such devices. If so, it's fine for you to skim the present section. Readers for whom 'Turing machine' is a mystery are strongly advised to read the present section more carefully. This recommendation is one we issue even to those who, despite being unfamiliar with Turing machines, have substantial experience with one or more modern programming languages (e.g, Java, or C$^{++}$). The reason is that, while certainly an intellectual travesty, introductory programming courses routinely eschew coverage of the formal essence of the programs and programming patterns that students are exposed to. This wrongheaded approach has led to a situation in which, while students can write some programs, and while these programs, when executed, cause computation to unfold on digital computers, students nonetheless haven't the foggiest idea as to what computation and computers really and truly are. In the present book, such ignorance would be fatal, and we must head it off now.

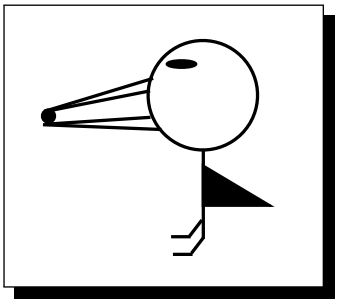
In the Preface, we made reference not to any of the "usual suspects" taken to define the essence of what a finite computer or finite computation, formally speaking, is. For instance, in the Preface we referred to **Turing machines** (devices consisting of a tape divided into squares on which symbols can be inscribed, along with a read/write head that reads and writes symbols on these squares), and also to **KU-machines**, the idealization of computation that we generally prefer. We now offer our own friendlier version of KU-machines, which will serve to explain what we take computation and computers to fundemantally be, and to anchor subsequent reference to that which can be processed by a finite computer. Our version of KU-machines is in part inspired by the methods of Gaspard de Prony, a French mathematician and innovative engineer. de Prony set out to produce vast logarithmic and trigonometric tables, by using a great number of humans who functioned as mechnical calculators. Many of these — as we now can informatively call them — "human computers" were unemployed hairdressers, whose rich clients, after the French Revolution, disappeared. (Well, yes, 'disappeared' is rather mild, given the rather ... unsavory side of this revolution. Marie Antoinette's acephelous end is presumably a case in point.)

It should at this juncture be noted that there are in general two avenues leading to the production of a fixed framework that pins down, with sufficient rigor, computation and computers, and which therefore in turn pins down what we shall heretofore mean when we say such things as that a given set $A$ is **decidable**, or that a given function $f$ is **computable**. These avenues are quite different. The first avenue is to provide the student with a tedious, decidedly sub-human characterization of computers and computation. The second route, a more cognitively plausible and hopefully less tedious one, and the route we shall take in the present volume, is to start with what human beings can do, and then scale back from that capacity to a fragment of what humans can do when operating mechanically, in precise and

slavish obedience to a set of finite instructions.[12]

So that the contrast between these two routes is plain, we now travel appreciably down the first one, by describing "Raven" machines, or, for short, **R-machines**. These machines include, first and foremost, a raven: Roger. Roger is a thoroughly obedient bird whose range of activity is highly restricted. Roger is shown in Figure 1.1. You will note that he is holding something in his beak. What is it? It's a little round stone. Roger doesn't fly (at least when he is working for us); when we tell him to start a work session, he simply moves little round stones around, in accordance with programs that we provide to him, and he halts when we tell him to conclude a work session. More specifically, his movement of the stones is confined to moving them into and out of numbered boxes. For any given work session, we provide Roger with $n$ boxes to start, and if his program makes reference to the number $m$ of a box beyond the ones he intially has, Roger calls out "More," and instantly a new box numbered $m$ appears for him to employ.[13] R-machines consist of the combination of: programs to instruct Roger, Roger himself with his perception and action powers, and the stones and boxes. Figure 1.2 shows a snapshot of an R-machine during some computation.

Figure 1.1: Roger the Raven



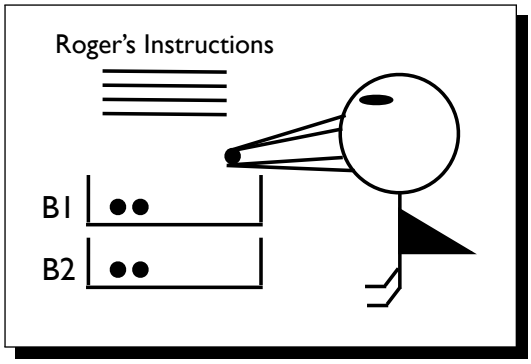
We are interested in having R-machines compute functions from $\mathcal{N}^n$ to $\mathcal{N}$. In order to enable this, we shall understand a given natural number $n$ to correspond to $n$ stones located in a given box. The natural number 0 will correspond to the absence of any stones; so an empty box is assumed to be holding 0. Hence Figure 1.2 shows a configuration in which box $B_1$ holds stones representing the numeral '2,' and box $B_2$ holds stones representing the numeral '3.' Ordinary addition + of natural numbers is of course such that

$$+ : \mathcal{N}^2 \longrightarrow \mathcal{N},$$

[12] To their credit, both Turing (1937) and Post (1943), while regarded to be originators of the first route, did both briefly orient their tedious, sub-human frameworks (resp., **Turing machines** and **Post canonical systems**) around an intuitive concept of what human information-processesors, operating in uninspired, mechanical fashion, can do.

[13] Alternatively, we could imagine that Roger's call for another box results in a box from an infinite supply provided at the outset of his efforts, moving into his work area.

Figure 1.2: Snapshot of an R-machine in Operation




and readers have known since a very young age how to compute this simple arithmetic function. Can Roger compute it? Yes, easily. But in order to see how, we need to specify the format of the instructions that we provide him with.

Each instruction to Roger is one of five possible types. We now define this quintet, by giving the schema for each one, and in addition an intuitive explanation of what the instruction communicates to our bird. Note that every instruction begins with a natural number $l$ that serves as its label.

1. $l$ SET $\mathrm{B}_i = \mathrm{B}_i - \bullet$

   This instruction tells Roger to take away one stone from box $\mathrm{B}_i$. If this box happens to be empty, Roger doesn't do anything, and simply moves on to the next instruction.

2. $l$ SET $\mathrm{B}_i = \mathrm{B}_i + \bullet$

   This instruction tells Roger to add one stone to box $\mathrm{B}_i$.

3. $l$ IF $\mathrm{B}_i = e$ THEN $j$ ELSE $k$

   This instruction tells Roger that if box $\mathrm{B}_i$ is empty, he should shift his attention to, and follow, instruction with label $j$; otherwise he should move to instruction with label $k$.

4. $l$ ROGER, POINT TO $\mathrm{B}_i$

   This instruction tells Roger to point to box $\mathrm{B}_i$, in order to inform us that this is output he wishes us to have.

5. $l$ ROGER, HALT

   This instruction simply tells Roger to halt. In any set of instructions given to Roger (i.e., in any **R-program** given to him), there can only be one instruction of this type.

Let's now put these schemas into action, in the form of an R-program for Roger that carries out addition. In order to do that, we shall assume that box $B_1$'s contents denotes the first of the two numbers to be added, and that box $B_2$'s contents denotes the second. Here then is a program for addition:

0 IF $B_1 = e$ THEN 5 ELSE 1
1 IF $B_2 = e$ THEN 6 ELSE 2
2 SET $B_2 = B_2 - \bullet$
3 SET $B_1 = B_1 + \bullet$
4 IF $B_2 = e$ THEN 6 ELSE 2
5 ROGER, POINT TO $B_2$
6 ROGER, POINT TO $B_1$
7 ROGER, HALT

You should make sure that you see how this simple program, when followed by Roger, gets the job done. With an initial input of $\bullet\bullet$ in box $B_1$, and $\bullet\bullet\bullet$ in box $B_2$ (i.e., an initial configuation that constitutes a request to Roger that he tells us what $2+3$ is), the program given here will cause Roger to point to $B_1$ when it contains $\bullet\bullet\bullet\bullet\bullet$, at which point he will stop. We encourage the reader to verify his or her understanding of R-machines by creating an R-program for Roger that carries out multiplication of two natural numbers; this is one of the exercises given below.

How do R-machines answer the questions "What is computer?" and "What is computation?"? There are certainly more powerful computers than what R-machines capture, but the answer to the first question, for purposes of the present book, is that computers are those things that can do what R-machines can do; and correspondingly, computation is then taken here in to consist in what R-machines do as through time they follow the programs given to them.

We can introduce notation and terminology that unpacks and renders rigorous these answers, as follows. We can write

$$R : \bullet^n \longrightarrow \bullet^j$$

to express that a given R-machine $R$, upon starting with input $\bullet^n$ (located by convention in box $B_1$), outputs $\bullet^j$ and halts.[14] We can then say that a function $f$ that maps $\mathcal{N}$ to $\mathcal{N}$ is an **R-computable** function if and only if there exists an R-machine $R$ such that

$$R^+ : \bullet^n \longrightarrow \bullet^{f(n)}.$$

It's useful not only to have machines available that can compute functions, but also for example to make decisions. We can interpret an output of just one stone in box $B_1$ to convey a "Yes" from Roger, and the output of an empty $B_1$ to convey a "No" from him. Given this convention, we can say that a set $A$ of natural numbers is **R-decidable** just in case there is an R-machine $R$ such that, for every $n \in \mathcal{N}$, when $n$ is in $A$, $R$ goes from $\bullet^n$ to $\bullet$, and when $n \notin \mathcal{N}$, $R$ goes from $\bullet^n$ to $e$. For instance,

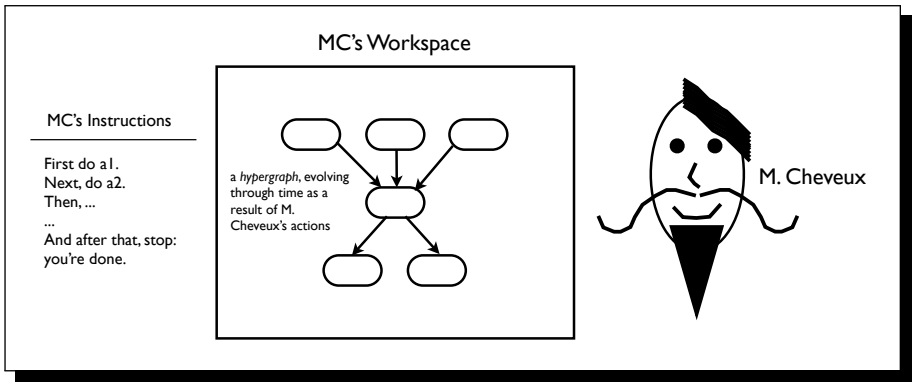
[14] This notation can be extended in the obvious way to cover not just unary functions, but $n$-ary ones. We rest content with the unary case here.

the set of even natural numbers is obviously R-decidable, and you should find it easy enough to devise an R-program that enables Roger to decide this set.

We say that devising such a program is *easy*; that doesn't mean that doing so isn't tedious; for most people, it is. This brings us to the second route to clarifying the nature of computers and computation. As we have said, in this route, instead of using mere animal minds to anchor the concepts of computer and computation, we start with *our* minds.

Our more human-like computing machines, **MC-machines** (which, recall, are inspired by KU-machines) include a workspace in which hypergraphs are processed by a meticulous hairdresser, Monsieur Cheveux. M. Cheveux operates on hypergraphs and their components in accordance instructions supplied to him. Figure 1.3 summarizes the basic situation: it indicates that M. Cheveux is present, and ready to follow the commands given to him; it gives a few (admittedly vague) instruction schemas; and it provides a glimpse at his workspace, from a birds-eye point of view.

Figure 1.3: Pictorial Summary of Basic Elements of MC-machines

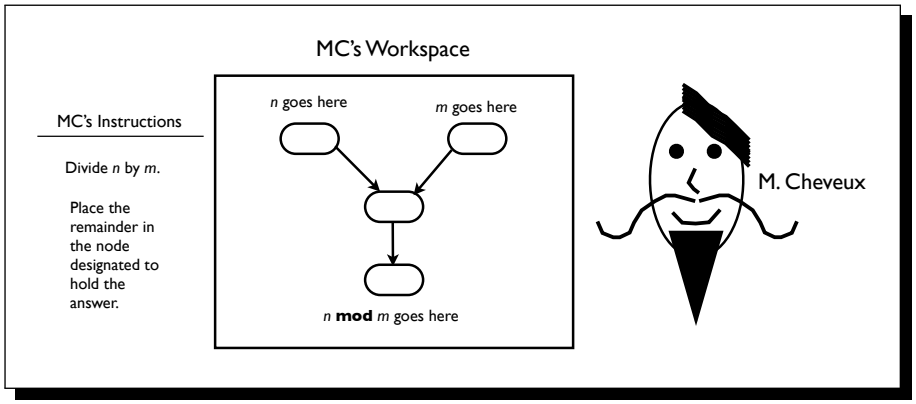



What is shown in Figure 1.3 is very abstract. To bring things into focus a bit, and to show that we opt for MC-machines because they provide a simple and cognitively realistic model of computation that accords with the kind of thing you do as a problem-solver, let's consider not addition, as we did in connection with Roger, but a different arithmetic function: the so-called **modulo** function. This is a binary function taking two natural numbers $m$ and $n$ and returning the remainder after $m$ is divided by $n$.

You will agree that the task of specifying an R-machine that computes this function would take you a while. For in order to to it, you would first have to write an R-program to compute the binary function $\div$, and that would not be done in just a few seconds. Things are radically easier when we use an MC-machine. This is revealed by Figure 1.4, which shows the counterpart of an R-program for Roger. M. Cheveux will be done mighty quick, for all he needs to do is follow the two-step algorithm given to him, once the two top nodes are stocked with the inputs.

From this point on, when we say that some function is computable, or that some set is decidable, this will be short for saying that the function in question is

Figure 1.4: An MC-machine Ready to Compute $n$ **mod** $m$




**MC-computable** and the set in question is **MC-decidable**. Moreoever, when we want to show that some function is computable or some set is decidable, we need only provide an algorithm that could be used by M. Cheveux as he works with nodes in a hypergraph. As you will soon see, when you work with Slate, you will sometimes by operating pretty much in the matter of M. Cheveux. Notice that we say *sometimes*. Plenty of the problem-solving you will need to pull off will require not just mechanical following of an algorithm, but creativity that cannot, as far as we know, be reduced to the realm of the algorithmic.

Some readers may wonder whether whatever M. Cheveux can do, Roger can do. The answer is "Yes." We do not supply the proof here, and leave its construction to diligent readers. That is to say, where **R** denotes the set of all number-theoretic functions computable by an R-machines, and **C** denotes the set of all such functions computable by MC-machines, we leave it to the reader to prove that $\mathbf{R} = \mathbf{C}$.[15]

### 1.6.1 Exercises

1. Recall that the R-program above enables Roger to compute ordinary addition; we dubbed this program $R^+$, as you will remember. We claimed that

$$R^+ : (\bullet^n, \bullet^m) \longrightarrow \bullet^j \text{ iff } n + m = j.$$

   Prove that this claim is correct.

2. Is it true that for every function $f$ from $\mathcal{N}$ to $\mathcal{N}$ there is an R-machine that computes $f$? Prove that you are correct.

[15] Alert readers may suspect that there are many such identities, involving various other formal models of computation; that that all these models are ultimately equivalent. This suspicion is veridical. For instance, another model is the set of **recursive functions** (= **REC**), and they are equivalent to **R** and **C**. Likewise, where **T** is the set of Turing-computable functions (recall that we have alluded to Turing machines), we have

$$\mathbf{R} = \mathbf{C} = \mathbf{REC} = \mathbf{T}.$$

3. Write an R-program that causes Roger to compute the multiplication function $\times : \mathcal{N} \longrightarrow \mathcal{N}$.
4. Write an R-program that causes Roger to decide the set of prime numbers.

## 1.7 Why Study Formal Logic?

We guarantee there are exceptionally good reasons for you to study formal logic. In fact, if we knew a little more about your life, your background, your goals, and your personality, we could provide reasons for studying formal logic that relate to your own individual situation. For example, if your goal is be an entrepreneur and start a high-tech company in hopes of becoming a billionaire and making the world a much better place in the process, studying logic would be a very wise thing to do. Why? The reason is that in a free-market economy (which is where we assume you find yourself) it's impossible to make headway unless you can persuade other people to do what you want them to do. For example, you may well need to convince a venture capitalist to invest money in your dream, to convince employees to work hard and long in support of your dream, and to convince people to purchase what your company produces. At least if the venture capitalists, employees, and customers are *rational*, your best bet is to use reasoning to try to convince them. Put more explicitly, your best bet is to offer cogent *arguments* in support of your claim that they ought to behave as you wish them to. Given this, perhaps you sense that knowing logic would come in rather handy, and could potentially make rather a difference in your business life.

But maybe you have much more abstract and philosophical goals. For instance, perhaps you want to be more rational, less vulnerable to error, and better able to make correct decisions. If so, logic is for you. In fact, without training in logic, it's very hard to see how one can navigate the modern world, which increasingly confronts its denizens with problems that require careful and rigorous reasoning.

Do you really need to be more rational than you are? You be the judge. Consider the following simple problem.

> Assume that the following sentence is true:
>
> > If there is a king in the hand, there's an ace in the hand; or, if there *isn't* a king in the hand, there's an ace in the hand; and not both of the preceding if-then statements is true.
>
> What can you infer from this sentence?

Most college-educated people (in fact, most professors "from Seattle to Stockholm") answer that what can be inferred is that there is an ace in the hand.[16] If asked to justify this response, usually some such thing as the following is supplied: "Well, either way, whether there's a king in or not, there's an ace. So there's got to be an ace

[16] The puzzle here is from the master of such fiendish puzzles: Philip Johnson-Laird. See e.g. (Johnson-Laird 1997).

in the hand." Unfortunately, not only is this answer wrong, but the very opposite of what is said by our typical respondent holds; that is, what *can* be deduced is that there *isn't* an ace in the hand.[17]

Perhaps you breezed through this king-ace problem. That's unlikely, but possible. We could give you an endless string of ever-harder puzzles, each no more difficult than the problem of deciding, say, what investments to make — but we instead rest content with asking you to trust us that your performance on these puzzles would not be worth writing home about, and that study of logic can raise your performance.

[17] Here's an informal proof of the correct answer:

> **Proof**: We are told that either of two if-then statements holds, but not both. This means that one of the if-thens is false. There are two cases; abbreviating for economy, they are (case 1) that 'if $K$ then $A$' is false, and (case 2) that 'if not-$K$ then $A$.' Consider case 1. Here it follows immediately that $K$ holds, but $A$ doesn't. (The only way an if-then (or, to use the technical term, a **material conditional**) can be false is when the "if part" of the conditional holds but the "then part" doesn't.) In case 2, we have again a conditional that is assumed to be false, but by the same logic, not-$K$ holds but $A$ doesn't. In both cases, which are exhaustive, there is no ace in the hand. **QED**

# Chapter 2

# The Propositional Calculus

Our first logical system is the **propositional calculus**, in which, by constructing proofs, we'll regiment and verify deductive relationships between certain formulas and sets thereof. As you might expect given the terminology introduced in the previous chapter, formulas in the propositional calculus usually represent some statements; and these statements in turn express certain underlying propositions. In this light, it might be more accurate to refer not to the 'propositional calculus,' but rather to the 'statement calculus.' Nonetheless, for whatever reason, the phrase 'propositional calculus' is used more frequently on the modern-logic scene, and we won't buck this habit. Besides, supporting the traditional label is the fact that propositional content does ultimately get expressed and processed in the propositional calculus, via the propositions-to-statements-to-formulas-to-proofs progression. You should at any rate be aware of the fact that the other phrases we've mentioned are in play in certain quarters, and indeed some other phrases are as well: some authors, for example, use the term 'sentential calculus' or 'sentential logic' to refer to the propositional calculus; other more exotic terms are sometimes also used. Once you grasp the formal essence of the propositional calculus (soon!), you will be able to tell at glance, regardless of what words you see used down the road, whether what you are seeing is the real article presented in the present chapter.

## 2.1 A Preview by a Simple Example

One important deductive relationship holds between propositions (or statements, or formulas) when one or more propositions (or statements, or formulas) **deductively entails** another. We now give a simple example that previews, in nutshell form, the propositional calculus, and which displays the aforementioned progression from propositions, to statements, to formulas, to proofs.

The proposition that both Larry and Lucy are llamas most assuredly entails the proposition that Lucy is a llama. If the first holds, then the second *must* as well. This you see instantly at the propositional level. But to demonstrate and verify what you see, we must first know what statements express the propositions in question,

and then carry out a proof on the formulas that express those statements. So here we go.

Acceptable statements in this example are

**Statement 1** Larry is a llama and Lucy is a llama.

**Statement 2** Lucy is a llama.

A sensible set of building-block formulas in the propositional calculus that could be used to express this pair of statements, is: A for 'Larry is a llama,' U for 'Lucy is a llama.' (This being a preview, we pulled these formulas out of thin air. Soon we'll specify the alphabet and grammar that define the set of well-formed formulas in the propositional calculus.) We can next exploit the fact that the propositional calculus gives us a symbol for the connective 'and': $\wedge$. And then we can express our two statements about Larry and Lucy as formulas:

**Formula 1** $\mathsf{A} \wedge \mathsf{U}$

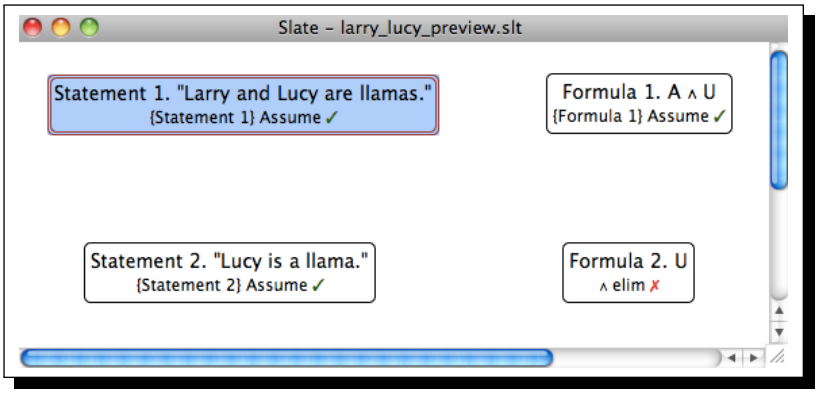
**Formula 2** $\mathsf{U}$

The first of these formulas is a so-called **conjunction**; it contains two **conjuncts**, A and U. Our formal proof of the second of these formulas from the first can be done in a single step, because it will turn out that there is a dedicated rule in the propositional calculus that allows us to deduce any of the conjuncts in a conjunction from that conjunction. This rule is known as 'conjunction elimination,' and is abbreviated in Slate as '$\wedge$ `Elim`.'

To make the preview truly end-to-end, we have started things off by preparing a Slate file for you, in which you can construct your first proof, and witness Slate's verification of it. Please open the file `larry_lucy_preview.slt` now. In order to do this, if Slate isn't already open before you, you will need to launch it, and when you see options for what kind of workspace you wish to select, choose `Propositional Calculus`. With a blank workspace now open in front of you, you will be able to open the relevant file. Please do so. Okay, now you are presented with something that looks at least very much like what is shown in Figure 2.1.

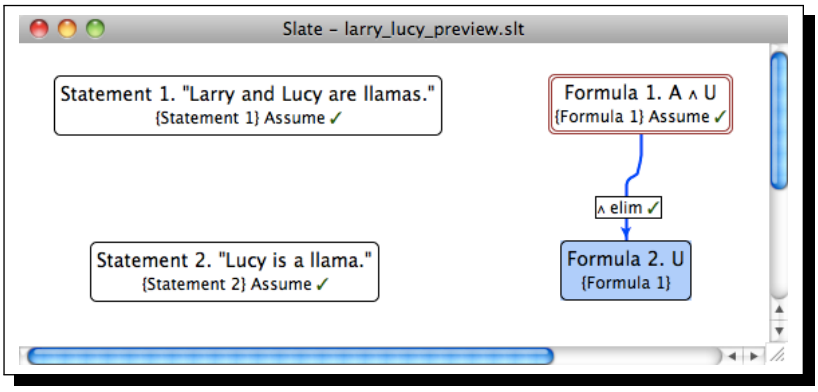
Look carefully at the screen. On the left, you will see that we have pre-loaded the two statements in question; they are labeled as such, and appear here as reminders as to what our formulas represent. To the right of the two statements are their corresponding formulas; each is given a separate node in the workspace. The task you face is to figure out how to prove Formula 2 from Formula 1. We have given you a head start, in that we have already configured the node for Formula 2 so that the rule of inference to be used in obtaining it from some other node or nodes is $\wedge$ Elim. Do you see that the string '$\wedge$ Elim' appears in the lower right-hand corner of the node for Formula 2? Good. And do you see that just to the right of this string, which again is the name of the rule of inference to be used, there appears a red ✗? This red symbol means that Slate rejects the claim that you have made; that is, the claim that Formula 2 follows by the rule in question. Of course, *you* didn't make any such claim; *we* did, because we pre-configured the file in question. But regardless, you can easily fix the situation.

Figure 2.1: First Step in Proof of Larry-Lucy Preview Example




In order to repair things, establish an arc from Formula 1 to Formula 2, and some nice magic will instantly occur. Did you see what happened? The bad-news ✗was instantly replaced by Slate with the good-news ✓— but more importantly, a brand-new node appeared, one which announces the rule of inference, and carries that pleasant green check-mark. The arrow from Formula 1 passes through the new node and then reaches the node for Formula 2. You can think of the new, intervening node as a gatekeeper; he has not only permitted the arrow to pass through, but has declared, by his posting the green check-mark, that the arrow is *allowed* to pass through. The situation should look something very much like Figure 2.2. You succeeded in discovering your first proof! Better yet: Your proof is valid, and Slate has verified that it is.

Figure 2.2: Second and Final Step in Proof of Larry-Lucy Preview Example

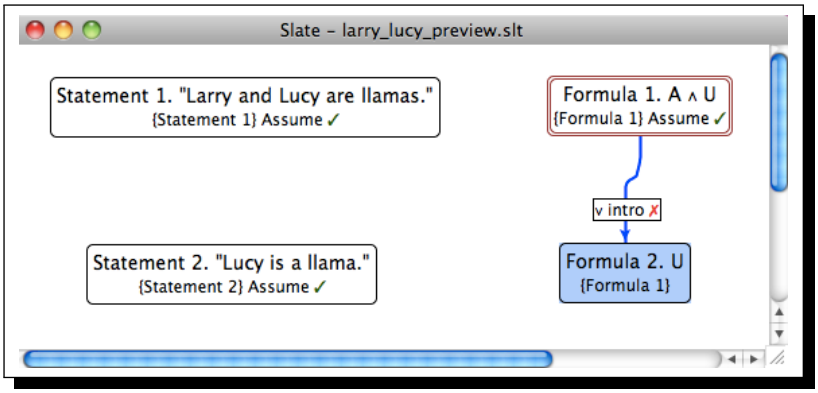



It's very important to realize that proofs can be invalid. Everyday usage tends to hide this crucial fact, because we say things like: "Of course you can believe it: he proved it!" And perhaps you've heard the expression: "A proof is a proof!" Such

utterances are elliptical. The first simply conveys that he did prove the thing in question, *and* his proof is valid. And the second utterance is a reliable and reassuring dictum, but what it means is simply that a *valid* proof is utterly unshakeable. If you know anything at all about computer programming, you know that one can write a *bona fide* computer program which is nevertheless filled with bugs — to the degree that the program in question may be quite useless. Of course, even bug-filled computer programs can have great value, because for instance they can be stepping-stones to rock-solid ones that get the desired job done. Proof are rather similar in this regard. You can construct invalid proofs (and can be given invalid proofs touted by others as valid), for sure; but often you do so on the way to a valid one that establishes your desired goal.

To hammer home and concretize the point made in the previous paragraph, consider a variant of the Slate workspace we congratulated you on obtaining; the variant is shown in Figure 2.3. You will see that there is certainly still a proof here. But the proof is invalid, as can be seen instantly in light of the fact that the unwanted ✗has returned. Why has this happened? Well, the rule of inference selected in the "gatekeeper" node simply doesn't work in this context. (You will learn later in the chapter what this rule of inference means.)

Figure 2.3: A Defective Proof (in Larry-Lucy Preview Example)




## 2.2 More on Entailment

Entailment will be a very important concept for us; we will clarify it thoroughly, and you will gain an ability to prove, in robust scenarios far exceeding the simple one just dealt with, that one or more propositions (or statements, or formulas) entails another. Notice our use of the adjective 'robust' here. Don't let the simple Larry-Lucy example here fool you: Sometimes proving entailment can make you rich and famous. We'll soon enough get to deeper examples. Using the propositional calculus, we will be able to clarify and evaluate many important reasoning patterns that occur in many domains, including mathematics, science, politics, and everyday

life. The fact is, every day on Earth millions of people make entailment claims. The only way to really and truly check whether these claims are correct is to use formal logic, and to start right where we are starting: with the propositional calculus.

The phrase 'entailment claim' is accurate, but pedantic. People make entailment claims simply by presenting **arguments**, a concept briefly discussed previously in the Introduction 1, and with which we are sure you are familiar; hence one can view the study of formal deductive logic as the scientific study of deductive arguments. We're not talking here about *quarrels*. Sometimes the term 'argument' is used to refer to a quarrel, but that won't do for us. We mean by 'argument' simply a linked list of propositions, where the links are inferences; recall that we discussed such structures in the Introduction 1. Such a linked list is an independent object that is quite inanimate, and hence bereft of emotion. A quarrel, by contrast, is often filled with emotion, and generally denotes a disagreement between one or more people. Unfortunately for the human race, decisions of great consequence are sometimes made on the basis of quarreling, rather than on the basis of unemotional analysis of arguments in the logician's sense.

## 2.3 The Boolean Values, Meta-variables, and Boolean Connectives

Given a proposition (or statement or formula), we may affirm or assert it, declaring it to be true, or deny it, declaring it to be false. (In the Introduction, we allowed a third truth-value, UNKNOWN, but the standard propositional calculus, which will be the focus of this chapter, only allows for TRUE and FALSE.) Every proposition (or statement of formula) is either true or false, but not both true and false. For instance, we affirm the proposition that Trondheim is north of Bergen, but deny the proposition that $2+2=5$. "True" and "false," recall, are said to be the potential truth-values of a proposition. These are often called **Boolean truth-values**, or sometimes simply **Boolean values**, or sometimes just **Booleans**.

To be able to affirm or deny single propositions (or statements or formulas) is important, but to study the relationships that may hold of such objects or between them (for instance, whether a proposition is necessarily true, or whether a proposition follows from another), requires some additional machinery for constructing formulas. Much of the discussion that follows will depend on a proposition or statement or formula in general, but not on any *particular* such thing. For example, while, as we have recently seen, $\mathsf{A}$ is a particular formula, it will be convenient if we have available a class of **meta-variables** to stand for formulas in general. Accordingly, we will use miniscule Greek letters to stand for arbitrary formulas. For example, $\phi$ could stand for the formula $\mathsf{A} \wedge \mathsf{U}$, and $\psi$ could stand for the formula $(\mathsf{A} \wedge \mathsf{U}) \wedge (\mathsf{A} \wedge \mathsf{U})$; you get the idea. You may have noticed that we quietly inaugurated this meta-variable use in the Introduction. It will also prove to be exceptionally useful to employ majuscule Greek letters to represent sets of formulas, as for example in using $\Phi$ to represent the set $\{\mathsf{A} \wedge \mathsf{U}, (\mathsf{A} \wedge \mathsf{U}) \wedge (\mathsf{A} \wedge \mathsf{U})$. Our sets of formulas needn't be finite in size. For example, we know that there are an infinite number of

positive integers: 1, 2, 3, and so on *ad infinitum*. We also know that the statement '$n$ is greater than 0' holds for every positive integer $n$; that is, the sequence

$$1 > 0, 2 > 0, 3 > 0, \ldots$$

goes on forever. Suppose that each of the statements in this sequence is represented by a corresponding formula; for example $\mathsf{G}_1, \mathsf{G}_2, \mathsf{G}_3$, etcetera. Then we can refer to the set $\Psi$ composed of each of these formulas.

We now develop the **Boolean connectives**, operators by which we construct from given formulas new formulas whose truth-values are completely determined by the truth-values of the given formulas.

We construct the **negation** of a formula $\phi$ by prepending the symbol $\neg$ to $\phi$, producing $\neg\phi$. A formula $\neg\phi$ is TRUE when $\phi$ is FALSE, and false otherwise. Since $\phi$ must be exactly one of TRUE or FALSE, $\neg\phi$ is TRUE exactly when $\phi$ is FALSE and FALSE exactly when $\phi$ is TRUE. It was stated earlier that least while we are in purely classical mode, every formula (and, for that matter, every statement or proposition as well) is either TRUE or FALSE, and not both TRUE and FALSE. By the definition of the truth conditions for $\neg$, it must be the case that for any formula $\phi$, exactly one of $\phi$ and $\neg\phi$ is TRUE. Thus $2+2=4$ is TRUE, but $\neg(2+2=4)$ is FALSE. The negation of a formula is yet another formula which may also be negated. For instance, it should be clear that the truth-value of a formula $\neg\neg\phi$ must be the same as the truth-value of $\phi$. The negation $\neg\phi$ can be variously read as "not $\phi$," "it is not the case that $\phi$," "$\phi$ does not hold," and so on. When $\phi$ is a formula expressed by a declarative sentence (which as you will recall is a type of statement), there may be other ways of verbalizing $\neg\phi$. For instance, $\neg$(John is tall) may be read "John is not tall."

Using negation, we construct a new formula from a single given formula, but we are often concerned with the combination of multiple formulas. In fact, negation is the only Boolean connective that requires but a single given formula; the remaining Boolean connectives all make use of two or more given formulas.

Turning now to more formal coverage of what we anticipated in the earlier Larry-Lucy example, given two formulas $\phi$ and $\psi$, or even more generally, $n$ formulas $\phi_1, \ldots, \phi_n$, we construct their **conjunction** by inserting the $\wedge$ symbol between each formula, producing $\phi\wedge\psi$, or in the more general case, $\phi_1\wedge\ldots\wedge\phi_n$. The formulas $\phi_1$, $\ldots$, $\phi_n$ are called the **conjuncts** of the conjunction $\phi_1\wedge\ldots\wedge\phi_n$. The truth-value of a conjunction of formulas $\phi_1, \ldots, \phi_n$ is TRUE exactly when each $\phi_i$ is TRUE, and false otherwise. Notice that an equivalent definition is that the conjunction $\phi_1\wedge\ldots\wedge\phi_n$ is FALSE exactly when at least one $\phi_i$ is FALSE, and TRUE otherwise.

As an example, we observed that exactly one of the two formulas $\phi$ and $\neg\phi$ must be true; therefore, we see that the proposition $\phi\wedge\neg\phi$ must be false. We also observed that the truth-value of $\neg\neg\phi$ is the same as the truth-value of $\phi$; we can see then that $\phi\wedge\neg\neg\phi$ must also have the same truth-value as $\phi$. Sometimes, following the conventions for sums and products, the conjunction $\phi_1\wedge\ldots\wedge\phi_n$ will be denoted by $\bigwedge_{i=0}^{n}\phi_i$ or by $\bigwedge_{i=0,\ldots,n}\phi_i$. A conjunction $\phi\wedge\psi$ may be read "$\phi$ and $\psi$," "both $phi$ and $\psi$," and so on.

We will continue to introduce three more boolean connectives, but before we do that, we pause to consider some of what we can express with the two connectives we already have.

We noted that an alternate definition of the truth-value of a conjunction would be that a conjunction is FALSE precisely when at least one of its conjuncts is FALSE. A conjunction is TRUE when *all* of its conjuncts are TRUE, but this alternate definition gives a truth-value to the conjunction based on *any* of its conjuncts having a particular truth-value, which has a more disjunctive feel to it. Indeed, using just negation, conjunction, and the propositions $\phi_1, \ldots, \phi_n$, we can construct a proposition TRUE exactly when at least one of $\phi_1, \ldots, \phi_n$ is TRUE. (It is a good exercise to try to construct such a formula before continuing.) One such formula is the negation of the conjunction of the negations of $\phi_1, \ldots, \phi_n$; that is, $\neg\big(\neg\phi_1 \wedge \ldots \wedge \neg\phi_n\big)$. This formula is TRUE exactly when the conjunction $\big(\neg\phi_1 \wedge \ldots \wedge \neg\phi_n\big)$ is FALSE, which holds precisely when at least one $\neg\phi_i$ is FALSE. But $\neg\phi_i$ is FALSE exactly when $\phi_i$ is TRUE. Thus $\neg\big(\neg\phi_1 \wedge \ldots \wedge \neg\phi_n\big)$ is TRUE exactly when at least one $\phi_i$ is TRUE.

To express that at least one of $\phi_1, \ldots, \phi_n$ is TRUE is a common enough requirement that the proposition $\neg\big(\neg\phi_1 \wedge \ldots \wedge \neg\phi_n\big)$ quickly becomes tiresome. Instead, we introduce **disjunction**. The disjunction $\phi_1 \vee \ldots \vee \phi_n$ is TRUE exactly when at least one $\phi_i$ is true. $\phi_1, \ldots, \phi_n$ are the **disjuncts** of the disjunction $\phi_1 \vee \ldots \vee \phi_n$. Disjunction does not add any expressive power to the language of the propositional calculus, provided that we already have negation and conjunction: Using the technique just described, we can construct for any disjunction $\phi_1 \vee \ldots \vee \phi_n$ a formula using just negation and conjunction which is true under exactly the same conditions. It is worth noting that while we have presented negation and conjunction first, and shown that disjunction can be expressed in terms of these, we could have just as well presented negation and disjunction first, and shown that conjunction can be expressed in terms of them. We leave it to the reader to show how a formula equivalent to $\phi_1 \wedge \ldots \wedge \phi_n$ can be expressed using just negation and disjunction. A disjunction $\phi \vee \psi$ may be read "$\phi$ or $\psi$," "either $\phi$ or $\psi$," "at least one of $\phi$ and $\psi$," and so on. However, note that while the English "or" can be used inclusively or exclusively, $\vee$ represents the inclusive disjunction: $\phi_1 \vee \ldots \vee \phi_n$ is true exactly when *at least* one $\phi_i$ is true, but more than one $\phi_i$ may be true. As with conjunctions, large disjunctions may be written using a summation-like notation; that is $\phi_1 \vee \ldots \vee \phi_n$ may be written as $\bigvee_{i=0}^{n} \phi_i$ or $\bigvee_{i=0,\ldots,n} \phi_i$.

The third Boolean connective we introduce is the **conditional**. The conditional is intended to formalize certain statements of the form "if … then …," where each "…" is another statement. For instance, from the statements "John is tall" and "Mary is John's mother" we may form the conditional "If John is tall then Mary is John's mother." We write the symbol $\rightarrow$ between the "if" part of the conditional, called the antecedent, and the "then" part of the conditional, called the consequent. The above conditional is rendered "$\big($John is tall$\big), \big($Mary is John's mother$\big) \rightarrow$ ." Its antecedent is "John is tall," and its consequent "Mary is John's mother." We said that the conditional captures only certain types of if-then statements — the conditional is a Boolean connective whose truth-value is determined solely by the truth-value of its antecedent and consequent. Specifically, a conditional is TRUE exactly when its antecedent is FALSE or its consequent is TRUE. If these conditions for TRUE seem strange, an equivalent definition is that a conditional is FALSE only in the case that its antecedent is TRUE and its consequent FALSE, and is TRUE in every other case.

As an example, consider the cases Alvin might cite to support his claim that his mother was wrong when she told him, "You will do well on the exam if you study hard."

- *Alvin studies hard, and does well on the exam.* This case does not seem to be inconsistent with the mother's claim, and is, in fact, certainly a case in which we want to affirm her conditional.
- *Alvin does not study hard, but does well on the exam.* His mother's claim may seem less relevant in this case, but it certainly isn't false. After all, she did not claim that Alvin would do well on the exam only if he studied hard. The conditional does not seem to be false, so we accept it as true in this case.
- *Alvin studies hard, but does not do well on the exam.* In this case, it is difficult to deny that Alvin's mother was wrong. Alvin did, in fact, study hard for the exam, and yet did not do well; his mother's if-then claim is refuted.
- *Alvin does not study hard, and does not do well on the exam.* Here we see nothing to invalidate his mother's claim, and so in this case as well we accept it.

We will consider other types of conditionals later, but within the propositional calculus we concern ourselves exclusively with the type just seen, which are known as **material conditionals**. A material conditional is false when both its antecedent is true and its consequent is false, and is true in every other case.

The final Boolean connective is the **biconditional**, which formalizes statements of the form "… if and only if …," and is written by placing a $\longleftrightarrow$ between the formulas standing in for each "…". As the English phrase "if and only if" suggests, a biconditional $\phi \longleftrightarrow \psi$ is equivalent to the conjunction of the two conditionals $\phi \rightarrow \psi$ and $\psi \rightarrow \phi$; that is, $\phi \longleftrightarrow \psi$ is equivalent to, or may be viewed as an abbreviation for, $\big(\phi \rightarrow \psi\big) \wedge \big(\psi \rightarrow \phi\big)$. Given this equivalence, the truth conditions for a biconditional are seen to be that a biconditional is TRUE if its constituent propositions are both TRUE or both FALSE, and is FALSE otherwise. Even more simply, a biconditional is TRUE exactly when its left and right hand sides have the same truth-value. Biconditionals are also variously known as **co-implications** and **equivalences**.

Until now we have used English to write specific declarative sentences, and English-like language to write specific declarative statements (with the two classes, as you will recall, being combined into the class of statements); and we have deployed Greek letters to refer to arbitrary formulas, and have combined these formulas with Boolean connectives to form even more formulas. But to determine whether a given sequence of symbols is, in fact, a formula of the propositional calculus requires being able to decide whether these smallest components (which are constructed from Boolean connectives) are themselves propositions. To decide whether "John is tall $\rightarrow \neg\big($Hoo fis gnaz$\big)$" is a formula, we must know whether "John is tall" and "Hoo fis gnaz" are formulas. For the sake of defining a formal system, which is, after all, our goal, we pusch these sorts of decisions away and legislate a set of symbols which are exactly our **atomic formulas**, that is, core formulas not constructed by using Boolean connectives.

| Syntax | Formula Type | Sample Representation |
|---|---|---|
| $\mathsf{P}, \mathsf{P}_1, \mathsf{P}_2, \mathsf{Q}, \mathsf{Q}_1, \ldots$ | Atomic Formulas | "Larry is lucky." as $\mathsf{L}_l$ |
| $\neg\phi$ | Negation | "Gary isn't lucky." as $\neg\mathsf{L}_g$ |
| $\phi_1 \wedge \ldots \wedge \phi_n$ | Conjunction | "Both Larry and Carl are lucky." as $\mathsf{L}_l \wedge \mathsf{L}_c$ |
| $\phi_1 \vee \ldots \vee \phi_n$ | Disjunction | "Either Billy is lucky or Alvin is." as $\mathsf{L}_b \vee \mathsf{L}_a$ |
| $\phi \rightarrow \psi$ | Conditional (Implication) | "If Ron is lucky, so is Frank." as $\mathsf{L}_r \rightarrow L_f$ |
| $\phi \leftrightarrow \psi$ | Biconditional (Coimplication) | "Tim is lucky if and only if Kim is." as $\mathsf{L}_t \leftrightarrow L_k$ |

Table 2.1: Syntax of the Propositional Calculus. Note that $\phi$, $\psi$, and $\phi_i$ stand for arbitrary formulas.

As our atomic formulas we take the majuscule Latin letters, optionally subscripted with a natural number; the subscripting ensures that we have an infinite supply of atomic formulas. For instance, $\mathsf{Q}$ and $\mathsf{P}_3$ are atomic formulas, also sometimes called **propositional variables**. Unlike formulas formed from Boolean connectives, the truth-values of atomic formulas are not determined by the truth values of other formulas, but will usually be taken to have TRUE or FALSE as values from the start of our use of them. For instance, we may say "Assume that $\mathsf{P}$ holds." or "Suppose that $\mathsf{Q}_3$ is not the case."

Note that the truth-value of a formula depends ultimately only upon the truth-values of the atomic formulas appearing in that sentence. Thus, if the truth-value of every atomic formula is fixed, we can mechanically determine the truth-value of a formula relative to that fixing. Such a fixing of truth-values to atomic formulas is called a **truth-value assignment**. A truth-value assignment is said to **satisfy** a formula $\phi$ exactly when $phi$ is true on that truth-value assignment; and a formula that is true on some truth-value assignment is **satisfiable**. A formula satisfied by every possible truth-value assignment is called a **tautology**, and a formula satisfied by no truth-value assignment a **self-contradiction** (and is declared **contradcitory**); every such formula, obviously, is **unsatisfiable**. If a sentence is satisfied by some truth assignments and unsatisfied by others, the sentence is said to be **contingent**.

We have now defined a syntax for formulas in our formal language for the propositional calculus by defining a set of atomic formulas and a number of Boolean connectives with which may be build other, more complex, formulas. (These complex formulae are sometimes called **molecular** formulae.) The syntax is summarized informally in Table 2.1, and as a formal grammar in Figure 2.4. As we have done already without mention, we may insert various brackets as necessary to avoid ambiguity in writing formulas. The truth conditions for the various Boolean connectives are shown in Table 2.2.

## 2.4 Slate Interlude: Inputing Formulas

Unless you've supererogatorily read ahead, you can't at the moment do all that much in Slate, since we have yet to discuss, in detail, the ways in which formulas can be linked by inferences to form arguments and proofs. You'll recall that in the previous Larry-Lucy example, we did catch a quick glimpse of two inference rules —

| **Formulas** | | **Compound Formulas** | | | | |
|---|---|---|---|---|---|---|
| $\phi$ | $\psi$ | $\neg\phi$ | $\phi \wedge \psi$ | $\phi \vee \psi$ | $\phi \rightarrow \psi$ | $\phi \longleftrightarrow \psi$ |
| **T** | **T** | **F** | **T** | **T** | **T** | **T** |
| **T** | **F** | **F** | **F** | **T** | **F** | **F** |
| **F** | **T** | **T** | **F** | **T** | **T** | **F** |
| **F** | **F** | **T** | **F** | **F** | **T** | **T** |

Table 2.2: Semantics of the Propositional Calculus. Though only binary conjunctions and disjunctions are shown here, in general a conjunction is TRUE when all of its conjuncts are TRUE, and a disjunction when at least one of its disjuncts is TRUE. **T** and **F** abbreviate "TRUE" and "FALSE," respectively.

Figure 2.4: Grammar of Propositional Calculus Presented as a Formal Grammar. (Note that this grammar doesn't permit the iteration of the same connective in order to make conjunctions/disjunctions with multiple conjuncts/disjuncts. How would you adjust the grammar here to permit this possibility, which is explicitly allowed in the informal presentation of the language of the prop. calc. given in Table 2.1?)

| | | |
|---|---|---|
| *Formula* | $\Rightarrow$ | *AtomicFormula* |
| | \| | (*Formula Connective Formula*) |
| | \| | $\neg$ *Formula* |
| *AtomicFormula* | $\Rightarrow$ | $\mathsf{P}_1 \mid \mathsf{P}_2 \mid \mathsf{P}_3 \mid \ldots$ |
| *Connective* | $\Rightarrow$ | $\wedge \mid \vee \mid \rightarrow \mid \longleftrightarrow \mid$ |

but 'glimpse' is indeed the operative word here. Details regarding inference rules in the propositional calculus will soon be provided (in the next section, 2.5). But before covering inference rules, we should take a bit of time to explain how to input propositions to a Slate workspace. In the previous Larry-Lucy example, of course, we had ahead of time stocked the workspace that was displayed; the same thing holds for the first Larry-Lucy example, shown in the Preface: we set up the files the screenshots of which were shown to you in that chapter.

Until now, we have used mathematical symbols such as $\rightarrow$, $\wedge$, and $\neg$, for writing formulas. The vast majority of keyboards lack keys for these symbols, and alternative methods of entering them directly (e.g., by two or three keystrokes in sequence for one symbol) can be cumbersome. We have also, to this point, omitted parentheses and brackets when possible, but we have still required that all our sentences be unambiguous. We have not resorted to defining any precedence order among the connectives. To illustrate, the arithmetic expression $1-2\times3+4$ is equivalent to $(1-(2\times3))+4$, since multiplication takes precedence over subtraction and addition, and operators with equal precedence, in this case $-$ and $+$, are performed leftmost first. At least at the moment, we have no similar rules to determine whether for

instance $\neg P \vee Q \wedge R$ should be interpreted as $\neg(P \vee (Q \wedge R))$, $(\neg P) \vee (Q \wedge R)$, or some other alternative.

Open the Slate application now, and proceed to open a new workspace of the type Propositional Calculus. You are presented with a fresh workspace: no nodes, no arcs, no content. Right-click on the workspace and select New Formula. (On OS X, hold down Control key and click on the workspace. Alternatively, on any system, choose New Formula from the Edit menu.) It is in the Formula field that we write propositions. To enter the proposition P, simply type P, and then press OK. (Don't worry about the Justification field for now.) You should now see something like the following in the workspace.

1. P
{1} **Assume** ✓

For the moment, don't worry about the numbers and text below the formula — the important part is that this new element in the workspace contains the formula P. In a similar way you can enter the formula Q. While we've been using individual majuscule Roman letters in the text for atomic formulas, any sequence of letters will serve in Slate. Try entering `JoeIsTall` as an atomic formula. Note that Slate is case sensitive, so the formula `WeWalkedTogether` is different from `WeWalkedToGetHer`.

Now, on to the Boolean connectives. In general, we enter formulas by writing `(`⟨connective⟩ `...)` where ⟨connective⟩ is replaced by `not`, `and`, `or`, `if`, or `iff`, and ... is replaced by an appropriate number of sentences. In particular, `not` must be followed by exactly one sentence, `if` and `iff` by exactly two, and `and` and `or` by two or more sentences. Try entering the sentence $P \rightarrow \neg(Q \vee \neg R)$ by creating a new formula and typing the text `(if P (not (or Q (not R))))`. If you still have P in the workspace from before, you should see something like this in the workspace:

2. $P \rightarrow \neg(Q \vee \neg R)$
{2} **Assume** ✓

In spite of the fact that the strings you key in to get atomic formulas into nodes are case-sensitive, the input forms of Boolean connectives are not. That is, `(not P)`, `(NOT P)`, and `(nOt P)` all produce the same sentence, $\neg P$. In this book, we will always write the Boolean connectives using miniscule letters, since we will most often use single majuscule letters (e.g., `P`, `Q`) for the names of atomic formulas. In general, a formula whose primary connective Π will be entered as `(`Π `...)`. The input syntax for writing sentences in Slate is summarized in Table 2.3.

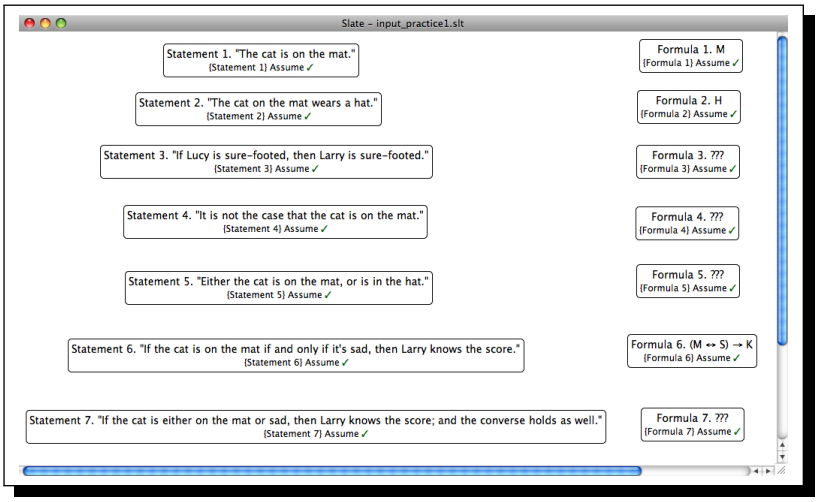
We have set up a Slate file for you to practice on at this point. Open the file `input_practice1.slt` now. What you see on your screen should closely resemble what's shown in Figure 2.5. We have stacked, on the left, a series of seven statements, expressed in English. On the right is a corresponding stack of seven formulas — but four of them are empty, and awaiting your input. Your task is to input formulas in each of these four cases that accurately express the statement with which they are paired. In order to make sure that you are on our wavelength, we supply here a list of the atomic formulas that should be used:

| **Sentence** | **Input Syntax** |
|---|---|
| $P, P_0, Q, Q_0, \ldots$ | Any sequence of letters |
| $\neg\phi$ | `(not` $\phi$`)` |
| $\phi_1 \wedge \ldots \wedge \phi_n$ | `(and` $\phi_1$ `...` $\phi_n$`)` |
| $\phi_1 \vee \ldots \vee \phi_n$ | `(or` $\phi_1$ `...` $\phi_n$`)` |
| $\phi \rightarrow \psi$ | `(if` $\phi$ $\psi$`)` |
| $\phi \leftrightarrow \psi$ | `(iff` $\phi$ $\psi$`)` |

Table 2.3: Syntax of the Propositional Calculus in Slate. Note that $\phi$, $\psi$, and $\phi_i$ stand for arbitrary propositions.

| **Statement** | **Atomic Sentence** |
|---|---|
| "The cat is on the mat." | M |
| "The cat is on the mat wears a hat." | H |
| "Lucy is sure-footed." | U |
| "Larry is sure-footed." | A |
| "The cat is in the hat." | I |
| "The cat is sad." | D |
| "Larry knows the score." | K |

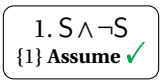
Figure 2.5: Practice Inputing Formulas




Have you finished inputing the data so that each of the statements now is symbolized by a corresponding formula? Good. In order to check you work, consult the Slate file `input_practice1_sol.slt`.

One final remark before proceeding to the next section. You have doubtless noticed that Slate gave a green check-mark to each and every node in the workspace you just finished using. Why is that? Why is Slate happy with the content in each

and every node, despite the fact that this content is thoroughly *ad hoc*? Does Slate for example agree that the cat is on the mat? The answer is due to the fact (which you should visually verify) that the rule of inference selected in each node is Assume. This rule gives you *a lot* of lattitude. Basically, you are allowed to assume any (grammatically correct) formula at any time. You could even assume something entirely absurd, and Slate would issue the green check-mark. To make sure we're right, let's represent the statement 'Llamas are sure-footed' by S, and then consider the patently absurd conjunction $S \wedge \neg S$, which expresses the proposition that llamas are sure-footed and are not sure-footed. You are encouraged to verify that Slate does indeed match the situation shown here:




Now try an experiment: Change (in the drop-down Justification menu available to you for editing the node you have built) the justification from Assume to PC $\vdash$, and then click OK. Now the green check-mark has been supplanted with ✗. By doing this, you switched from telling Slate that you simply wish to assume both S and $\wedge$S, to telling Slate that you claim this conjunction to be provable in the propositional calculus. That claim, of course, is wrong. We now turn to coverage of inference rules (including the one you just selected) and the proofs they make possible.

## 2.5 Inferences

We've seen that for every formula $\phi$, either $\phi$ or its negation, $\neg\phi$, is TRUE, and thus $\phi \vee \neg\phi$ must always be TRUE. For a particular sentence, say $P \rightarrow \neg Q$, we can use the tabular notation seen in Table 2.2 to justify the claim that $(P \rightarrow \neg Q) \vee \neg(P \rightarrow \neg Q)$ must always be TRUE. Since every truth-value assignment must make each of P and Q either TRUE or FALSE, there are four possible combinations that a truth-value assignment might dictate for P and Q. We can tabulate these, as well as the other constituent formulas, as follows.

| P | Q | $\neg Q$ | $P \rightarrow \neg Q$ | $\neg(P \rightarrow \neg Q)$ | $(P \rightarrow \neg Q) \vee \neg(P \rightarrow \neg Q)$ |
|---|---|---|---|---|---|
| **T** | **T** | **F** | **F** | **T** | **T** |
| **T** | **F** | **T** | **T** | **F** | **T** |
| **F** | **T** | **F** | **T** | **F** | **T** |
| **F** | **F** | **T** | **T** | **F** | **T** |

(2.1)

Now we reason as follows: Every truth-value assignment must make one of these four assignments to P and Q; that's a clear starting point. But in each case, $(P \rightarrow \neg Q) \vee \neg(P \rightarrow \neg Q)$ is satisfied. It thus follows that every truth-assignment must satisfy $(P \rightarrow \neg Q) \vee \neg(P \rightarrow \neg Q)$, and thus this formula is a tautology.

We can use the same approach to show that $P \rightarrow (Q \rightarrow R)$ is a contingent formula, and that $P \leftrightarrow (\neg P \wedge R)$ is contradictory. However, if we attempt the first of these, we notice that we need more than the four rows used above; we need eight altogether. This number is no coincidence: there are three atomic formulas, viz.,

P, Q, and R, in the compound formula, and $2^3 = 8$ rows are needed to enumerate the possible combinations of their truth-values. Thus, while the tabular technique can, in principle, determine whether any formula is tautologous, contingent, or contradictory, physical limitations quickly make it impossible to apply in practice. We shall soon develop practicable ways of discovering and confirming that these properties hold of formulas.

Whether there exist satisfying truth-value assignments for a particular formula is an issue that involves only a single formula, but there are interesting relationships for multiple formulas based on the relevant truth-value assignments. For instance, we may consider whether an arbitrary truth-value assignment that satisfies a formula $\phi$ must also satisfy a formula $\psi$. And we might ask whether there are any truth-value assignments satisfying a *collection* of formulas, or whether the set of truth-value assignments satisfying a formula is exactly the same as those satisfying another formula. We shall focus primarily on one particularly useful relationship.

A formula $\psi$ is said to be a **consequence** of a set $\Phi$ of formulas if and only if every truth-value assignment that satisfies every formula in $\Phi$ also satisfies $\psi$. This relationship, a positively key one in in the propositional calculus and beyond, is written compactly as:

$$\Phi \models \psi.$$

In the case that a formula $\psi$ is satisfied by every truth-value assignment, that is, that $\psi$ is a tautology, we may abbreviate even further to just $\models \psi$.

As an example, the formula Q is a consequence of the set $\{P, (P \rightarrow Q)\}$. We can make this argument in prose, saying, "Suppose a truth-value assignment satisfies P and $P \rightarrow Q$. Then the truth-value assignment must make P TRUE, and also $P \rightarrow Q$ TRUE. But $P \rightarrow Q$ is TRUE only if P is FALSE or Q is TRUE. Since P is not FALSE, Q must be TRUE; and so Q is satisfied. Thus any truth-value assignment satisfying P and $P \rightarrow Q$ must also satisfy Q." Alternatively, we might use a tabular approach.

| P | Q | $P \rightarrow Q$ |
|---|---|---|
| **T** | **T** | **T** |
| **T** | **F** | **F** |
| **F** | **T** | **T** |
| **F** | **F** | **T** |

(2.2)

Let's check whether Q is TRUE in every row where P and $P \rightarrow Q$ are TRUE. There is only one such row satisfying both P and $P \rightarrow Q$: the first; and in it Q is also satisfied. Thus we conclude that every truth-value assignment satisfying both P and $P \rightarrow Q$ must also satisfy Q. Hence Q is a consequence of P and $P \rightarrow Q$.

As a second example, perhaps a more interesting one that may catch at least some readers by surprise, is Q a consequence of P and $\neg P$? Again, we construct a

table and examine the rows in which all the premises are TRUE.

| P | Q | ¬P |
|---|---|---|
| **T** | **T** | **F** |
| **T** | **F** | **F** |
| **F** | **T** | **T** |
| **F** | **F** | **T** |

(2.3)

However, since the premises include P and ¬P, there are no rows in which all the premises are TRUE, and we therefore conclude that there are no truth-value assignments which satisfy both P and ¬P. Hence trivially, albeit perhaps strangely, all of the truth-value assignments that satisfy both P and ¬P also satisfy Q. Thus, Q is a consequence of P and ¬P. The choice of Q was of course arbitrary: *Every* formula is a consequence of P and ¬P. Generalizing even further, every formula is a consequence of an unsatisfiable set of formulas. And since in our formal-logic enterprise, as explained in the "Preliminaries" chapter (1), formulas represent statements, and statements express propositions, we can, in the light of what we have just discovered and proved, assert that any set of propositions (statements) that is unsatisfiable implies any proposition (statement).

The tabular approach quickly grows unmanageable as we work with formulas containing a greater number of atomic formulas within them. We would rather, then, turn to syntactic approaches that allow us to derive one formula, a conclusion, from a (possibly empty) set of formulas, called premises. For instance, though we worked with the specific formulas P, $\mathsf{P} \rightarrow \mathsf{Q}$, and Q above, analysis similar to the above prose shows that for any formulas $\phi$, $\phi \rightarrow \psi$, and $\psi$, a truth-value assignment satisfying $\phi$ and $\phi \rightarrow \psi$ must also satisfy $\psi$. We would be justified in creating a rule saying that "for any $\phi$ and $\psi$, if $\phi$ and $\phi \rightarrow \psi$ are satisfied, then $\psi$ is also satisfied."

What other rules can we sanction? Since a conjunction is satisfied only when all of its conjuncts are satisfied, we should accept "for any $i$ from 1 to $n$, if $\phi_1 \wedge \ldots \wedge \phi_n$ is satisfied, then $\phi_i$ is satisfied." Similarly, we should accept "if each of $\phi_1, \ldots, \phi_n$ is satisfied, then $\phi_1 \wedge \ldots \wedge \phi_n$ is satisfied."

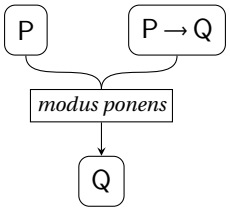
We could continue to develop these rules, but instead we turn to developing a general notion of proof, and with that done, we then return to similar, but slightly different, types of rules.

### 2.5.1 Proofs

Since we have decided against taking a direct semantic approach (i.e., one based on manipulating truth-values) to derivations, we must instead come up with a syntactic alternative. Though we have already been, in a very real sense, proving things every time we justify our reasoning, we now focus on *formal* proofs. We characterized the concept of a proof in the Introduction in general terms; here we more crisply define a formal proof as a directed hypergraph whose vertices contain formulas and, often, other relevant information. A hypergraph is simply a collection of vertices and a collection of hyperedges, each of which is a collection of vertices, one of which is designated as the conclusion of the edge, the rest being premises.

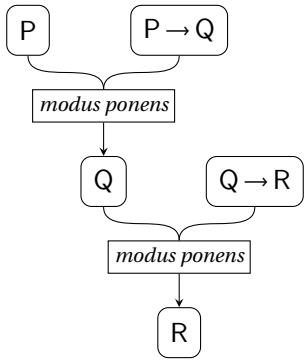
In the two earlier hands-on Slate interludes in this chapter you have already seen, and indeed manipulated, hypergraphs.

The preceding definition is clarified greatly by an example. What follows is a portion of a proof in which Q is inferred from P and P → Q by the rule *modus ponens*. The rule *modus ponens* sanctions inferring the consequent of a conditional from the conditional and its antecedent. The following is not a complete proof, since P and P → Q are not themselves justified; in a proper proof each vertex should be justified by some accepted inference rule.




(2.4)

Proofs typically contain more than one inference step, of course. Were Q → R also in our proof, we could add another *modus ponens* inference by which to conclude R from it and Q. This might look like the following.



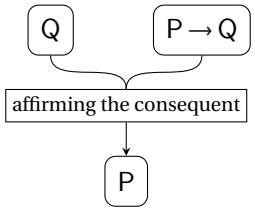

(2.5)

Of course, the Q → R could also appear directly to the right of P → Q rather than to the right of Q, or the whole network could point to the left or the right; the exact layout of the proof does not change the *graph structure* of the proof. Nonetheless, we will usually conform to the convention that the conclusion of an inference will be drawn below its premises, and we will often keep the premises of an inference aligned vertically. We strongly encourage you to follow this stylistic convention. For what it's worth, students in the classes through which Slate has been honed are in no uncertain terms required to abide by this convention — and they learned a lot!

The above are not complete proofs, due to the presence of unjustified vertices, so we still need some way of starting proofs. We also need to decide what set of rules we may use in our proofs.

Before developing our inference rules, we stress the point that they are *syntactic*, based therefore only upon the shape of the symbols that constitute premises and conclusion (and, perhaps, on some shape and structure of the surrounding proof). Although the rule *modus ponens* allows us to infer a consequence we discovered

earlier, namely that the consequent of any conditional is a consequence of the conditional and its antecedent, *modus ponens* is a syntactic rule allowing us to infer the consequent of a conditional from the conditional and its antecedent. We could, though it would be unwise, accept a rule called "affirming the consequent" by which we would infer $\phi$ from $\psi$ and $\phi \rightarrow \psi$. We could use the rule as shown in the following example. If we accept "affirming the consequent" as an inference rule, then the following example would conform to our system, although its conclusion is not a consequence of its premises.




(2.6)

### 2.5.2 Inference Rules

We concluded the prior section by pointing out that our inference rules are syntactic: whether an inference rule is properly applied is determined solely by the shape or structure of premises and conclusion, and possibly of the surrounding proof; *meaning* is irrelevant. We now develop a set of inference rules that capture deductive reasoning, wherein, as noted in the Introduction, the conclusion of an inference cannot be false when all of its premises are true. Were our aim to be able to express every argument ever made, we might include rules such as "affirming the consequent" (since the irrational among us are inclined to use such invalid rules), but since we aim to capture genuine deductive reasoning, and only such reasoning, we will firmly and irrevocabley reject "affirming the consequent."[1]

What rules should we have, then? If our goal is to capture deductive reasoning, accepting only conclusions which cannot be false when their premises are true, we might try to work just a single "consequence" rule: one that is properly applied when its conclusion is, in fact, a consequence of its premises. To check a "consequence" inference, we might use the tabular approach that we developed earlier. Unfortunately, the "consequence" rule has a shortcoming: it is not at all clear that we can check it syntactically. The upshot at this point is that we are trying to avoid the tabular approach, but we do not have a syntactic alternative for determining consequence.

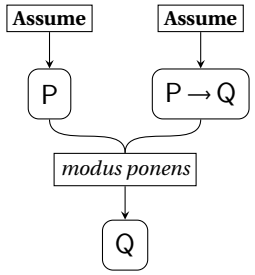
Instead, we follow one of the traditions in **natural-deduction style proofs systems**: for each of our Boolean connectives, we will have an introduction and an elimination rule. The introduction rule for a Boolean connective will provide a way to infer sentences whose primary connective is that connective. The elimination rule for a Boolean connective gives a way to remove, in some sense, a connective. These terms will become more clear as we develop our rules.

[1] We do not accept "affirming the consequent" because $\phi$ *can* be false when $\psi$ and $\phi \rightarrow \psi$ are true. Can you devise a specific example which shows this?

#### 2.5.2.1 Assumptions

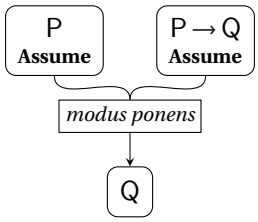
The examples we gave above were not proofs because there were unjustified vertices. Yet, continuing to suppose for the moment that we have *modus ponens* as an inference rule, if we want to show that Q follows from P and P → Q, *modus ponens* will justify Q, but what of P and P → Q? When we say that Q follows from the formulas P and P → Q, we are really saying that *were* we able to infer P and P → Q, we could infer Q from them.

When we want to work under the assumption of some sentence, we use the inference rule **Assume**, which of course we have already seen in our two previous Slate interludes. With **Assume** we can turn our earlier not-quite-proofs into genuine proofs.



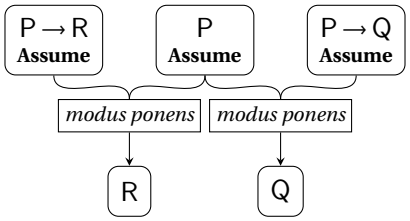

(2.7)

Since **Assume** should never have any premises, we will not bother drawing boxes and arrows for it, but from now on will simply label vertices that are justified by **Assume**. Thus our short proof becomes:




(2.8)

Later, we will have a bit more to say about the role that assumptions, that is, formulas justified with **Assume**, play in proofs, but for now it suffices to know that with **Assume** we can add vertices to proofs. We also note that conclusions may be proved relative to assumptions. For instance, in the following proof, R is proved relative to the assumptions P and P → R, while Q is proved relative to P and P → Q.




(2.9)

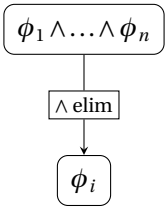
We now continue to assemble a set of inference rules to use in constructing proofs. A point on terminology: If there is a proof which concludes a formula $\phi$ whose in-scope assumptions are a set $\Phi$, we say that $\Phi$ **entails** $\phi$, and we will often write this relationship as

$$\Phi \vdash \phi.$$

. Later we shall have more to say concerning entailment.

#### 2.5.2.2 Conjunction Rules

As we said above, for each Boolean connective, we develop an introduction and an elimination rule. We begin with the conjunction. By **conjunction elimination** we may infer any of a conjunction's conjuncts from the conjunction. That is, given a conjunction $\phi_1 \wedge \ldots \wedge \phi_n$, we may infer any of the $\phi_i$. In writing proofs, we will use a symbolic abbreviation of "conjunction elimination," namely "$\wedge$ elim". This abbreviation is fairly standard in the literature, and some authors shorten this even further to "$\wedge$ E". Another name for conjunction elimination is **simplification**. The following diagram shows the general form of conjunction elimination.

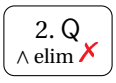



(2.10)

**2.5.2.2.1 Slate Interlude: First Proofs** Now that we have learned our first rule for operating on premises, let us turn back to Slate and see how to start constructing proofs. With conjunction elimination we should be able to infer, for instance, Q from P $\wedge$ Q $\wedge$ R. Open a (Propositional Calculus) Slate workspace and add P $\wedge$ Q $\wedge$ R as a new assumption; the input form for P $\wedge$ Q $\wedge$ R is `(and P Q R)`. You should now see the following in the workspace.


(2.11)

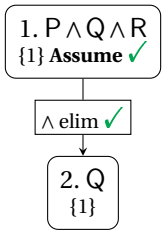
Now add Q to the workspace. The input format is simply `Q`, but be sure to select "$\wedge$ elim" as the justification. This should produce another vertex in the workspace like this:




(2.12)

The ✗ as you already know, indicates that the rule is not properly justified. This is not a surprise, of course: conjunction elimination needs a single premise, and our vertex does not yet have one. Click on the Q vertex to select it, and then hold the Alt

key and click on the $P \wedge Q \wedge R$ vertex.



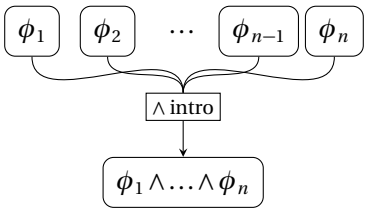

(2.13)

Congratulations! The vertices have been connected, the "$\wedge$ elim" label has moved to the box on the link connecting the vertices, and now has a ✓ since it is a proper application of the rule.

A few notes are in order.

- In general, when a vertex has no premises, its rule is written within the vertex itself, but when there are premises, the rule is written on the link connecting the vertex to its premises. Wherever the rule is written, it is followed by ✓ or ✗ depending on whether it is a correct application. When it is an incorrect application, letting the mouse hover over the vertex will give some information about what went wrong.
- When a vertex is selected, holding Alt and clicking on other vertices will add or remove them as premises of the vertex.
- The {1} in the vertices is still a mystery, but one which we will investigate shortly. You may also have noticed some red outlines on the vertices; this too will be made clear soon.

As conjunction elimination allowed us to infer from a conjunction a formula that is not necessarily a conjunction, **conjunction introduction** allows us to infer from a number of formulas their conjunction. Formally, if each of $\phi_i, \ldots, \phi_n$ are justified, then we may infer the conjunction $\phi_1 \wedge \ldots \wedge \phi_n$. Conjunction introduction is abbreviated "$\wedge$ intro".




(2.14)

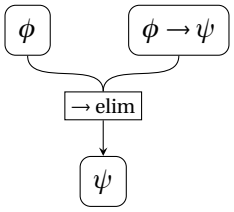
The two conjunction rules, as perhaps you'll agree, are fairly simple, but even with just the tools we have now, viz., **Assume**, conjunction elimination, and conjunction introduction, we can prove some useful results. For instance, we can already prove each of $(P \wedge Q) \wedge R$ and $P \wedge (R \wedge Q)$ follows from the other; this shows that conjunction is associative. We can also prove each of $P \wedge Q$ and $Q \wedge P$ from the other; this demonstrates the commutativity of conjunction.

**2.5.2.2.2 Slate Interlude: Conjunction-Rule Diagnostics** Try using **Assume** and the two conjunction rules to prove the following theorems in Slate. Notice the file-names: we have pre-configured the files for you with your goal in each case.

- $\mathsf{P} \wedge \mathsf{Q} \vdash \mathsf{Q} \wedge \mathsf{P}$. Filename: `PandQ_from_QandP.slt`.
- $(\mathsf{P} \wedge \mathsf{Q}) \wedge \mathsf{R}$ from $\mathsf{P} \wedge (\mathsf{R} \wedge \mathsf{Q})$.

### 2.5.2.3 Conditional Rules

Now that we have a feel for how inference rules work, we move onto rules for the conditional. The conditional elimination rule is straightforward; in fact, it's exactly the rule *modus ponens*. From a conditional and its antecedent, we may infer the conditional's consequent. We could continue to call the rule *modus ponens*, but for consistency with other rules, we will use the name 'conditional elimination,' or "$\rightarrow$ elim".




(2.15)

**2.5.2.3.1 Slate Interlude: Conditional Elimination Diagnostics**

- Try assuming $\mathsf{P} \rightarrow \mathsf{Q}$ and $\mathsf{P}$ and deriving $\mathsf{Q}$ using "$\rightarrow$ elim".
- Assume $\mathsf{P} \rightarrow (\mathsf{Q} \wedge \mathsf{R})$ and $(\mathsf{R} \wedge \mathsf{P}) \rightarrow \mathsf{S}$. Prove $\mathsf{Q} \wedge \mathsf{S}$.

Our rule of conditional introduction, abbreviated "$\rightarrow$ intro" is more complicated then conditional elimination, and is the first rule that depends on the structure of the surrounding proof. Before presenting the rule proper, let us consider exactly what conditional introduction must produce. It is by conditional introduction that we will infer conditionals, that is, if-then formulas. In informal reasoning, we typically reach conclusions of the form "if … then …" by temporarily supposing the "if" part of the sentence, drawing some conclusions from it, and finally reaching the "then" part of the sentence. Note well, however, that the ultimate conclusion, the "if … then …" sentence, does not depend on the temporary supposition of the "if" part of the sentence.

For instance, suppose we are presented with the algebraic expression $5x = y - 1$, and the question "is $y$ odd?" We do not have enough information to answer the question, but neither are we completely uninformed. We might respond, saying, "if $x$ is even then $y$ is odd." How would we justify a such a claim? Our reasoning could be along the lines of the following. "Suppose that $x$ is even. Then the product of 5 and $x$ is even. The sum of 1 and the product of $5x$ is $y$, and the sum of an even number and 1 is odd, and so $y$ is odd. Therefore, *if* $x$ is even, *then* $y$ is odd." Notice

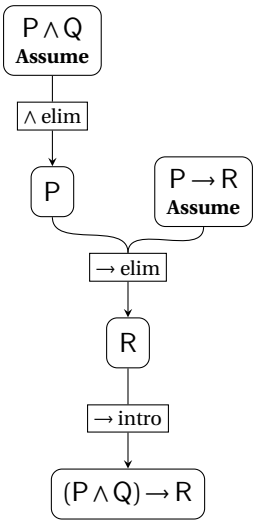
that at the end of our explanation we no longer are operating under the assumption that $x$ is even — this assumption is said to have been **discharged** when we concluded with the if-then formula. This pattern of reasoning, wherein we assume some proposition or statement and draw some intermediate conclusion, finally concluding the conditional whose antecedent is the assumption and whose consequent is the intermediate conclusion, is what we want conditional introduction to capture.

We already have a mechanism for making assumptions with our **Assume** rule, and know how to infer sentences from other sentences. The final ingredient is we need is a way to discharge assumptions. The basic idea of discharging assumptions is one that will be familiar to all readers. We gave an algebraic example in the previous paragraph, but the phenomenon is one you have seen outside of mathematics: Any time you have said something like "Let's assume for the sake of argument ..." you have prepared to later discharge this assumption. People find it quite irresistable to reason in this way. For example, how might one argue that it would be a bad idea to raise the speed limit on roadways in the United States to 200 mph? No doubt there are countless ways to try to show that this move would be unwise, but certainly one way is to reason as follows: "Suppose that we do indeed raise the limit to 200 mph across the land. Then plenty of people, human nature being what it is, and automotive technology being as powerful as it is, are quite literally going to drive, say, 150 mph. But think about that! We're not talking about race-car drivers here, and we're not talking about race tracks. It's inevitable that unprecedented carnage will ensue." This reasoning may or may not in fact consitute a sound, persuasive argument, but regardless, one thing is plain: If this argument is successful, it is because it establishes that if the limit is so raised, carnage will result. It's *not* successful because it establishes that the limit will in fact be raised. The assumption that the limit is raised is only temporarily made in order to get the hearer to confront a rather distasteful state-of-affairs that would ensue. The assumption itself is "discharged" once the main point is made. The kind of reasoning is rendered precise and mechanical in Slate.

To make this clearer, and more formal, let's look at a short proof, and examine the set of assumptions relative to which each vertex in the proof is derived. This is also our first proof in which conditional introduction appears. Conditional introduction involves the discharging of an initial assumption made in order to launch the reasoning in question.

| Sentence | In-Scope Assumptions |
|---|---|
| $P \wedge Q$ | $P \wedge Q$ |
| $P$ | $P \wedge Q$ |
| $P \rightarrow R$ | $P \rightarrow R$ |
| $R$ | $P \wedge Q, P \rightarrow R$ |
| $(P \wedge Q) \rightarrow R$ | $P \rightarrow R$ |

Table 2.4: Sentences of the proof 2.16 and their in-scope assumptions.



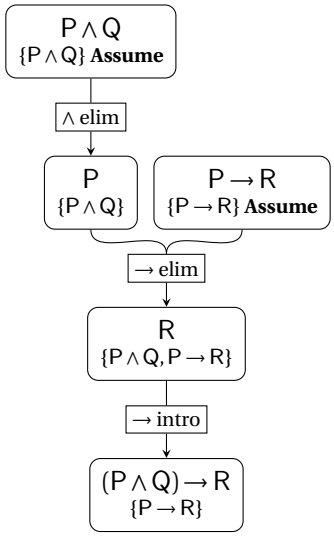

(2.16)

In this example, each formula has been proved relative to some set of assumptions. This set may be thought of as the set of assumptions on which each formula depends, or as those assumptions that still hold when a sentence is derived. For instance, the only assumption on which $P \wedge Q$ depends is itself. $P$ does not introduce any new assumptions, nor does it eliminate any, and so the assumptions that it depends on are simply those on which its premise depends; that is, $P \wedge Q$. $R$, with two premises, is an interesting case. $R$ does not introduce any assumptions, nor does it eliminate any. Where $P$ inherited, so to speak, the assumptions of its premise, $R$ takes all of the assumptions of all (both) of its premises. $P$ depends only on $P \wedge Q$, and $P \rightarrow R$ only on itself, so $R$ depends on $P \wedge Q$ and $P \rightarrow R$. $(P \wedge Q) \rightarrow R$ is the most important feature here, though. Since it is justified by $\rightarrow$ intro, it discharges an assumption. $\rightarrow$ intro's conclusion is a conditional, and the assumption it discharges is the antecedent of that conditional; that is, $P \wedge Q$. Thus, $(P \wedge Q) \rightarrow R$ depends only on the assumption $P \rightarrow R$. These observations are summarized in Table 2.4.

Let's make this all a bit more precise, and then we will continue to (finally) define conditional introduction. We call the set of assumptions on which a sentence depends, that is, the set of assumptions relative to which the formula is proved, its **in-scope assumptions**. In general, a formula's in-scope assumptions are all the

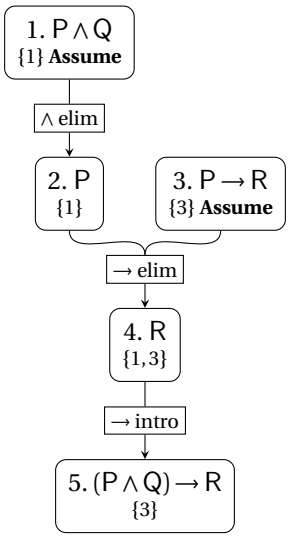
in-scope assumptions of the premises of the rule that justifies the sentence. The exception to this general rule is that some rules discharge assumptions, and so their conclusions have fewer in-scope assumptions than the general rule would dictate.

Determining the in-scope assumptions of a vertex in a proof manually can be an error-prone process: the general rule is easy enough to grasp, but the rules that discharge assumptions each have their own specific patterns, and it's easy to make a mistake. To ensure accuracy, we can keep track of the in-scope assumptions for each vertex by inscribing them within braces (i.e., "" and ""). For instance, our example now becomes the following.




(2.17)

Alas, this too is cumbersome, even if less error prone. One more trick will give us a manageable system: Rather than writing the formulas for in-scope assumptions, we will assign each vertex a numeric identifier, and will just annotate nodes with the identifiers of their in-scope assumptions. Using this technique, we finally achieve the following more perspicuous approach.



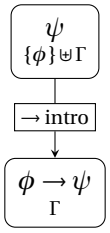

(2.18)

These techniques, of course, are only for our convenience. They help us keep track of the assumptions that we have made and discharged. When we are constructing proofs by hand, we may choose whether or not to use these techniques. Whether we use them or not, we still must pay careful attention to rules that discharge assumptions. In this textbook, we will sometimes omit these annotations when it is convenient to do so. In describing rules, especially those which discharge assumptions, we will often use the sets-of-formulae annotation.

*We finally know what the brace-enclosed number that have been appearing in vertices mean! They are keeping track of the in-scope assumptions of vertices. Slate knows about the general rule for determining in-scope assumptions as well as the special cases. As one more convenience, when a vertex is selected, all of its in-scope assumptions are drawn with red double-outline — this makes it a little bit easier to find them in a crowded workspace. Now we know about almost everything that we might see in a vertex. (In Chapter 5, there will be one more addition, but we don't need to worry about that yet.)*

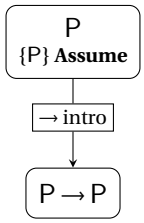
Finally, we can move on to conditional introduction! Conditional introduction allows us to infer the conditional $\phi \to \psi$ from a proposition $\psi$ whose in-scope assumptions include $\phi$, where the inferred conditional no longer has $\phi$ as an in-scope assumption. (The disjoint union, denoted by $\uplus$, indicates that the sets of the

union are disjoint. In this case, it ensures that $\phi \notin \Gamma$.)




(2.19)

Since conditional introduction allows for the discharge of assumptions, its conclusion may have no assumptions at all. A vertex in a proof that has no in-scope assumptions, assuming that the proof is, in fact, correctly formed, is called a **theorem**. For instance, the sentence $\mathsf{P} \rightarrow \mathsf{P}$ is a theorem established by the following proof. (When a vertex has no in-scope assumptions, we won't bother to write the symbol for the empty set.)



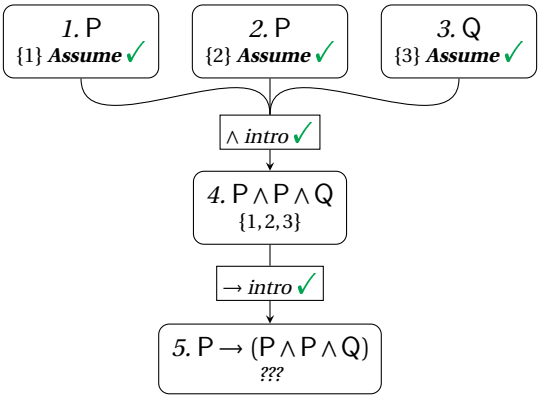

(2.20)

*Prove the following theorem in a file that you create, naming it* `cond_elim_practice.slt`*. (Note that you can deduce from the previous sentence, and our trustworthiness, that one file can hold multiple proofs. We are asking you to create a file that hold three proofs; or put in terms of the workspace (which of course here again will be of the type* Propositional Calculus*, we are asking that your workspace, once you are done, shows three separate proofs/graphs.)*

- $\mathsf{P} \rightarrow \mathsf{P}$
- $(\mathsf{P} \wedge \mathsf{Q}) \rightarrow \mathsf{P}$
- $\mathsf{P} \rightarrow (\mathsf{Q} \rightarrow \mathsf{P})$ *(This one is a bit harder. Here's a motivating question: If you assume* P *and you assume* Q*, how can you get a vertex for* P *with both* P *and* Q *as in-scope assumptions?)*

***A note on the difference between keeping track of in-scope assumptions as sets of formulae and as sets of numeric identifiers:*** *(This note can probably be skipped on the first reading.) When we described conditional introduction, we treated the in-scope assumptions as sets of formulae, not as sets of vertices. Most of the time there will not be any noticeable difference. However, we are not prohibited in Slate from assuming the same formula more than once. For instance, the*

*following is a correct Slate proof, but we have left out the in-scope assumptions of the final conclusion. What should the in-scope assumptions of the fifth vertex be?*



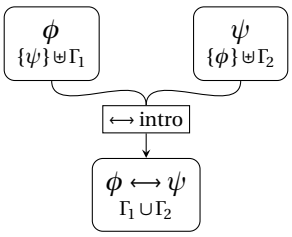

(2.21)

*In fact, conditional introduction discharges all of the assumptions whose formula is the antecedent of its conclusion; the set of in-scope assumptions of the fifth vertex is simply* {3}*. This behavior does not come up often, and is unlikely to cause you any trouble, but it is good to understand what conditional introduction (and other rules) are doing, and why they might look like they are discharging more assumptions than they ought to be.*

#### 2.5.2.4 Biconditional Rules

A biconditional $\phi \longleftrightarrow \psi$ is the conjunction of the two conditionals $\phi \rightarrow \psi$ and $\psi \rightarrow \phi$, and so now that we have all of our conjunction and conditional rules, we should be able to determine what biconditional introduction and elimination do.

To infer a conditional, we infer the conditional's consequent with the conditional's antecedent as an in-scope assumption. For instance, to infer $\mathsf{P} \rightarrow \mathsf{Q}$, we derive $\mathsf{Q}$ with an in-scope assumption $\mathsf{P}$. As a biconditional is rather like two conditionals; biconditional introduction is rather like two applications of conditional introduction. In particular, to infer a biconditional $\mathsf{P} \longleftrightarrow \mathsf{Q}$, we need to infer $\mathsf{P}$ in the scope of an assumption $\mathsf{Q}$, and to infer $\mathsf{Q}$ in the scope of an assumption $\mathsf{P}$. Schematically, biconditional introduction looks like this:

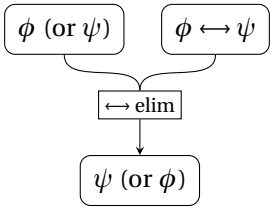



(2.22)

There may be overlap between $\Gamma_1$ and $\Gamma_2$, and there are no restrictions on what may appear in them. Of particular note, $\phi$ might appear in $\Gamma_1$ or $\psi$ might appear in $\Gamma_2$.

*Use biconditional introduction to prove* $(\mathsf{P} \wedge \mathsf{Q}) \longleftrightarrow (\mathsf{Q} \wedge \mathsf{P})$*, saving the resultant file as* `first_bicond_intro.slt`*.*

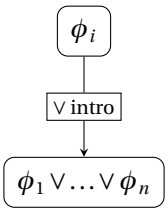
Unsurprisingly, biconditional elimination is very similar to conditional elimination, except that it can work in both directions, so to speak. It permits inferring one side of a biconditional from the biconditional and its other side.




(2.23)

#### 2.5.2.5 Disjunction Rules

We now turn to the inference rules for disjunction. The introduction rule for disjunction is very simple. From a formula we may infer any disjunction, as long as one of that disjunction's disjuncts is that original formula. That is, we may infer the disjunction $\phi_1 \vee \ldots \vee \phi_n$ from any of the disjuncts $\phi_i$.




(2.24)

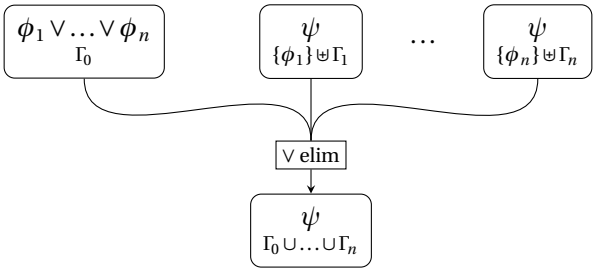
This may at first seem like a strange rule. Can we imagine ourselves saying, "it is raining outside, therefore it is either raining outside or it is sunny outside"? Probably not. Yet, when we consider under what conditions that disjunction is true, one of them is that it is raining outside, and so the inference, while odd, is perfectly valid. Often in informal arguments the reasoning that corresponds to disjunction introduction remains implicit, just as conjunction elimination is often left unelaborated. Our goal, of course, is to make all our reasoning explicit, bringing each inference step to light. Only when we do this can we be certain that our reasoning (or for that matter the reasoning of a computing machine or robot) is valid.

*Use disjunction introduction and the earlier rules to prove* $(\mathsf{P} \vee \mathsf{Q}) \wedge \mathsf{P}$ *from* $\mathsf{P}$*. Save your file as* `first_disjunction_intro.slt`*.*

Disjunction elimination is more complex than disjunction introduction, but captures a much more interesting aspect of reasoning: **proof by cases**. For instance, if we know that Susan has class on Monday or has class on Tuesday, we may conclude that Susan has class on a weekday, for in the first case, Susan has class on Monday,

and Monday is a weekday; and in the second case, Susan has class on Tuesday, and Tuesday is also a weekday. Not a very exciting example, but many of the greatest achievements in logic and mathematics (and the formal sciences, generally) make pivotal use of proof by cases.

In general, if we have a disjunction $\phi_1 \vee \ldots \vee \phi_n$ and can show that some $\psi$ follows from each $\phi_i$, then we may conclude $\psi$. That is, if we can, for each $\phi_i$, assume $\phi_i$ and show that $\psi$ follows, then we may conclude $\psi$ from the disjunction $\phi_1 \vee \ldots \vee \phi_n$ and the derivations of $\psi$. There is one more subtle point, however. In the days-of-the-week example above, the conclusion that Susan has class on a weekday should not be in the scope of both the assumptions that she has class on Monday and that she has class on Tuesday; these assumptions are *discharged*. Disjunction elimination discharges each assumption $\phi_i$ from the line of reasoning that corresponds to that case.

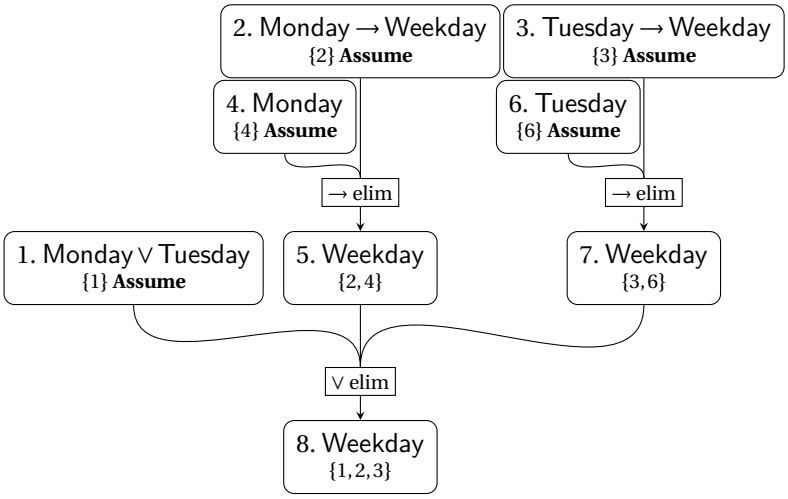



(2.25)

The various $\Gamma_i$ on the premises of disjunction elimination might make this rule seem more complicated than it really is. Their presence makes it clear that the only assumptions discharged from each line of reasoning is the assumption corresponding to that particular case.

Since this is a somewhat complex rule, let us work out the example involving Susan. We will use Monday, Tuesday, and Weekday, to stand for the formulas that represent the statements "Susan has class on Monday," "Susan has class on Tuesday," and so on. Our premises, then, are Monday $\vee$ Tuesday, Monday $\rightarrow$ Weekday, and Tuesday $\rightarrow$ Weekday. And now here's how the formal proof that parallels our above

informal reasoning runs:




(2.26)

Notice, in particular, that the conclusion, 8, is in the scope of assumptions 1, 2, and 3, but neither 4 nor 6, since these have been discharged.

#### 2.5.2.6 Negation Rules

The final pair of rules that we introduce deal with negation. For the other connectives, the introduction and elimination rules may have seemed a bit unbalanced. The disjunction rules are a significant example: disjunction introduction is trivial, while disjunction elimination is a rather complicated (but powerful) assumption-discharging rule. The other rules also dealt with connectives that linked two or more formulas; for instance, conditional elimination cites a conditional and an instance of its antecedent.

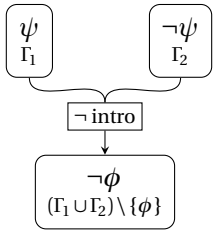
The negation rules require a slightly different approach. The only sub-formula of $\neg\phi$ is $\phi$, and clearly neither follows from the other. What, then, are the forms of our negation rules? The key is in the pattern known classically as *reductio ad absurdum*, reduction to the absurd, and today as **proof by contradiction**. With proof by contradiction, we make some assumption, show that it leads to a contradiction, and then conclude its opposite. Of course, we have not defined what "opposite" means, but we can immediately study the skeletal form of two lines of deductive reasoning that (whether you remember it or not) you have seen before:

(i) Suppose it is the case that $\phi$. ... But this has lead us to a contradiction! Therefore $\phi$ is not the case.

(ii) Suppose it is not the case that $\phi$. ... But this has led us to a contradiction! Therefore $\phi$.

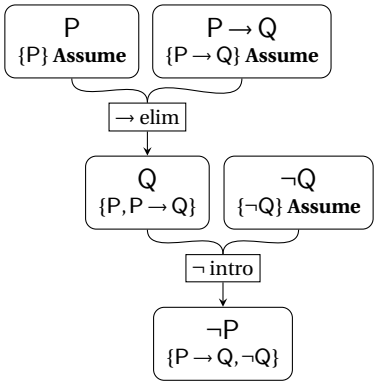
In the first line, if the assumption of $\phi$ leads us to a contradiction, we may conclude $\neg\phi$, discharging the assumption $\phi$. Similarly, if the assumption $\neg\phi$ leads

us to a contradiction, we may conclude $\phi$, discharging the assumption $\neg\phi$. The first pattern is what we take as negation introduction, and the second, negation elimination. But what exactly counts as a contradiction? We infer a contradiction by proving both some formula $\psi$ and its negation $\neg\psi$. Thus negation introduction has the following form, where at least one of $\Gamma_1$ and $\Gamma_2$ contains $\phi$.




(2.27)

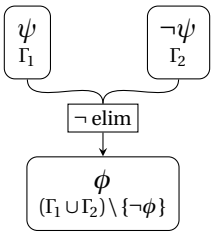
For instance, from $\mathsf{P} \to \mathsf{Q}$ and $\neg\mathsf{Q}$ we may conclude $\neg\mathsf{P}$ using negation introduction, by deriving $\mathsf{Q}$ in the scope of an assumption $\mathsf{P}$, but $\neg\mathsf{Q}$ is never under the scope of $\mathsf{P}$.




(2.28)

Notice that while we concluded $\neg\mathsf{P}$ under the scope of $\mathsf{P} \to \mathsf{Q}$ and $\neg\mathsf{Q}$, we could have just as well concluded $\neg(\mathsf{P} \to \mathsf{Q})$ under $\mathsf{P}$ and $\neg\mathsf{Q}$, or $\neg\neg\mathsf{Q}$ under $\mathsf{P}$ and $\mathsf{P} \to \mathsf{Q}$. The three formulas $\mathsf{P}$, $\mathsf{P} \to \mathsf{Q}$, and $\neg\mathsf{Q}$ cannot all be true at once, and negation introduction gives us the power to infer the negation of any one of them from the other two.

Negation elimination is almost exactly the same as negation introduction. The only difference is that while with negation introduction we discharge some assumption $\phi$ to conclude $\neg\phi$, with negation elimination we discharge some assumption $\neg\phi$ to conclude $\phi$. Similar constraints apply regarding in-scope assumptions: that is, $\neg\phi$ must be in at least one of $\Gamma_1$ and $\Gamma_2$.




(2.29)

- *Use the negation rules and biconditional introduction to prove the theorem* $\neg\neg P \longleftrightarrow P$*, saving the resultant file as* `notnotP_implies_P.slt`*.*
- *Try proving De Morgan's laws:*
  - $\neg(P \wedge Q) \longleftrightarrow (\neg P \vee \neg Q)$
  - $\neg(P \vee Q) \longleftrightarrow (\neg P \wedge \neg Q)$

### 2.5.3 The PC "Oracles"

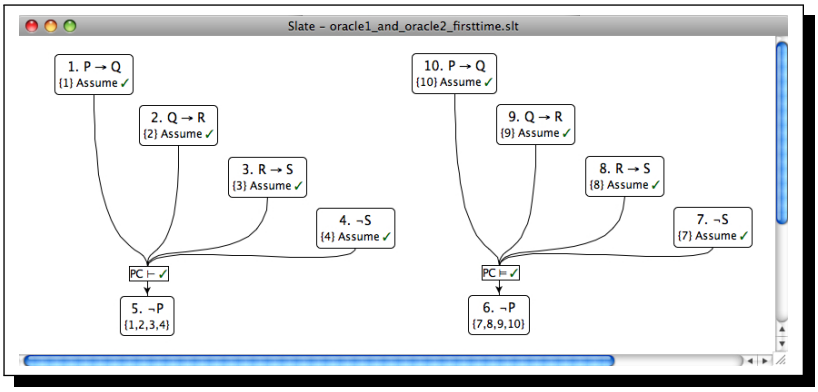
Some dilligent and observant students will at this point be wondering about two options under Slate's Justification menu that we have yet to discuss: PC $\vdash$ and PC $\models$. (If you haven't spotted them, take a look now before proceeding so you know what we're referring to.) Well, while it's true that we haven't explicitly discussed this pair, we have discussed consequence, which as we now know is captured by $\models$, and we've also discussed entailment, which as we now know is captured by $\vdash$. As you can probably deduce, then, these two options in Slate pertain to consequence and entailment. But what, specifically, do they allw you to do as pilot of Slate? They allow you to call either of two "oracles" — or at least that's how you can think of them. The first oracle will give you an answer when you ask it whether some set $\Phi$ of formulas entails a given formula $\phi$. The second oracle will give you an answer when you ask it whether some formula $\phi$ is a consequence of a set $\Phi$ of formulas. Being oracles, both are infallible. If you're skeptical, give these two equations a shot in Slate, using the two oracles:

$$\Phi = \{P \rightarrow Q, Q \rightarrow R, R \rightarrow S, \neg S\} \vdash \neg P$$

$$\Phi = \{P \rightarrow Q, Q \rightarrow R, R \rightarrow S, \neg S\} \models \neg P$$

What happened? Hopefully you witnessed a "Yes" returned by each oracle. Well, not exactly. What you saw returned *as* a "Yes" was the welcome ✓, with which you are now quite familiar. If you didn't see this happen on your screen, you did something wrong, and you should compare your work with what is shown in Figure 2.6. If you're still having trouble getting the desired results, open the file `oracle1_and_oracle2_firsttime.slt` and inspect it until any mystery in mind has evaporated. After that, for good measure, change the individual formula $\neg P$ to the right of $\vdash$ ($\models$) and see what the two oracles say. Now a "No" (i.e., a ✗) is the answer given.

Figure 2.6: First Use of the Two PC Oracles

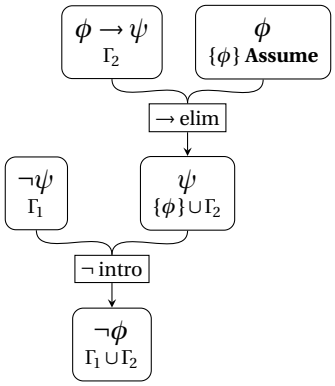



## 2.5.4 Derived Inference Rules

The inference rules that we have presented constitute a powerful set of rules for the propositional calculus. There is a pleasing balance between a small set of rules (just **Assume** and two per boolean connective) and enough expressiveness to capture the structures of argument commonly used in both informal and formal reasoning. Even so, there are patterns of reasoning that are not captured directly by any of our inference rules, though they are often thought of as single steps. We now consider a few examples, and show derived inference rules that capture these common patterns.

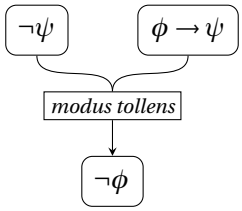
### 2.5.4.1 Modus Tollens

The rule of *modus tollens* is used to infer the negation of a conditional's antecedent from the conditional and the negation of its consequent. For instance, from "Bill is not wet" and "if Bill is standing in the rain, then he is wet" we conclude, using *modus tollens*, that "Bill is not standing in the rain." In general, from $\neg\psi$ and $\phi \rightarrow \psi$, we conclude $\neg\phi$.




(2.30)

We would really like to do this in just one step, moving from $\neg\psi$ and $\phi \rightarrow \psi$ to $\neg\phi$ with a single inference, *modus tollens*. The set of rules we assembled from **Assume** and the pairs for the boolean connectives we will take as a core set of inference rules. We will, in fact, allow patterns of reasoning, such as *modus tollens*, in proofs, but these we will call **derived** inference rules, since they can be derived from the core rules. Using *modus tollens* as a inference rule, we can simplify the example above.

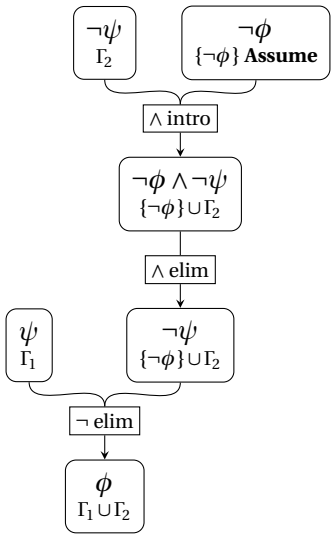



(2.31)

Some of the boolean rules had both conditional-citing and assumption-discharging variants. It should be clear that each variation could be viewed as a derived rule employing the other. For instance, the disjunction elimination that cites conditionals can be expressed using the disjunction elimination that discharges assumptions. We shall make use of this fact later, when we examine the relationship between entailment and consequence.

#### 2.5.4.2 Ex Falso Quodlibet

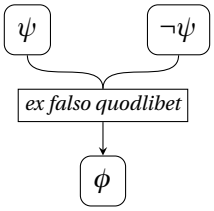
We saw that the disjunction elimination rule captures proof by cases. It is straightforward, for instance, to infer R from $P \rightarrow R$, $Q \rightarrow R$, and $P \vee Q$. (In fact, with the conditional citing form of disjunction elimination, this is just one step.) More difficult, however, would be to conclude P from $P \vee Q$ and $\neg Q$. Such a proof still makes use of disjunction elimination, but depends on being able to show that P holds in the case that P holds (trivial) and in the case that Q holds. In fact, we already have all the inference rules we need in order to complete this proof. Before showing a derived rule that will allow us to finish this inference, recall that in our discussion of consequence, we noted that every proposition follows from an unsatisfiable set of premises. The rule we develop now captures this fact. First, the pattern that we will abbreviate:




(2.32)

(We presume that $\neg\phi \notin \Gamma_1$ and $\neg\phi \notin \Gamma_2$. Were $\neg\phi$ in either $\Gamma_1$ or $\Gamma_2$, the final negation elimination would be sufficient.)

Notice the use of conjunction introduction and conjunction elimination for the purpose of deriving $\neg\psi$ within the scope of an assumption $\phi$. This can be a useful tactic when working with rules that discharge assumptions, and, though we do not develop it now, is a candidate for a useful derived inference rule. The abbreviated form of our "anything from a contradiction" rule, which follow the standard in-scope assumption pattern, follows. We will use the rule in the following section where we finally derive $\mathsf{P}$ from $\mathsf{P} \vee \mathsf{Q}$ and $\neg\mathsf{Q}$.

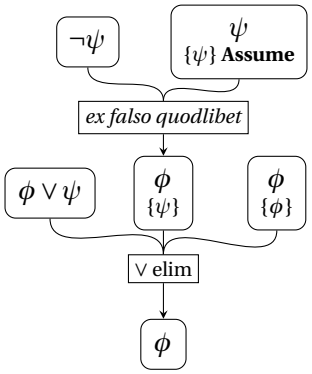



(2.33)

#### 2.5.4.3 Disjunctive Syllogism

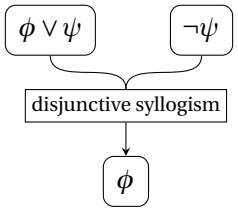
The rule of disjunctive syllogism allows us to conclude $\phi$ from $\phi \vee \psi$ and $\neg\psi$. The intuition behind disjunctive syllogism is clear; if (at least) one of two possible cases must hold, and one is known not to hold, then the other must. Using our tabular method can also easily confirm that $\phi$ is a consequence of $\phi \vee \psi$ and $\neg\psi$. However, to show that $\phi$ is entailed by $\phi \vee \psi$ and $\neg\psi$ is a bit trickier. An intuitive argument might run as follows. "One of the $\phi$ and $\psi$ must be the case. In the case that $\phi$ holds, then, obviously, clearly $\phi$. It cannot be the case that $\psi$ holds, as we know $\neg\psi$. Therefore $\phi$."

We do not have to a way to express in our proofs that a case cannot hold. But what do we really mean when we say that a case cannot hold? We mean precisely that if that case did in fact hold, we would observe a contradiction. This is a prime candidate for the derived rule we just defined. We revise our intuitive argument, thus: "One of $\phi$ and $\psi$ must be the case. In the case that $\phi$ holds, we obviously have $\phi$. In the case that $\psi$ holds, there is a contradiction, as we know $\neg\psi$. Since anything follows from a contradiction, $\phi$ holds in the case that $\psi$ holds. Therefore, in both cases $\phi$ holds, and we conclude $\phi$."




(2.34)

This pattern is abbreviated as disjunctive syllogism.




(2.35)

## 2.6 Truth Trees (*not* Truth Tables)

You will have observed that we have repeatedly made use of tables containing the truth-values **F** and **T** in formulae. We now specifically bring to your attention again (recall the beginning of §2.5) that such tables, so-called **truth tables**, in conjunction with some simple algorithms, can be used to ascertain whether a given formula $\phi$ in the propositional calculus is a validity, and whether a given consequence claim $\Phi \models \phi$ (or equivalently, whether a given entailment claim $\Phi \vdash \phi$) holds.[2] For the first case, here is an algorithm (**TABLE**) for deciding whether $\models \phi$ holds for a given $\phi$, an algorithm we alluded to earlier in passing:

[2] This equivalence is a meta-theoretical fact about the propositional calculus that isn't established until Chapter 7. To be clear, the fact consists in this:

$$\Phi \vdash \phi \text{ if and only if } \Phi \models \phi.$$

1. Build and completely fill in a truth-table for $\phi$.
2. Check if the column for the main connective of $\phi$ contains only the truth-value **T**.
   - If so, declare $\phi$ to be a validity; otherwise, declare that it's not a validity.

This is a very inefficient algorithm. Suppose that $\phi$ has four atomic formulae within it. How many rows would be in the truth-table in this case, at the conclusion of running **TABLE** on $\phi$? ... The answer is: $2^4 = 16$. Do you see the underlying function, and how inefficient it is? Where $n$ is the number of atomic formulae in a given $\phi$ to be checked for validity via **TABLE**, the number of rows required in the truth-table is

$$2^n$$

and hence the number of rows grows exponentially on the number of atomic formulae provided as input. Such exponential growth is a hallmark of computational intractability.[3] It's very easy to concretely verify the intractability in question: run **TABLE** by hand now on the formula

$$[\neg(\mathsf{P} \rightarrow \mathsf{Q}) \wedge \neg(\mathsf{Q} \rightarrow \mathsf{R})] \rightarrow \mathsf{S}$$

If you choose to verify this way, make sure to set aside plenty of time before you begin.

There is a much better way: **truth-trees**! As an added bonus, such trees can be efficiently and reliably built, and verified, in Slate, as we now explain.

The trees we are talking about are upside down relative to the real-world trees you're familiar with: the roots of truth-trees are at the top, their trunks grow from these roots going downward, and when branches emerge they also reach downward. Let's build a simple truth-tree in order to explain this basic geometry by way of an example, a truth-tree that confrms

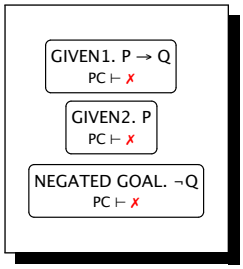
$$\{\mathsf{P} \rightarrow \mathsf{Q}, \mathsf{P}\} \vdash \mathsf{Q}.$$

This is of course *modus ponens*. In order to build our tree, we first build the trunk out of the formulae that are given; in this case we have a pair. In addition, to complete the trunk, we include the *negation* of what is supposed to be provable from the givens. Figure 2.7 shows the trunk that we have thus far. Don't be disturbed by the fact that the rule of inference selected for each of the formulae in the trunk of the tree is the provability oracle at the level of the propositional calculus; the wisdom of selecting the oracle will be explained soon.

So far our truth-tree has no branches. Branches emerge as soon as we decompose the formulae in our trunk into the sub-formulae that ensure the truth of

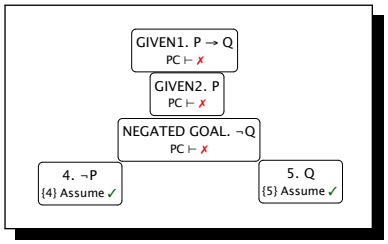
[3] Of course, as some sharp readers will have already seen, there certainly will be cases in the running of **TABLE** when observant humans will be able to stop before constructing a full table. This would be the case if such a human found FALSE for some row early in the process of testing whether some $\phi$ is TRUE in every row. In asserting that $2^n$ is the function that characterizes **TABLE**, we are following standard practice by taking the *worst case* to be definitive. In addition, we aren't considering modifications to the algorithm designed to produce a gain in efficiency. The fact is, none of these gains will produce a new algorithm that is better than the truth-tree one we are about to present.

Figure 2.7: Trunk of the Truth-Tree for Verifying *Modus Ponens*




formulae in that trunk. We do this for each formula in the trunk, starting from the top. In the case of the top-most formula in the trunk of the truth-tree depicted in Figure 2.7, such decomposition results in two possibilties: namely, that ¬P holds, or that Q holds; so we have the tree shown in Figure 2.8.

Figure 2.8: Two Branches Emerge in the Truth-Tree for Verifying *Modus Ponens*




But look what happens now if you start from the bottom of each branch and work up. Take the left (from your perspective) branch first. Notice that you run into an explicit contradiction: ¬P, and then in your second step moving up you hit P — contradiction indeed! Now let's start at the bottom of the right branch. Here we first observe Q, and then we start working upwards … and we reach ¬Q immediately, which gives us another contradiction. Since these are the only branches in the tree, and everything in the trunk has been decomposed, our verdict is that $\{P \rightarrow Q, P\} \vdash Q$ is true. Of course, we already knew that this equation was true; we used it only to get our truth-tree feet wet. Let's try a somewhat harder pair of challenges. But before we do so, let's have our full list of decomposition rules before us. The first set of such rules are **branching rules**, one of which (the one for conditionals) we just used in building our first truth tree. The complete set of such rules is shown in Figure 2.9. There are five branching rules, one for each formula type.

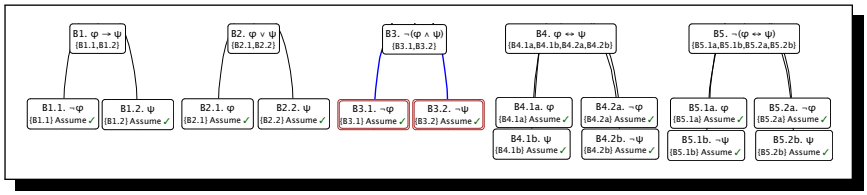
Consider first the first three types: conditionals ($\phi \rightarrow \psi$), disjunctions ($\phi \vee \psi$), and negated conjunctions ($\neg(\phi \wedge \psi)$. In each of these three cases, make sure you understand that each member of each pair of formulae is sufficient to guarantee the truth of the compound formula immediately above. For instance, consider the negated conjunction. Here, if $\neg\phi$ (the left branch) holds, then $\neg(\phi \wedge \psi$ must

hold. (This has been verified in Slate, by linking upwards from $\neg\phi$ to $\neg(\phi \wedge \psi)$ and consulting the PC oracle. We have hidden the green-check certification provided by Slate, as well as the node showing that PC $\vdash$ is the rule that has been invoked.) In the somewhat more complicated cases of the biconditional, and the negated biconditional, you will see from Figure 2.9 that the two branches that are created are "two rows long," unlike the shorter branches in the other three cases. Do you see why that is? The answer is that in the caase of these two formula types, there are two ways to guarantee the truth of that formaula. For example, in the case of a negated biconditional,

$$\neg(\phi \longleftrightarrow \psi),$$

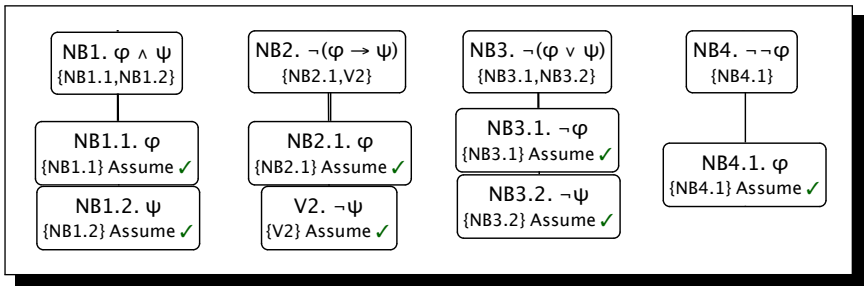
the case of both $\phi$ and $\neg\psi$ (= the left branch) holding secures the truth of the negated biconditional, *and* the case in which both $\neg\phi$ and $\psi$ holds (= the right branch) likewise secures the truth of that same negated biconditional.

Figure 2.9: Branching Rules for Constructing Truth-Trees



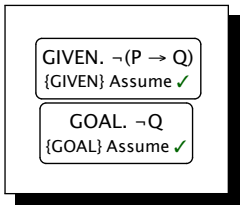

The branching rules given in Figure 2.9 don't constitute a complete set of rules for building truth trees: we're missing the *non*-branching decomposition rules. These additional four rules are shown in Figure 2.10. When applying any rule from this quartet, one simply builds out the branch on which the formula to be decomposed resides.

Figure 2.10: Non-Branching Rules for Constructing Truth-Trees



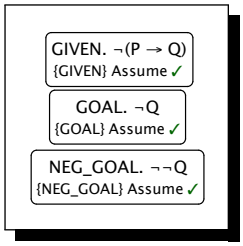

Let's now proceed start to finish on the construction of a truth-tree that shows $\neg(P \rightarrow Q) \vdash \neg Q$ to hold. Our first step is to establish our trunk by simply stacking our given content, given first, followed immediately thereafter by goal that supposedly follows from that content. See Figure 2.11.

Figure 2.11: Step 1 in Construction of Truth-Tree for $\neg(P \rightarrow Q) \vdash \neg Q$




Next, we negate our goal and place it, which extends the trunk down; see Figure 2.12.

Figure 2.12: Step 2 in Construction of Truth-Tree for $\neg(P \rightarrow Q) \vdash \neg Q$




In our next move, we decompose the topmost formula in our trunk (the given) by rule NB2, and expand the truth of our tree down. This yields what's now shown in Figure 2.13. At this point we finished with the construction side of things, and can simply examine our tree, to see if, when working bottom to top along each branch back to and through our trunk we find an explicit contradiction, that is a pair of formulae having the form $\phi$ as the first member of the pair, and $\neg\phi$ as the other. If our examination locates such a clash for every branch, we declare entailment to hold between givens and the goal. In the case before us, specifically in Figure 2.13, we find the contradictory pair: $\neg Q$ and $\neg\neg Q$. Hence our verdict is "Valid argument!" (or "Entailment holds!"). We have suitably annotated the truth-tree showin in Figure 2.13, and have in addition used the PC oracle in Slate to confirm that the branch indeed contains an aburdity, by showing that this oracle can prove that $0 = 1$ from the branch. It's by the way crucial to note that in the case before us, *every* branch yields an absurdity. Any completed truth-tree with a branch that is consistent is confirmation that the argument (individual formula) under review fails (isn't a validity).

### 2.6.1 Exercises

1. Construct a truth-tree that shows $[\neg(P \rightarrow Q) \wedge \neg(Q \rightarrow R)] \rightarrow S$ to be a validity.

Figure 2.13: Step 3 in Construction of Truth-Tree for $\neg(P \rightarrow Q) \vdash \neg Q$

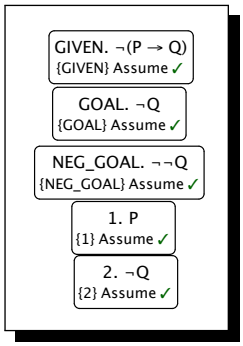
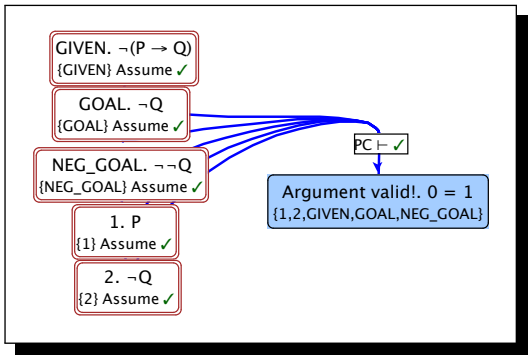



Figure 2.14: Annotated Truth-Tree for $\neg(P \rightarrow Q) \vdash \neg Q$




## 2.7 Other Proof Systems

### 2.7.1 Propositional Calculus in *Principia Mathematica*

The field of artificial intelligence (AI), at least in its modern form, was born in 1956 at Dartmouth College, when a group of scholars assembled to consider how human-level intelligence might be given to a computing machine. Among these scholars was future-nobelist[4] Herbert Simon, and he undeniably stole the show. He did so by revealing his stunning computer program LOGIC THEORIST,[5] which was able to discover and verify a shocking number of theorems from Whitehead and Russell's

[4] Simon won the nobel prize in economics in 1978. His nobel-prize lecture, which we heartily recommend to you, is available at

http://www.nobelprize.org/nobel_prizes/economic-sciences/laureates/1978/simon-lecture.pdf

[5] This program is discussed in more detail at

http://en.widipedia.org/wiki/Logic_Theorist#Citations

and in Russell & Norvig (2009) in connection with the original AI conference to which we've alluded.

*Principia Mathematica* (= *PM*), a landmark three-volume work of 1927[6] intended to specify firm foundations for mathematics, in the wake of a valiant but failed attempt by Gottlob Frege to reach the same goal. (Later on we shall have occasion to see why and how Frege's program failed (see §3.8.2).) Such theorems had hitherto been the province only of human beings, and Simon's work marked the dawn not only of AI, but more narrowly of the field of **automated theorem proving**, which, as you know, is at the heart of our modern approach to logic, makes Slate possible, and is still steadily progressing today. We now briefly discuss the axiom system for the propositional calculus presented in *PM*.

The axiom system **PM** is composed of the following elements. First, there are four axiom schemas:[7]

A1 $(\phi \vee \phi) \rightarrow \phi$

A2 $\phi \rightarrow (\phi \vee \psi)$

A3 $(\phi \vee \psi) \rightarrow (\psi \vee \phi)$

A4 $(\psi \rightarrow \chi) \rightarrow ((\phi \vee \psi) \rightarrow (\phi \vee \chi))$

Notice that we say these are axiom *schema*. This is so because each of the four is allowed to be instantiated on the basis of any consistent assignments to $\phi$, $\psi$, and $\chi$. For example, we can infer from A1 (by the rule we shall call 'instantiate') that, where M represents the statement that the Moon is made of green cheese:

A1* $(\mathsf{M} \vee \mathsf{M}) \rightarrow \mathsf{M}$

The only other rule of inference in **PM** is the familiar *modus ponens*, which as you know corresponds directly to conditional elimination in Slate. What about formulae that contain truth-functional connectives other than the ones appearing in A1–A4? After all, as you will no doubt have noticed, this quartet makes no reference to negation, conjunction, or the biconditional. The answer is that **PM** includes three definitions:

DefCond $\phi \rightarrow \psi =_{def} \neg\phi \vee \psi$

DefConj $\phi \wedge \psi =_{def} \neg(\neg\phi \vee \neg\psi)$

DefBiCond $\phi \leftrightarrow \psi =_{def} (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$

So that we can employ these definitions in Slate, we make explicit the fact that each '$=_{def}$' indicates a biconditional by replacing each with a biconditional, and in addition, we call out both the left and right directions that are implicit in these three definitions. This yields:

DefCondL2R $(\phi \rightarrow \psi) \rightarrow (\neg\phi \vee \psi)$

[6] This is the year of the first "combined" release. Volume I was originally published in 1910; II in 1912, and III in 1913. The 1927 "combo" version included modifications and additions to the original, separately released three volumes.

[7] Actually, Whitehead and Russell provided five, but it was later discovered that one of the axioms was redundant (it could be proved from the quartet we list here, using the deductive machinery W&R availed themselves of, which we present immediately below). The redundant fifth axiom schema:

A5 $(\phi \vee (\psi \vee \chi)) \rightarrow (\psi \vee (\phi \vee \chi))$

| | |
|---|---|
| DefCondR2L | $(\neg\phi \vee \psi) \rightarrow (\phi \rightarrow \psi)$ |
| DefConjL2R | $(\phi \wedge \psi) \rightarrow \neg(\neg\phi \vee \neg\psi)$ |
| DefConjR2L | $\neg(\neg\phi \vee \neg\psi) \rightarrow (\phi \wedge \psi)$ |
| DefBiCondL2R | $(\phi \leftrightarrow \psi) \rightarrow (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$ |
| DefBiCondR2L | $(\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi) \rightarrow (\phi \leftrightarrow \psi)$ |

Every propositional-calculus theorem provable in Slate is provable in **PM**, and vice versa. We could prove this meta-theorem (and perhaps you could as well), but we won't here. Instead, we rest content with showing that a particular theorem easily established in Slate can be proved in **PM** as well; the theorem is:

$$\mathsf{P} \rightarrow \mathsf{P}$$

We can show this by finding a Slate proof of $\mathsf{P} \rightarrow \mathsf{P}$ in which the only rules used are `instantiate` and *modus ponens* (= conditional elimination).[8] We begin by using instantiate on A4 to obtain:

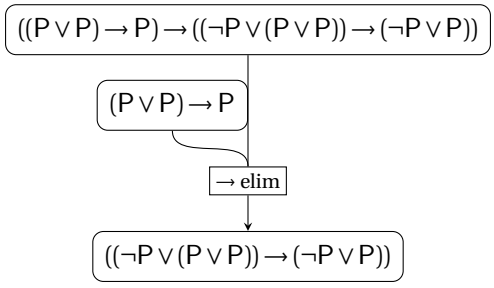
Instantiate:A4 $((\mathsf{P} \vee \mathsf{P}) \rightarrow \mathsf{P}) \rightarrow ((\neg\mathsf{P} \vee (\mathsf{P} \vee \mathsf{P})) \rightarrow (\neg\mathsf{P} \vee \mathsf{P}))$
$\{\}$ PC $\vdash$ ✓

Notice that here the official rule of inference used is the oracle for entailment, but this is only because Slate doesn't have the rule `instantiate` built-in. Technically speaking, the node here is not justified by appeal to one of the oracles in Slate, but rather by the fact that, as can be verified by inspection, `Instantiate:A4` is indeed a deduction from A4. You should be sure that you see what is assigned to the meta-variables in A4 in order to obtain `Instantiate:A4`.

The next step is to instantiate A1 to obtain this node in Slate:

Instantiate:A1 $(\mathsf{P} \vee \mathsf{P}) \rightarrow \mathsf{P}$
$\{\}$ PC $\vdash$ ✓

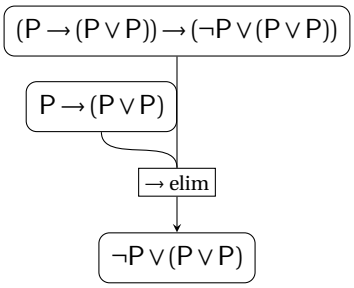
Our next step is to infer from this node and the one preceding it that we have the consequent of `Instantiate:A4`; i.e., leaving aside labeling information, we have the following proof, which will be one of our key building-block sub-proofs:




(2.36)

Our next building-block proof is produced by suitably instantiating DefCondL2R, and suitably instantiating axiom A2, in order to deduce $\neg\mathsf{P} \vee (\mathsf{P} \vee \mathsf{P})$. Here's this proof:

[8] This proof is courtesy of Naveen Sundar Govindarajulu.




(2.37)

As you have perhaps already observed, the immediate consequence of the building-block proofs we have constructed is

$$\neg P \vee P$$

So, if we had on hand

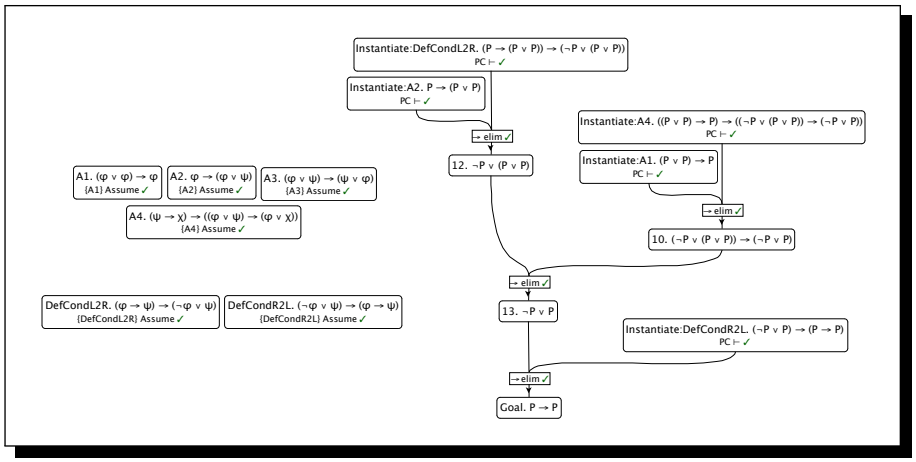
$$(\neg P \vee P) \rightarrow (P \rightarrow P)$$

we would be able to instantly deduce our desired goal; viz., $P \rightarrow P$ by *modus ponens*. But alert readers will have already observed that we can obtain

$$(\neg P \vee P) \rightarrow (P \rightarrow P)$$

from Instantiate:DefCondR2L; hence we are done! To sum things up and present the composite proof, we encourage you to study Figure 2.15.

Figure 2.15: Full Slate Proof of *Principia*'s Derviation of $P \rightarrow P$




## 2.7.2 Resolution

Another kind of deduction that students (esp. those interested in AI, computer programming, computer science, etc.) should be familiar with is **resolution**. For the

moment, we can think of resolution as consisting simply of a form of "cancellation." The idea here is that whenever two formulas are contradictories, that is we have a pair $\phi$ and $\neg\phi$, these cancel out and disappear, leaving other formulas that are in play.

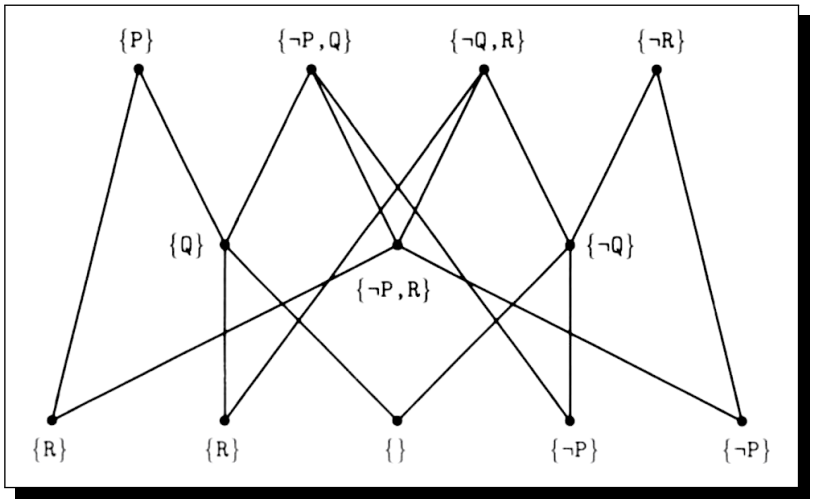
#### 2.7.2.1 Resolution Graphs Captured by Slate

Deduction by resolution is sometimes presented in the form of graphs. For example, Genesereth & Nilsson (1987) present a **resolution graph** corresponding to this entailment:

$$\Delta \vdash \bot,$$

where $\Delta := \{\mathsf{P}, \neg\mathsf{P} \vee \mathsf{Q}, \neg\mathsf{Q} \vee \mathsf{R}, \neg\mathsf{R}\}$. In resolution, as we've noted, a contradition is denoted by the empty clause, that is the empty set; and the elements of non-empty sets are disjuncts. Hence, in the resolution framework, $\{\neg\mathsf{P}, \mathsf{Q}\}$ is equivalent to $\neg\mathsf{P} \vee \mathsf{Q}$, and $\{\}$ is a contradiction. Now please see the resolution graph shown in Figure 2.16.

Figure 2.16: A Resolution Graph Given by (Genesereth & Nilsson 1987)

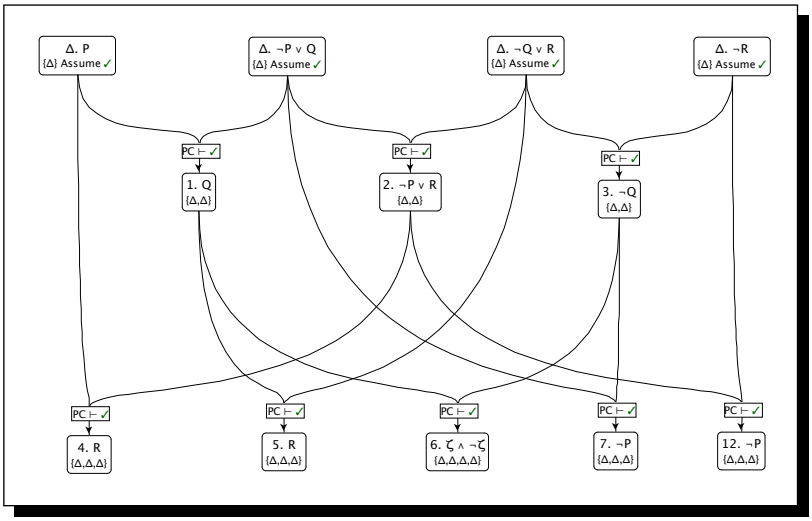



Here's an encouraging theorem regarding the relationship between resolution graphs and Slate:

> **Theorem**: Every resolution graph $\mathscr{G}_r$ can be expressed as a Slate hypergraph $\mathscr{G}_s$; and all that is provable in extensions of $\mathscr{G}_r$ is provable in an extension of $\mathscr{G}_s$.

We find this encouraging because it means that the power of Slate includes what can be done via resolution. We don't prove the theorem, but we do provide a Slate hypergraph that expresses the resolution graph shown in Figure 2.16: see Figure 2.17. In this figure, the formula $\zeta \wedge \neg\zeta$ is used as a stand-in for $\bot$.

Figure 2.17: A Slate Hypergraph the Captures the Resolution Graph in Figure 2.16




## 2.8 Programming via Slate

We promised in the Preface that our modern approach to logic forges an intimate connection to the particular sciences; and we in particular promised to make a connection to computer science and (to a lesser degree) physics. Toward making good on this promise, let's now establish part of the link to computer science, by explaining how to program a computer by using Slate in a straightforward manner. Be forewarned: We won't formulate in this section any robust computer program, but rather some confessedly simple ones. But it's a start! And though the programs we begin with are simple, they are instructive, in no small part because their limitations reveals the kind of processing we surely want, at least eventually, from any ultimately acceptable programming language.

As you now know, one of the basic operations that you must be proficient at in order to productively use Slate is the keying in of formulas to nodes. And as you also know, this keying in happens when when you type formulas into Slate in infix format, without the special symbols that denote Boolean connectives. This format, in the lingo of computer programming, is that of an **S-expression**, which is the fundamental building block of the oldest programming language in existence: Lisp.[9] For example, if you want to create a node in Slate that contains the formula $\mathsf{P} \wedge \mathsf{Q}$, you type `(and P Q)` and then click on OK. And if you wanted to create $n$ nodes, each with its own formula, you would type the input for each and every one of these nodes. For instance, it you wanted to create a node for each of

[9] Well, some say that Fortran is a tad bit older, but we disagree — and at any rate, we do bring to your attention the rather interesting fact that Lisp and its variants are still in vibrant (but not, we confess, widespread) use in our new millennium, and the same, believe it or not, goes for Fortan!

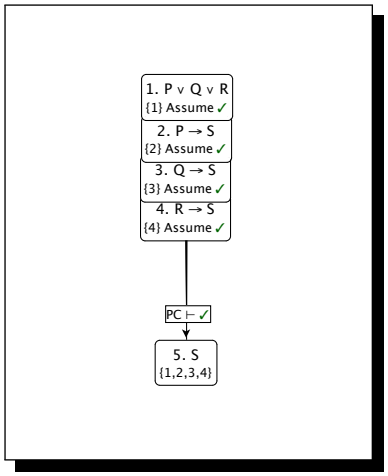
- $P \vee Q \vee R$
- $P \rightarrow S$
- $Q \rightarrow S$
- $R \rightarrow S$

you would type (and enter), respectively,

- `(or P Q R)`
- `(if P S)`
- `(if Q S)`
- `(if R S)`

We say in this case that you have **asserted** this quartet of formulae, and say that for purposes of writing a program, the union of what you assert composes the **assumption base**. Note that the assumption base is simply a list of formulas. In the example just given, we denote the assumption base by 'Γ'. If you have followed along to this point in the present section with Slate at your side, you are facing a workspace that includes Γ, you therefore you have already written a computer program. But what can you *do* with your program? Well, for starters, you can get answers to queries from it. For instance, you could query against Γ to see if Q holds given this assumption base. You can do that by using the oracle for the propositional calculus (with which, by now, you are quite at home), and asking specifically whether given Γ, it follows that Q does in fact hold. Figure 2.18 shows the situation, with the answer of YES returned (where ✓is interpreted as YES).

Figure 2.18: Progam1 Returns a YES to a Query Against $\Gamma_1$



You might regard this to be some pretty inconsequential programming. Let's see if we can do something a bit more interesting, by applying the same approach not to the propositional calculus, but to the pure predicate calculus, and by creating an assumption base ($\Lambda$) about the llamas Larry and Lucy in Llama Land.[10]

Here are some things we can tell you about Larry and Lucy, so that $\Lambda$ can be populated. To begin, both Larry and Lucy are llamas, and Larry is male while Lucy is female. We also know that Larry has three brothers: Lance, Lester, and Luke. In addition, Lucy is motherless, but her father is Hank. As a general principle of Llama Land, let's assume that anyone with three brothers likes the father of any motherless llama. One seemingly sensible way to assert this information into Slate (save for the general principle just mentioned; we'll get to it a bit later) is to assert the following nine formulas, and we assume that you will now do exactly that in a predicate-calculus workspace.

```
(Llama larry)
(Llama lucy)
(Male larry)
(Female lucy)
(BrotherOf lance larry)
(BrotherOf lester larry)
(BrotherOf luke larry)
(Motherless lucy)
(FatherOf lucy hank)
```

How good is the program you have now written? Let's find out, by issuing four simple queries against it. You should attempt each one. (In order to do so, of course, you will need to ask the oracle if the corresponding formula, which you will have to invent, can be proved from $\Lambda$.)

Q1 Is it true that both Larry and Lucy are llamas?

Q2 Is it true that if Lucy is motherless, then she's a female?

Q3 Is Lucy male?

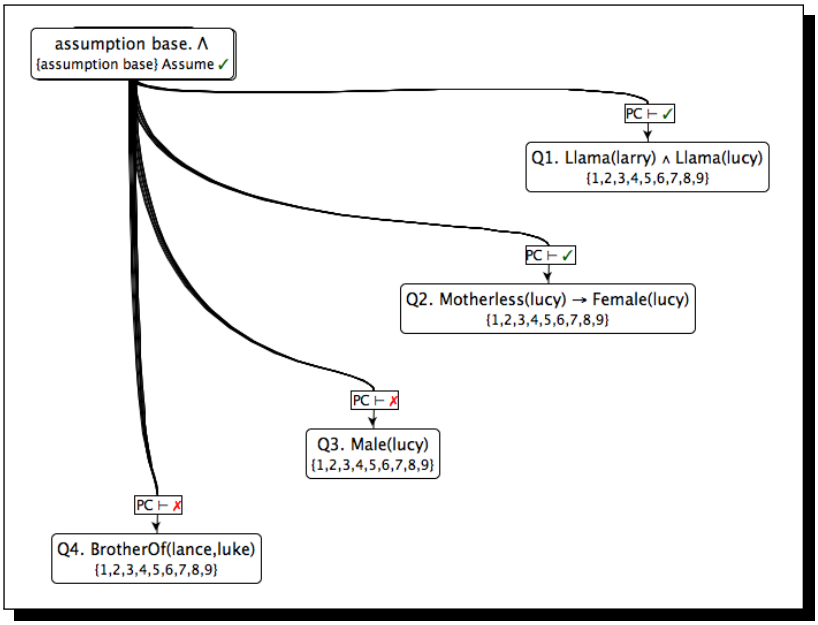
Q4 Is Lance a brother of Luke's?

What do the results look like? If you haven't made a mistake, you found that the first two queries are answered in the affirmative, as desired; but you also received what would appear to be some not-so-good news: the third and fourth queries return the dreaded ✗, or NO. In short, your workspace should look like the one shown in Figure 2.19.

But how bad is the news, really? Let's take some time to think about it; and let's specifically think about each of the two negative responses separately. Consider first the response to the query about Lance and Luke. Are we sure that the NO response is

[10] In the catchy song "Come Fly With Me," in his album *Come Fly With Me* of 1958, Sinatra melodiously beckons his listeners to travel with him to "Llama Land." The song was written and arranged for "The Voice" by Sammy Cahn and Jimmy Van Heusen.

Figure 2.19: Program2 Responds to Four Queries Against Λ




erroneous? Yes. What's the problem? The problem is that Slate doesn't understand that if $x$ is the brother of $y$, and $z$ is the brother of $y$, then $x$ and $z$ are brothers. And there's a double-whammy here: In the pure predicate calculus we can't express general principles that make use of variables. This is rather inconvenient, since, as we remember from even elementary calculus, such equations as '$3x = 12$' are useful and perfectly understandable; and in them, $x$ is a variable (which here can be instantiated by '4' in order to confirm the identity). The fact is, while the pure predicate calculus is on the one hand impressively clear, on the other hand, you can't say a lot in it. Part of the seminal genius of Edgar Allen Poe's poem "The Raven" is that the raven featured in it does manage to impressively say an awful lot with but the one deep and dreary word (the ingeniously ambiguous 'Nevermore'), but at the end of the day one word is one word, and a vocabulary limited to it is very limited. Likewise, the pure predicate calculus, lacking variables (and therefore lacking use of them), is severely limited. This limitation will be surmounted when we move to the next chapter, and first-order logic.

Yet we can patch the situation for now by simply augmenting Λ with the specific conditional[11] that's needed; viz.,

[11] Alert readers will see that our program gives Slate no reason to believe another principle of brotherhood that we know to be obvious: If $x$ is a brother of $y$, then $y$ is a brother of $x$. Since, again, variables are verboten in the pure predicate calculus, we can't add this principle to our program directly. Instead, we would need to add the relevant concrete formula for every relevant individual. Here's the formula for Lance and Larry:

```
(if (BrotherOf lancy larry) (BrotherOf larry lance))
```

```
(if (and (BrotherOf lance larry) (BrotherOf luke larry)) (BrotherOf lance luke))
```

With this formula added to Λ, you should see that instantly Q4 is now answered in the affirmative. But now what about query Q3? Is vLucy male? Here, as we have noted, the response was also the red cross. What this specifically means is that it can't be proved that Lucy is male. And, when you think about it, this is indeed exactly what we would desire as output. But what happens now if we change our query to this one:

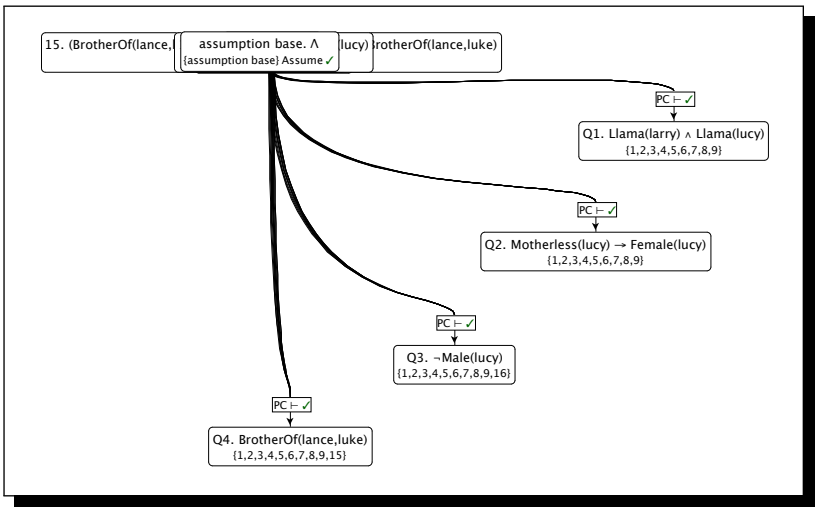
Q5 Is it the case that Lucy is not male?

As you will see, if you issue this query against Λ, the response will be the ✗, but this time this response is surely unwanted, and reveals a bug in our program. After all, since Lucy is a female, she's not a male. The solution is to add a second new concrete formula to Λ, to wit:

```
(if (Female lucy) (not (Male lucy)))
```

At this point, with these additional formulas asserted, we obtain our desired responses: see Figure 2.20. Yet as "green-checked" as this result is, there are two loose ends that need to be resolved, namely:

Figure 2.20: Program3 Correctly Responds to Four Queries Against Λ




1. The strategy of asserting concrete conditional facts as a substitute for variable-based principles has only been partially implemented. This is easy to see, if we consider a query that's a variant of Q4:

   Q4′ Is Lester a brother of Luke's?

As things stand at the moment, $\Lambda$ doesn't contain the correlate of the conditional that would be needed to secure a `yes` in response to this query from Slate. But we assume it would be easy enough for you to assert now the additional facts needed for this, and for other straightforward queries that will occur to you.

2. Finally, we note that we are going to have to return later to the problem of how to express general principles with variables in computer programs written in Slate. Recall that we stated at the beginning of the present section that Llama Land is governed by this general principle:

   3B Anyone with three brothers likes the father of any motherless llama.

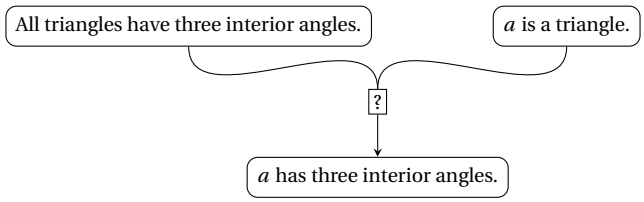
   There is simply no way to express this principle in the pure predicate calculus. The best we can do is to start listing and asserting each and every specific, variable-free fact corresponding to the principle, but that would be most distasteful, and horribly inefficient.[12] If Llama Land contains, say, a million families in which there are three or more brothers, and, say, a million motherless llamas who have fathers, the programmer restricted to the pure predicate calculus would likely go insane. (How many formulas would be needed to enforce the general principle in the case of this many agents?) So the best strategy is to wait until we can increase the expressive power of Slate programs so as to be able to express 3B with just one relatively simple formula. There won't be much waiting, because the next chapter, to which we now turn, provides the needed expressive power, in the form of **first-order logic**.

[12] So that this point is vivid, we encourage the you to devise and assert all the variable-free formulas that are needed. To test your work, you can then ask, for each of the three brothers, whether they like Hank.

# Chapter 3

# First-Order Logic

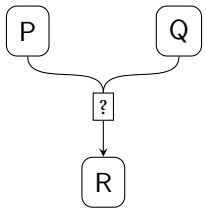
The propositional calculus is a powerful tool that has allowed us to formalize and evaluate many arguments, and to derive many useful results. Unfortunately, there are many kinds of "real-life" sentences, including many in mathematics and other disciplines, whose meaning cannot be captured by the propositional calculus. For instance, consider the obviously valid argument in (3.1).[1]




(3.1)

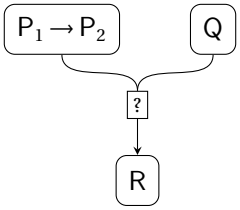
How might we go about formalizing this using the propositional calculus? Earlier, we decomposed natural-language sentences according to truth-functional connectives, and replaced the non-decomposable propositions with atomic formulas. If we apply this approach here, what happens? "All triangles have three interior angles." cannot be decomposed, and so we assign it an atomic formula; say, P. Similarly, we assign "$a$ is a triangle" to Q. Finally, "$a$ has three interior angles" is assigned R. Unfortunately, the resulting argument structure, shown in (3.2), is not a valid application of any of our inference rules. Indeed, this is rather an understatement: from the standpoint of valid deduction, which as we have noted is distinguished by the fact that if the premises are true the deduced conclusion *absolutely must* be true, (3.2) is almost pitiful.

[1] We here assume that the context is standard Cartesian geometry.



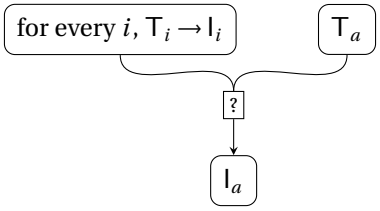

(3.2)

Reflecting upon the meaning of "All triangles have three interior angles," we recognize that an equivalent phrasing is: "For every thing, if the thing is a triangle, then it has three interior angles." This is promising, because we now observe an "if ... then ..." structure, and this is familiar; we know how to formalize if-then statements using the conditional. So let's try to formalize as follows. To "the thing is a triangle" we assign the atomic formula $\mathsf{P}_1$, and to "the thing has three interior angles" the atomic formula $\mathsf{P}_2$. For "$a$ is a triangle" and "$a$ has three interior angles" we will continue to use $\mathsf{Q}$ and $\mathsf{R}$. This refined argument structure is shown in (3.3).




(3.3)

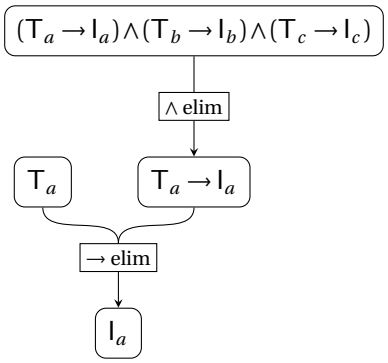
This structure is a bit more promising, for if $\mathsf{P}_1$ were $\mathsf{Q}$ and $\mathsf{P}_2$ were $\mathsf{R}$, we could apply conditional elimination. Let's keep this hope in mind: the move here isn't applicable yet, but perhaps we can use it later. We lost something, note, in turning "for everything, if the thing is a triangle, then the thing has three interior angles" into $\mathsf{P}_1 \Rightarrow \mathsf{P}_2$, because in the English there is a connection between "the thing is a triangle" and "the thing has three interior angles," namely "the thing," but there is no corresponding connection between $\mathsf{P}_1$ and $\mathsf{P}_2$; they are two distinct atomic formulas. A better representation would preserve some connection between these things. As a slightly different approach, let's try representing any phrase of the form "$x$ is a triangle" by $\mathsf{T}_x$, and "$x$ has three interior angles" by $\mathsf{I}_x$. With this representation, our first premise is really more of a schema; it is an assertion of $\mathsf{T}_i \Rightarrow \mathsf{I}_i$ for any $i$ that we consider. Then our argument would look something like what is shown in (3.4).




(3.4)

Now, if when we wrote "for every $i$" when we had some finite set of possibilities, say, $\{a, b, c\}$, then "for every $i$ $\mathsf{T}_i \Rightarrow \mathsf{I}_i$" is really just shorthand for the conjunction

$(\mathsf{T}_a \Rightarrow \mathsf{I}_a) \wedge (\mathsf{T}_b \Rightarrow \mathsf{I}_b) \wedge (\mathsf{T}_c \Rightarrow \mathsf{I}_c)$. In such a case, we can construct the appropriate proof in the propositional calculus; for instance, the proof in (3.5) is a suitable proof.




(3.5)

In fact, if we can guarantee that the number of things about which we ever wish to refer is finite, we can always use this approach, though it will quickly become too cumbersome to be genuinely viable. (Consider the assertion that every resident of New York City lives south of Montreal, expressed with help from a list of the names of such residents. Following the propositional-calculus approach to build a vast conjunction would require rather a lot of time!) On the other hand, since formulas in the propositional calculus cannot be infinitely long, this representation fails if we care to speak of an infinite number of things (for instance, the integers). We see, then, that the expressive power of the propositional calculus is fundamentally limited, and this fact motivates the development of a new language, **first-order logic** (or just **FOL**), which has greater expressive power. (Alert readers will remember that we discussed FOL back in the Preface.)

## 3.1 What's New?

Of course, we do not want to discard all the groundwork that is already laid courtesy of the propositional calculus and the previous chapter; hence we will extend the syntax of the language for this calculus; and later we will discuss a new semantics that builds atop that used for the the propositional calculus (truth tables). One source of the limited expressivity of the propositional calculus is an inability to refer to specific entities. We might let an atomic formula $P$ stand for "John admires Mary" and $Q$ for "Mary is older than John," but inspection of $P$ and $Q$ gives no information about the entities involved; that is, no information about John and Mary, nor about the relations among involved entities, that is, 'admiration' and 'seniority.' The extensions by which we move to first-order logic address these needs, by letting us describe particular individuals and relationships that may hold among them.

### 3.1.1 Describing Individuals

When we speak of individuals, we commonly refer to them by names. In the examples given earlier, we spoke of 'John' and 'Mary.' These are names that denote specific individuals, and we shall call such names **constants**. To formally represent the individuals denoted by 'John' and 'Mary' we use constant symbols, such as john and mary. (We have introduced some terminology, "constant symbol," without formal definitions, but soon give a formal account.)

We refer to some individuals, however, not by a direct name, but in terms of other individuals. For instance, 'Antonia' and 'the mother of Claudius' refer to the same individual, although the only names used by the latter phrase are those of other individuals. 'Antonia' can be expressed by antonia, and 'Claudius' by claudius, but we introduce a function symbol, mother−of to represent 'the mother of Claudius' by mother−of(claudius).

We can also refer to individuals by way of **variables**. We use this technique when the individuals in question fall within some range of possibilities, but are not fully specified. You first saw this technique in action when you were presented with equations in elementary algebra; for example equations like

$$x + y = 12.$$

Here $x$ and $y$ are of course the variables, and the idea is that they range over a host of possible pairs of numbers that would allow the equality to hold. One possibility is that $x$ is 6 and so is $y$; another is that $x$ is 1 and $y$ is 11; and so on. You can see that the variables do indeed refer to individuals, but not to *particular* individuals.

Let's consider another pair of examples from arithmetic, to see the simple use of variables in action. We saw above that we may use a function symbol to represent 'the mother of' some individual, provided that we know how to speak of the individual. But consider the sentences 'Every number is less than its successor.' and 'There is a number such that 3 is its successor.' We haven't yet discussed how we would represent such sentences, but this much should be immediately clear: within these sentences, we speak of 'it'—meaning, in the first case, any number we care to think of, and, in the second case, some particular number whose identity we do not entirely specify (but whose existence we affirm). Using a function symbol for successor, say, succ, and a constant symbol for a particular number, say two for the number 2, we could represent the successor of 2 by succ(two). To represent the unspecified entities, we can turn again to variables, for example to $x$. (We haven't yet addressed how to ensure that use of a variable captures the intended meaning, e.g., that $x$ in succ($x$) can range over 'every number.' Don't worry; we'll get to this soon.)

### 3.1.2 Describing Relations

We now have the ability to refer to individuals, but referencing individuals is not of much use if we are unable to say anything about them. We considered earlier the sentence 'John admires Mary.' and can now refer to John and Mary using the constant symbols john and mary. To express the relationship between them, we introduce a

relation symbol, Admires, and in notation like that for function symbols, we write Admires(John, Mary). For some relationships, we may use a more traditional infix notation. For instance, for equality, we could write antonia = mother−of(claudius) and two ≤ three.

### 3.1.3 Describing Quantification

We described the use of variables to stand for objects for which we do not have a particular name, but left open the question of how to express sentences that use these kind of references. In first-order logic, we use two quantifiers, the universal quantifier, ∀, and the existential quantifier, ∃. To express, for instance, the sentence, "Every number is less than its successor," we first cast it more formally (making explicit an implicit conditional) as, "for every $x$, if $x$ is a number, then $x$ is less than $x$'s successor." This latter form is represented as $\forall x\ [\mathsf{Number}(x) \Rightarrow (x \leq \mathsf{succ}(x))]$. As another example, 'a son of Antonia was emperor' can be expressed by

$$\exists y \left[\mathsf{SonOf}(\mathsf{antonia}, y) \wedge \mathsf{Emperor}(y)\right].$$

### 3.1.4 Formal Grammar for First-Order Formulae

We have now described all the syntactic components of first-order logic, and will summarize by presenting a formal grammar for first-order expressions. We find it useful to break down expressions into two types, **terms** and **formulae**.

A term is either a variable, $v_i$; or a function symbol, $f^i_j$, followed by a left parenthesis, $i$ terms separated by commas, and finally a right parenthesis. In the case that $i$ is 0, we may omit the parentheses.

Formulae are either **atomic**, **compound**, or **quantified**:

- An **atomic formula** is a relation symbol $R^i_j$ followed by a left parenthesis, $i$ terms separated by commas, and finally a right parenthesis. Again, in the case that $i$ is 0, we may omit the parentheses.
- A **compound formula** is a formula constructed from truth-functional connectives and other formulae. The construction works as follows. If $\phi_1, \ldots, \phi_n$ are formulae, so are negations $\neg\phi_i$, conjunctions $\phi_1 \wedge \ldots \wedge \phi_n$, disjunctions $\phi_1 \vee \ldots \vee \phi_n$, conditionals (or implications) $\phi_i \Rightarrow \phi_j$, and biconditionals (or biïmplications) $\phi_i \Leftrightarrow \phi_j$.
- A **quantified formula** is a quantifier (either the universal quantifier, ∀, or the existential quantifier, ∃) followed by a variable and then by a formula, e.g., for any variable $v$ and formula $\phi$, $\forall v\ \phi$ and $\exists v\ \phi$ are quantified formulae.

This grammar is summarized in Table 3.1.

| | | |
|---|---|---|
| **Terms** | | |
| Variables | $v_0, v_1, \ldots$ | |
| Function Applications | $f_j^0$ | Constants |
| | $f_j^i(\tau_1, \ldots, \tau_i)$ | Where $\tau_1, \ldots, \tau_i$ are terms |
| **Formulae** | | |
| Atomic Formulae | $R_j^0$ | Propositions |
| | $R_j^i(\tau_1, \ldots, \tau_i)$ | Where $\tau_1, \ldots, \tau_i$ are terms |
| Compound Formulae | $\neg\phi_i$ | |
| | $\phi_1 \wedge \ldots \wedge \phi_n$ | |
| | $\phi_1 \vee \ldots \vee \phi_n$ | where $\phi_1, \ldots, \phi_n$ are formulae |
| | $\phi_i \Rightarrow \phi_j$ | |
| | $\phi_i \Leftrightarrow \phi_j$ | |
| Quantifications | $\forall v\ \phi$ | where $v$ is a variable and $\phi$ is a |
| | $\exists v\ \phi$ | formula |

Table 3.1: The syntax of first-order expressions.

## 3.2 Free and Bound Variables

The grammar given in Table 3.1 allows us to construct a number of kinds of formulae. All expressions that are formed according to this grammar for formulae are called **well-formed formulae**, or just **wffs**. We notice an interesting property of well-formed formulae: they may contain variables that are not within the scope of any quantifier quantifying over them. For instance, the first occurrence of $x$ in $Q(x) \wedge \forall x\ P(x)$ is not within the scope of a $\forall$ or $\exists$ associated with $x$.

This observation is important enough for us to introduce terminology to describe it: We say that an occurrence of a variable is **bound** if it appears within the scope of a quantifier that quantifies over that variable, and **free** otherwise. Note that what we say are free and bound are particular *occurrences* of variables. For instance, the first occurrence of $x$ in $Q(x) \wedge \forall x\ P(x)$ is free, while the second and third occurrences are bound. We denote by $\mathbf{fv}\phi$ and $\mathbf{bv}\phi$ the sets of variables of which there are free and bound occurrences, respectively, in $\phi$. $\mathbf{fv}(\phi)$ and $\mathbf{bv}(\phi)$ are not necessarily disjoint, for, as we have just seen, a formula may contain free and bound occurrences of the same variable.

A well-formed formula $\phi$ containing no free variable occurrences is called a **sentence**. Symbolically, $\phi$ is a sentence provided that $\mathbf{fv}(\phi) = \emptyset$.

### 3.2.1 Substitutions

We shall often find it useful to replace free variables in expressions with terms; this is process known as **substitution**. Our goal in this process is to replace free occurrences of a variable in a formula with a term, in such a way that the meaning

of the formula changes only in that what the formula said about the variable, it now says about the term. Life in the civilized world requires us to run the process of substitution very frequently, but we usually don't think about the process at all. For instance, anyone who, having in mind the general structure of an email address at the time of the writing of this sentence, knows that that structure is one in which left of the @ symbol is a string that essentially serves as a name, and right of the @ symbol is a string that classifies the kind of group into which the agent who is named falls. In light of this, if RPI student Sylvia says to you: "My email is just my first name, and then my university's acronym followed by 'edu'." you can by the process of substitution into the structure of an email address end up with:

`sylvia@rpi.edu`.

Of course, we must be more precise about the process of substitution.

We define the process of substitution incrementally, first for terms, and then for formulae. We shall say that the process, when applied, yields a **substitution instance** from what the process is applied to. For an expression $e$, $e\{\tau/x\}$ is the substitution instance produced by running the process; that is, by substituting term $\tau$ for free occurrences of $x$. Other notations, including $e\{x \mapsto \tau\}$ are also common, as are prefix notations; for example, you will sometimes see $\sigma(e)$ where $\sigma = \{\tau/x\}$. Notice that the notation in all these cases, including our own, is intended to point toward the simple algorithm of replacing a term with a variable.

For terms, substitution instances are simple to calculate, as any variable occurrences in a term, in the context of that term, are free. For a variable $v$, $v\{\tau/x\} = v$ (where $x \neq v$), and $v\{\tau/x\} = \tau$ when $x = v$. For a compound term $f^i_j(\tau_1, \ldots, \tau_i)\{\tau/x\} = f^i_j(\tau_1\{\tau/x\}, \ldots, \tau_i\{\tau/x\})$; that is, applying a substitution to a compound term produces a compound term with the same function symbol, and whose arguments are the results of applying the substitution to the arguments of the original compound term.

For an atomic formula $R^i_j(\tau_1, \ldots, \tau_i)\{\tau/x\} = R^i_j(\tau_1\{\tau/x\}, \ldots, \tau_i\{\tau/x\})$. In parallel to what we said about compound terms, applying a substitution to an atomic formula produces an atomic formula with the same predicate symbol, but with arguments produced by applying the substitution to the atomic formula's arguments.

For compound formulae, applying a substitution yields a formula with the same boolean connective, but in which the substitution has been applied to the sub-formula of the original formula. For instance, $(\neg\phi)\{\tau/x\} = \neg(\phi\{\tau/x\})$, and $(\phi \rightarrow \psi)\{\tau/x\} = \phi\{\tau/x\} \rightarrow \psi\{\tau/x\}$.

Applying a substitution to a term, atomic formula, or compound formula is straightforward, as none of these bind their variables; variable occurrences within them are, with respect to the term or formula, free occurrences and should be replaced. We must use a bit more care in applying substitutions to quantified formulae.

Suppose we want to substitute a term $\tau$ for a variable $x$ in a quantified formula. If the quantified variable is $x$, then there can be no free occurrences of $x$ within it, and so there are no occurrences to replace. Thus, we define $(\forall x\, \phi)\{\tau/x\} = \forall x\, \phi$, and similarly for existentially quantified formulae. It seems reasonable that if the

quantified variable is anything other than $x$, we could simply apply the substitution recursively to $\phi$, that is, that $(\forall y\, \phi)\{\tau/x\} = \forall y\, (\phi\{\tau/x\})$. This will work in many situations, but there is a significant edge case: What happens if the term $t$ has an occurrence of $x$? Blindly applying the substitution recursively would result in $(\exists y\ \mathsf{Likes}(x, y))\{\mathsf{f}(y)/x\} = \exists y\ \mathsf{Likes}(\mathsf{f}(y), y)$. A variable that was free in $t$ has become bound when $t$ replaced an occurrence of $x$; this is known as **variable capture**. Thus we define $(\forall y\, \phi)\{\tau/x\}) = \forall y\, \phi\{\tau/x\}$ so long as no occurrence of $y$ in $\tau$ becomes bound in the process. This, in effect, requires that either there are no free occurrences of $x$ in $\phi$, or $y$ does not appear in $\tau$.

This definition of substitution is summarized in Table 3.2.

| **Expression** $\omega$ | **Result** $\omega\{\tau/v\}$ | | |
|---|---|---|---|
| $v$ | $\tau$ | | |
| $v'$ | $v'$ | $(v' \neq v)$ | |
| $f_j^i(\tau_1, \ldots, \tau_i)$ | $f_j^i(\tau_1\{\tau/v\}, \ldots, \tau_i\{\tau/v\})$ | | |
| $R_j^i(\tau_1, \ldots, \tau_i)$ | $R_j^i(\tau_1\{\tau/v\}, \ldots, \tau_i\{\tau/v\})$ | | |
| $\neg\phi$ | $\neg(\phi\{\tau/v\})$ | | |
| $\phi_1 \wedge \ldots \wedge \phi_n$ | $\phi_1\{\tau/v\} \wedge \ldots \wedge \phi_n\{\tau/v\}$ | | |
| $\phi_1 \vee \ldots \vee \phi_n$ | $\phi_1\{\tau/v\} \vee \ldots \vee \phi_n\{\tau/v\}$ | | |
| $\phi \rightarrow \psi$ | $\phi\{\tau/v\} \rightarrow \psi\{\tau/v\}$ | | |
| $\phi \leftrightarrow \psi$ | $\phi\{\tau/v\} \leftrightarrow \psi\{\tau/v\}$ | | |
| $\forall v\, \phi$ | $\forall v\, \phi$ | | |
| $\forall v'\, \phi$ | $\forall v'\, (\phi\{\tau/v\})$ | $(v' \neq v)$ | $\star$ |
| $\exists v\, \phi$ | $\exists v\, \phi$ | | |
| $\exists v'\, \phi$ | $\exists v'\, (\phi\{\tau/v\})$ | $(v' \neq v)\, xyz$ | $\star$ |

Table 3.2: definition of substitution in terms and formulae. In lines marked with $\star$, either $v$ must not appear free in $\phi$, or $v'$ must not appear in $\tau$.

## 3.3 Example Revisited

Now we can return to our triangle example, and handle it.

When we write "all triangles…," we are really using English shorthand for the conditional, "For every thing, if the thing is a triangle…," and this will be reflected in our formalization.

$$\begin{array}{ll} & \forall v_0\ R_0^1(v_0) \Rightarrow R_1^1(v_0) \\ & R_0^1\left(f_0^0\right) \\ \hline \therefore & R_1^1\left(f_0^0\right) \end{array}$$

Using the subscript and superscript notations makes our arguments almost illegible, so let's use more meaningful names as placeholders. Particularly, for this example,

we let Triangle stand for $R_0^1$, a for $f_0^0$, and HasThreeInteriorAngles for $R_1^1$; and thus we have:

$$\begin{array}{ll} & \forall v_0\ \mathsf{Triangle}(v_0) \Rightarrow \mathsf{HasThreeInteriorAngles}(v_0) \\ & \mathsf{Triangle}(\mathsf{a}) \\ \hline \therefore & \mathsf{HasThreeInteriorAngles}(\mathsf{a}) \end{array}$$

Now the argument is legible, and is also something that we would like, at least intuitively, to affirm. We constructed the language for first-order logic as a syntactic extension to the one for the propositional calculus, and so we might expect to still have the same inference rules available to us for FOL that we had in the propositional case. These earlier rules are not sufficient to confirm the argument just given, though. The argument above seems rather like conditional elimination, but applied to a universally quantified sentence; so, perhaps we could adopt this structure as a kind of universal conditional elimination, or universal *modus ponens*. We would represent such a rule schematically as in (3.6).

$$\frac{\forall x\ \phi(x) \Rightarrow \psi(x) \qquad \phi(c)}{\psi(c)}\ \forall\text{-}\wedge \text{ elimination} \tag{3.6}$$

Though this rule is sound, in that it couldn't ever lead us from truth to falsity, we shall soon see that our inference rules for FOL are more limited in scope, and more fundamental than, universal *modus ponens*. Nonetheless, these rules will be sufficient to allow us to quickly prove in Slate that universal *modus ponens* is entirely legitimate. But before considering such a proof, we first turn to the **semantics** of FOL, which will we allow us to understand precisely when formulas in FOL are true versus when they are false.

## 3.4 Semantics

In the previous example, we developed a rule of $\forall$-$\wedge$ elimination that let us infer from "for every $x$, if $x$ has some property $P$ then it also has some property $Q$" and "some term $t$ has property $P$" that "$t$ has property $Q$." As we have noted, even the intuitive English structure of the rule makes it clear that we want ought to affirm the validity of this rule, but we have not yet developed a way to confirm or deny this rule formally. We now develop a semantics for first-order logic that will, among other things, give us a way to ascertain with complete confidence whether a given inference rule is valid or not.

### 3.4.1 Satisfaction

We begin by saying that an **interpretation** $\mathfrak{I}$ is a pair $\langle \mathscr{D}, \mathscr{I} \rangle$ where $\mathscr{D}$ is a non-empty collection of individuals called the **domain** of $\mathfrak{I}$, and $\mathscr{I}$ is a function mapping function and relation symbols to functions and relations on $\mathscr{D}$; the function $\mathscr{I}$ is called the **interpretation function** of $\mathfrak{I}$. A **variable assignment** $\mu$ is a function from variables to $\mathscr{D}$. Just as we spoke of a truth assignment satisfying (or failing to satisfy)

a sentence in the propositional calculus, we shall speak now of an interpretation and variable mapping **satisfying** a formula of first-order logic. When an interpretation and an associated variable mapping satisfy some formula $\phi$, the bottom line is that $\phi$ is classified as true. We shall also speak of what element of the domain a term **denotes**.

First let's consider how a term $t$ is mapped to an element of the domain by an interpretation $\langle \mathscr{D}, \mathscr{I} \rangle$ and a variable mapping $\mu$. The domain element to which $t$ is mapped is called the **denotation** of $t$, which we write as $\|t\|_{\mathfrak{I},\mu}$. If $t$ is a variable $v$, then $\|t\|_{\mathfrak{I},\mu}$ is $\mu(v)$. If $t$ is a compound term $\mathsf{f}(\tau_1, \ldots, \tau_n)$, then $\|t\|_{\mathfrak{I},\mu}$ is the value of $\mathscr{I}(f)(\|\tau_1\|_{\mathfrak{I},\mu}, \ldots, \|\tau_n\|_{\mathfrak{I},\mu})$; that is, the value of the function to which $\mathsf{f}$ is mapped at the tuple of the denotations of $\tau_1, \ldots, \tau_n$.

The interpretation of atomic formulae is similar to the denotation of the compound terms. An interpretation $\mathfrak{I}$ and variable mapping $\mu$ satisfy an atomic formula $\mathsf{R}(\tau_1, \ldots, \tau_n)$, written $\mathfrak{I}, \mu \models \mathsf{R}(\tau_1, \ldots, \tau_n)$, exactly when the denotations of the terms are related by the interpretation of $\mathsf{R}$; that is $(\|\tau_1\|_{\mathfrak{I},\mu}, \ldots, \|\tau_n\|_{\mathfrak{I},\mu}) \in \mathscr{I}(\mathsf{R})$. Of course, we can also have atomic formulae of the form $t = t'$. In this case, things work just as you would expect: This type of formula holds on some interpretation $\mathfrak{I}$ just in case what $\mathfrak{I}$ maps $t$ in the domain $\mathscr{D}$ is identical with what $\mathfrak{I}$ maps $t'$ to in that domain.

Determining satisfaction for compound formulae constructed from truth-functional connectives is also straightforward: an interpretation and variable mapping satisfy a negation $\neg\phi$ exactly when they do not satisfy $\phi$; satisfy a conjunction exactly when they satisfy each conjunct; and so on.

Satisfaction of quantified formulae is more interesting. By defining satisfaction of this type we will obtain a precise meaning for sentences whose meanings we have to this point understood only intuitively. An interpretation $\mathfrak{I}$ and variable mapping $\mu$ satisfy a formula $\exists v\ \phi$ exactly when there exists a $\mu'$ just like $\mu$ except that $\mu'$ may disagree with $\mu$ on $v$; that is, it may be the case that $\mu(v) \neq \mu'(v)$, and $\mathfrak{I}$ and $\mu'$ satisfy $\phi$. It is worth pointing out that in determining whether an interpretation $\mathfrak{I}$ and variable mapping $\mu$ satisfy a formula $\exists v\ \phi$, we considered whether a $\mu'$ similar to $\mu$ but possibly disagreeing on $v$ *exists*. To determine whether $\mathfrak{I}$ and $\mu$ satisfy a universal $\forall v\ \phi$, we consider whether, for every $\mu'$ similar to $\mu$ save for $v$, $\mathfrak{I}$ and $\mu'$ satisfy $\phi$. If so, satisfaction of the universally quantified formula is achieved; otherwise it's not.

Finally, we say that an interpretation $\mathfrak{I} = \langle \mathscr{D}, \mathscr{I} \rangle$ satisfies a first-order formula $\phi$ if and only if, for every variable mapping $\mu$, $\mathfrak{I}$ and $\mu$ satisfy $\phi$. An interpretation $\mathfrak{I}$ satisfying a formula $\phi$ is said to be a **model** of $\phi$. There is no harm and much to gain by remembering that if $\mathfrak{I}$ is a model of $\phi$, the bottom line is that $\phi$ is simply *true* on this interpretation $\mathfrak{I}$. Satisfaction is defined concisely in Table 3.3.

#### 3.4.1.1 Validity

In what is a parallel with the propositional calculus, we say that a formula $\phi$ is **valid** just in case it's true on all interpretations. In our now-familiar notation, this amounts to saying that $\phi$ is valid if and only if, for all interpretations $\mathfrak{I}$, $\mathfrak{I} \models \phi$. To

| Satisfaction | | Condition |
|---|---|---|
| $\mathfrak{I},\mu \models \mathsf{R}(\tau_1,\ldots,\tau_n)$ | iff | $(\|\tau_1\|_{\mathfrak{I},\mu},\ldots,\|\tau_n\|_{\mathfrak{I},\mu}) \in \mathscr{I}(\mathsf{R})$ |
| $\mathfrak{I},\mu \models \tau_1 = \tau_2$ | iff | $\|\tau_1\|_{\mathfrak{I},\mu} = \|\tau_2\|_{\mathfrak{I},\mu}$ |
| $\mathfrak{I},\mu \models \neg\phi$ | iff | $\mathfrak{I},\mu \not\models \phi$ |
| $\mathfrak{I},\mu \models \phi_1 \wedge \ldots \wedge \phi_n$ | iff | $\mathfrak{I},\mu \models \phi_i$ for each $\phi_i$ |
| $\mathfrak{I},\mu \models \phi_1 \vee \ldots \vee \phi_n$ | iff | $\mathfrak{I},\mu \models \phi_i$ for at least one $\phi_i$ |
| $\mathfrak{I},\mu \models \phi \Rightarrow \psi$ | iff | $\mathfrak{I},\mu \models \phi$ implies $\mathfrak{I},\mu \models \psi$ |
| $\mathfrak{I},\mu \models \phi \Leftrightarrow \psi$ | iff | $\mathfrak{I},\mu \models \phi$ if and only if $\mathfrak{I},\mu \models \psi$ |
| $\mathfrak{I},\mu \models \forall v\ \phi$ | iff | $\mathfrak{I},\mu' \models \phi$ for every $\mu'$ differing from $\mu$ at most on $v$ |
| $\mathfrak{I},\mu \models \exists v\ \phi$ | iff | $\mathfrak{I},\mu' \models \phi$ for some $\mu'$ differing from $\mu$ on at most $v$ |

Table 3.3: The satisfaction relation for an interpretation $\mathfrak{I} = \langle \mathscr{D}, \mathscr{I} \rangle$ and variable mapping $\mu$. We write $\mathfrak{I} \models \phi$ iff for every $\mu$ it is the case that $\mathfrak{I},\mu \models \phi$.

say that a formula $\phi$ is valid, we write:

$$\models \phi$$

#### 3.4.1.2 Some Satisfaction and Validity Examples

You should assure verify that you can ascertain whether given formulae are satisfied by given interpretations, but constucting some examples, and rendering a provably correct verdict.

### 3.4.2 Consequence

A formula $\phi$ is a **consequence** of a set $\Phi$ of formulae, which — following suit with what we wrote in the case of the propositional calculus — we symbolize as $\Phi \models \phi$, holds just in case every interpretation $\mathfrak{I}$ which satisfies all of $\Phi$ also is a model of $\phi$.

#### 3.4.2.1 Some Consequence Exercises

1. Prove:

$$\{\phi \vee \psi\} \models \delta \text{ iff } \{\phi\} \models \delta \text{ and } \{\psi\} \models \delta$$

Solution **Proof**: For left to right, suppose that $\{\phi \vee \psi\} \models \delta$. This implies (by the definition of consequence) that every interpretation satisfying $\phi \vee \psi$ also satisfies $\delta$. Now suppose for indirect proof that $\{\phi\} \not\models \delta$. Then there is an interpretation $\mathfrak{I}^{\star}$ which models $\phi$ but renders $\delta$ false. But since $\mathfrak{I}^{star}$ satisfies $\phi$ it also satisfies $\phi \vee \psi$ (row 5 in the definition constituted by Table 3.3), and must therefore satisfy $\delta$ — contradiction. In parallel fashion we can do another sub-proof by contradiction that begins with the assumption that $\{\psi\} \not\models \delta$.

For the right-to-left direction, ...

2. Prove:
$$\{\forall y \exists x \phi(x, y)\} \not\models \exists x \forall y \phi(x, y)$$
where $\phi(x, y)$ denotes a formula $\phi$ in which the only free variables are $x$ and $y$.

3. A formula $\phi$ is **satisfiable** (written Sat $\phi$) provided that there is an interpretation which satisfies it. Prove:
$$\Phi \models \phi \text{ iff it's not that case that: Sat } (\Phi \cup \{\neg\phi\})$$

## 3.5 New Inference Rules

In keeping with what we did in the previous chapter for the propositional calculus, for each new logical symbol, that is, for $\forall$, $\exists$, and $=$, we introduce an introduction and elimination rule.

### 3.5.1 Equality Introduction

The rule of **equality introduction** allows us to introduce an equation relating any term with itself. That is, for any term $t$, we may introduce the equation $t = \ldots = t$.

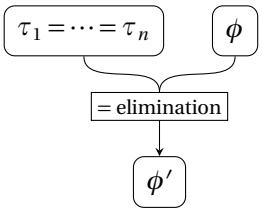
$$\boxed{\begin{array}{c} t = \cdots = t \\ = \text{intro} \end{array}} \tag{3.7}$$

It is worth noting that $t$ quite literally may be any term, whether variable or compound. The following derivations are all valid applications of equality introduction.

$$\boxed{\begin{array}{c} \mathsf{a} = \mathsf{a} = \mathsf{a} \\ = \text{intro} \end{array}} \qquad \boxed{\begin{array}{c} \mathsf{f}(\mathsf{a}, x) = \mathsf{f}(\mathsf{a}, x) \\ = \text{intro} \end{array}} \qquad \boxed{\begin{array}{c} x = x \\ = \text{intro} \end{array}} \tag{3.8}$$

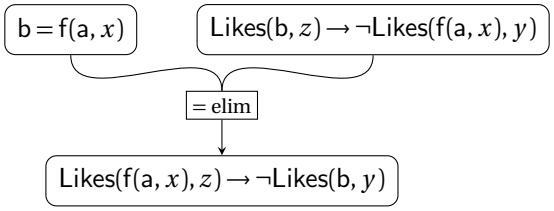
### 3.5.2 Equality Elimination

We use the rule of **equality elimination** to replace terms with equivalent terms in a formula. Given an equation $t_1 = \ldots = t_n$ and a formula $\phi$, we may infer $\phi'$ where $\phi$ is produced from $\phi$ by replacing any occurrences of any of $t_1, \ldots, t_n$ with any of $t_1, \ldots, t_n$.




$\phi'$ is identical to $\phi$ except that any occurrence of any $\tau_i$ may be replaced with any $\tau_j$ (3.9)

For instance, in (3.10), the equation $\mathsf{b} = \mathsf{f}(\mathsf{a}, x)$ allows replacement of occurrences of b and $\mathsf{f}(\mathsf{a}, x)$ in the other premise, $\mathsf{Likes}(\mathsf{b}, z) \rightarrow \neg\mathsf{Likes}(\mathsf{f}(\mathsf{a}, x), y)$.




(3.10)

$\{\tau/x\}$

### 3.5.3 The Pure Predicate Calculus

This is a suitable juncture to inform you that Slate comes ready to allow the student to explore a logic *between* the propositional calculus and FOL. This "in-between" logic is what we call the **pure predicate calculus** (for short, just PPP), which is based on the vocabulary of FOL, minus quantification. (The official, formal grammar for PPP is shown in Figure 3.1.) In other words, PPP allows one to build atomic formulas, and compound formulas from these atomics ones; but not more. We can thus say in PPP that Lucy truly like Christian (via e.g. $\mathsf{T} - \mathsf{Likes}(\mathsf{lucy}, \mathsf{christian})$), but we can't say that *everyone* truly likes Christian, since to say this more expansive thing requires a universal quantifier and corresponding variables.[2] In order to open a workspace in Slate based on the pure predicate calculus, you have only to issue the command for opening a new workspace, and to then select PREDICATE CALCULUS from among the options for the nature of the workspace. You will see the option PREDICATE CALCULUS just below PROPOSITIONAL CALCULUS, and just above FOL.

Let's go through a sample proof in PPP; doing so will serve to explain the logic, and to show the rule of identity elimination in action. In this proof, we ask the machine to automatically do reasoning at the level of the propositional calculus for us, so that we don't have to go back and operate at the level of the previous chapter. This allows the featured inferencing in our sample to be the last rule we introduced above: namely, = elimination. Specifically, in our example we shall employ the rule: PC $\vdash$, which asks Slate to go and try to find a proof on its own, and to give us the encouraging green check if one has been found. But we'll need to figure out how to use any inference rules that exceed the propositional calculus.

To begin, look at the challenge issued in Figure 3.2. Here we see that our task is to proof `GOAL`, armed with six premises. We observe that the goal is composed of three conjuncts, hence naturally hit upon the strategy of trying to proof each of these conjuncts in separate proofs, and to then draw from each of these three sub-proofs the final answer, `GOAL`.

We implement this strategy in four phases. The first phase consists of proving the rightmost conjunct in the `GOAL`: that Rensselaer is west of MIT, a geographical fact that turns out to be true in real life. To prove this intermediate conclusion,

[2] An acceptable formula in FOL would e.g. be $\forall x\ \mathsf{T} - \mathsf{Likes}(x, \mathsf{lucy})$.

Figure 3.1: Grammar of Pure Predicate Calculus Presented as a Formal Grammar.

| | | |
|---|---|---|
| *Formula* | ⇒ | *AtomicFormula* |
| | \| | (*Formula Connective Formula*) |
| | \| | ¬ *Formula* |
| *AtomicFormula* | ⇒ | (*Predicate* $Term_1 \ldots Term_k$) |
| *Term* | ⇒ | (*Function* $Term_1 \ldots Term_k$) |
| | \| | *Constant* |
| | \| | *Variable* |
| *Connective* | ⇒ | $\wedge \mid \vee \mid \rightarrow \mid \leftrightarrow \mid$ |
| *Predicate* | ⇒ | $P_1 \mid P_2 \mid P_3 \ldots$ |
| *Constant* | ⇒ | $c_1 \mid c_2 \mid c_3 \ldots$ |
| *Variable* | ⇒ | $v_1 \mid v_2 \mid v_3 \ldots$ |
| *Function* | ⇒ | $f_1 \mid f_2 \mid f_3 \ldots$ |

we can extract the relevant conjunct from `PREMISE1` by propositional reasoning (notice that we don't bother to cite the particular inference rule that justifies this step: conjunction elimination), and then use the recently introduced rule of identity elimination to derive from this conjunct the part of `GOAL` that we are aiming for. The completion of the first phase is shown in Figure 3.3.

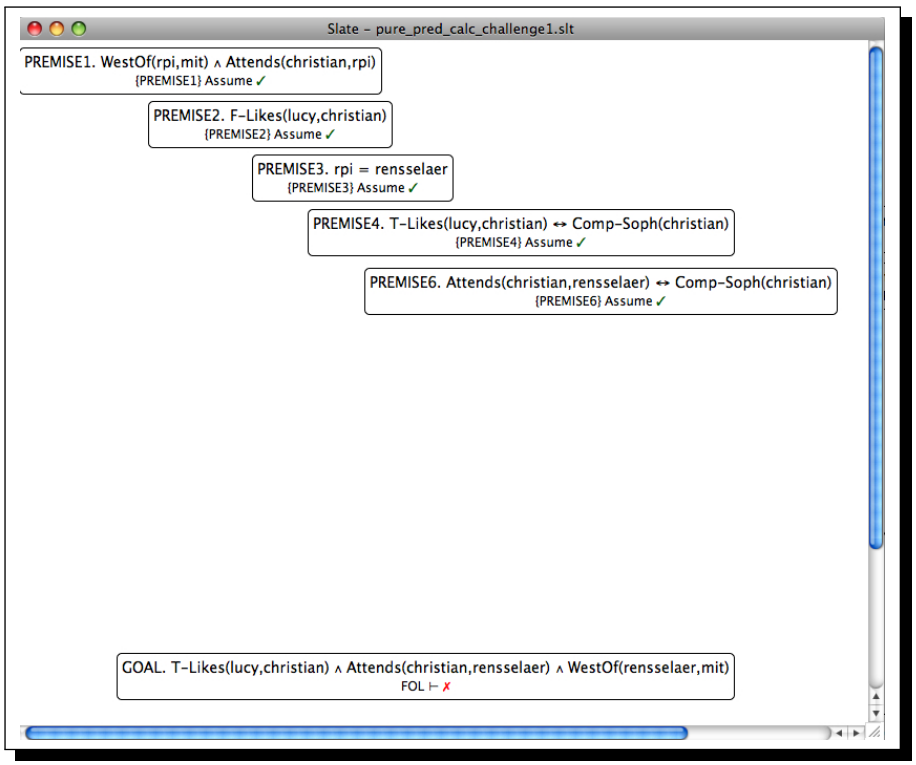
In phase 2, we prove the middle conjunct of GOAL; that is, that Christian attends Rensselaer. This is no harder than reaching phase 1; indeed, we make use of the same inference (= elimination). Using comments now to annotate the proof, you can find the situation at the end of phase 2 shown in Figure 3.4.

In phase 3, we chain through two of the biconditionals to arrive at the intermediate conclusion that Lucy truly likes Christian. (We let Slate find a proof automatically. Were we to do it ourselves, what rule of inference in the propositional calculus would be used?) And finally comes the planned phase 4, in which we infer the goal from the three atomic formulae we have deduced. Figure 3.5 shows the result of carrying out the final two phases.

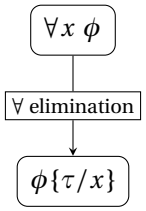
### 3.5.4 Universal Elimination

The rule **universal elimination** allows us to infer information about a specific individual when we already know more general information. For instance, from "all men are mortal," or, more specifically, "for every $x$, if $x$ is a man, then $x$ is mortal," we may infer "if Socrates is a man, then Socrates is mortal." We have eliminated the quantification "for every $x$" and replaced all bound occurrences of $x$ with "Socrates." In formulae of first-order logic, from a formula $\forall x\ \phi$, universal elimination allows us to drop the $\forall x$ prefix and substitute a term $t$ for $x$ in $\phi$. This rule is given

Figure 3.2: A Proof Challenge in the Pure Predicate Calculus

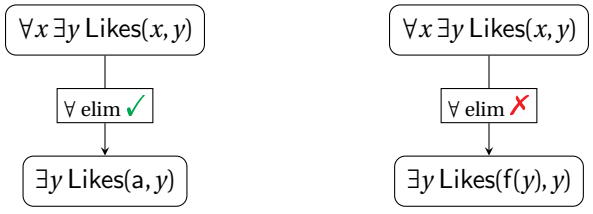



schematically in (3.11).



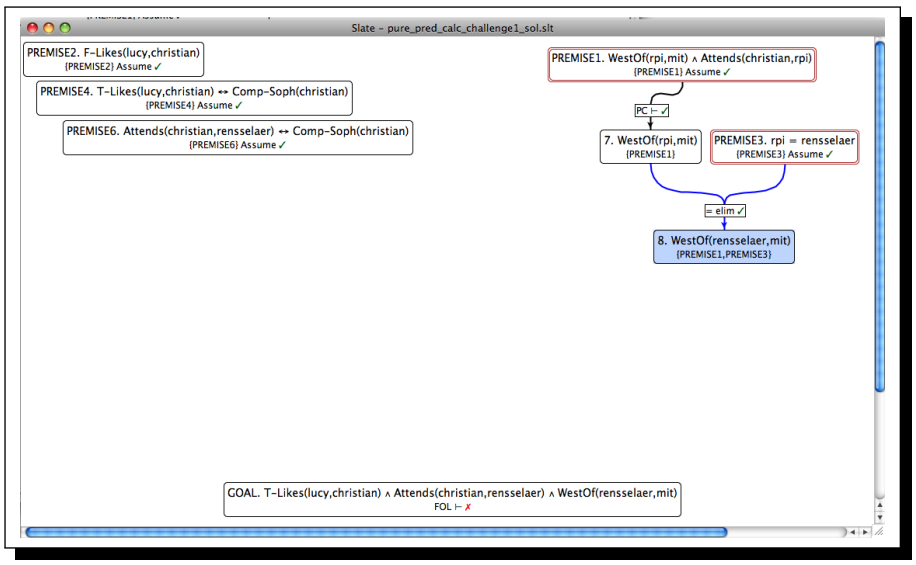

(3.11)

It is important to note that universal elimination is defined in terms of a substitution (as is **existential introduction**). Such a definition restricts the applications of universal elimination in some ways. For instance, compare the following two attempted applications of universal elimination.




(3.12)

Figure 3.3: Completion of Phase 1 in Meeting the Proof Challenge




The application on the left is valid, but the application on the right is not, as $\mathsf{f}(y)$ cannot be substituted for $x$ in $\exists y\ \mathsf{Likes}(x, y)$. Fortunately, the potential difficulties in rules for quantification do not actually arise much in proof construction. Indeed, running up against them may be an indication that another proof tactic is needed.

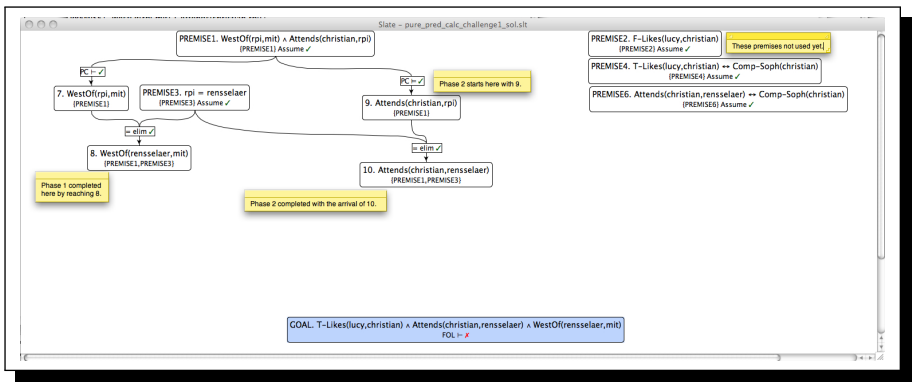
### 3.5.5 Existential Introduction

The rule **existential elimination** allows us to move from observations of particular individuals to more general claims of existence. For instance, knowing that "the number two is even and prime," we may infer that "there exists a number which is even and prime." Formally, from a formula $\phi\{\tau/x\}$ we may infer $\exists x\ \phi$, and this is given schematically in (3.13).

$$\begin{array}{c} \phi\{\tau/x\} \\ \hline \exists \text{ introduction} \\ \hline \exists x\, \phi \end{array} \tag{3.13}$$

It may seem strange that existential introduction yields a formula which, when substitution is applied to it, is the premise. Doesn't this have things backward? We described substitution as a way to replace a variable in a formula, and such a description may lead to our thinking of substitution as a one-way process. However, substitution is simply a mathematical function of three arguments: a term, a variable, and a formula; and the value of this function is a formula. As such, a

Figure 3.4: Completion of Phase 2 in Meeting the Proof Challenge




formula $\phi$ may be the result of substituting a number of terms for a number of variables. For instance, for $\phi_1 = \mathsf{Likes}(x, \mathsf{b})$ and $\phi_2 = \mathsf{Likes}(\mathsf{a}, y)$, it is the case that $\phi_1\{\mathsf{a}/x\} = \mathsf{Likes}(\mathsf{a}, \mathsf{b}) = \phi_2\{\mathsf{b}/y\}$.[3] To illustrate this particular example, both of the following are valid applications of existential introduction.

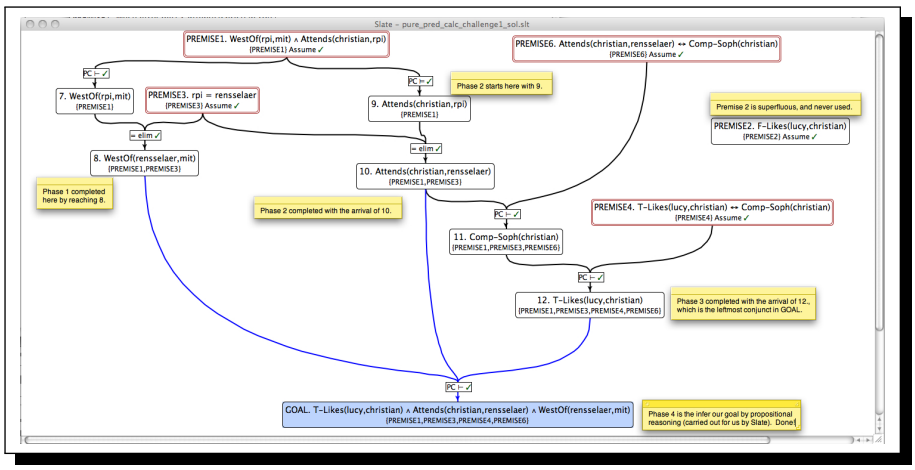

(3.14)

In the example on the left, the relevant substitution is $\{\mathsf{a}/x\}$, while the relevant substitution for the right is $\{\mathsf{b}/y\}$. The terms a and b in these substitutions are often called **witness terms**, or simply **witnesses**, as it is their "testimony" that justifies the application of existential introduction.

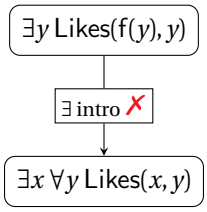
As with universal elimination, that existential introduction is defined in terms of a substitution has some important consequences. For instance, existential introduction will not let any variables escape, so to speak; thus (3.15) is not a valid application of existential introduction, for the $y$ in the witness term $\mathsf{f}(y)$ would escape its universal quantification. Another way to recognize this is that $\mathsf{f}(y)$ cannot be

[3] fact, substitutions that may be applied to distinct formulae to produce the same formula play an important role in automated theorem proving, particularly the **resolution algorithm** which is used by some versions of the "oracles" in Slate.

Figure 3.5: Completion of Phases 3 & 4 in Meeting the Proof Challenge




substituted for $x$ in $\forall y\,\mathsf{Likes}(x,y)$ because the free occurrence of $y$ would be captured.



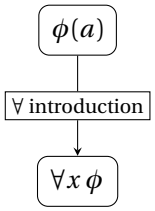

(3.15)

### 3.5.6 Universal Introduction

Inferring specific information from the general case, as universal elimination allows us to do, is relatively easy. (Actually, to put it barbarically, universal elimination is rather brainless.) To derive general knowledge, however, is more difficult. It's not valid to simply move from the specific, say, "it rained today," to the general, for instance "it rains every day." Likewise, if we happen to know that a particular llama, Alvin, is smart, we can't infer that *all* llamas are smart. To derive universally quantified formulae, we must choose a new, arbitrary name (or constant) $a$, and from that point derive some formula $\phi(a)$ that includes that name, and serves the purposes at hand. Provided that we use no information that would associate the chosen arbitrary name $a$ with some particular individual (which would instantly render that name *non*-arbitrary), and that we replace all occurrences of $a$ in $\phi a$ with the variable $\varphi$ of our choice, we may proceed to replace all occurrences of $a$ in $\phi(a)$ with $\varphi$ to yield $\forall\varphi\phi$.

Our grammar of first-order logic (Table 3.1) constrains the symbols appearing immediately after $\forall$ in a formula to be (a) variable(s), and so our arbitrary names will

be free variables or constants.[4] Since the arbitrary name is a free variable or constant that may appear in other formulae, we must obey a constraint. The constraint is this: We can't use information that would associate the arbitrary name with a particular individual. This is enforced by requiring that the arbitrary name does not appear free within any assumption in scope of the formula to which we add the universal prefix. Universal introduction thus is given schematically in (3.16).

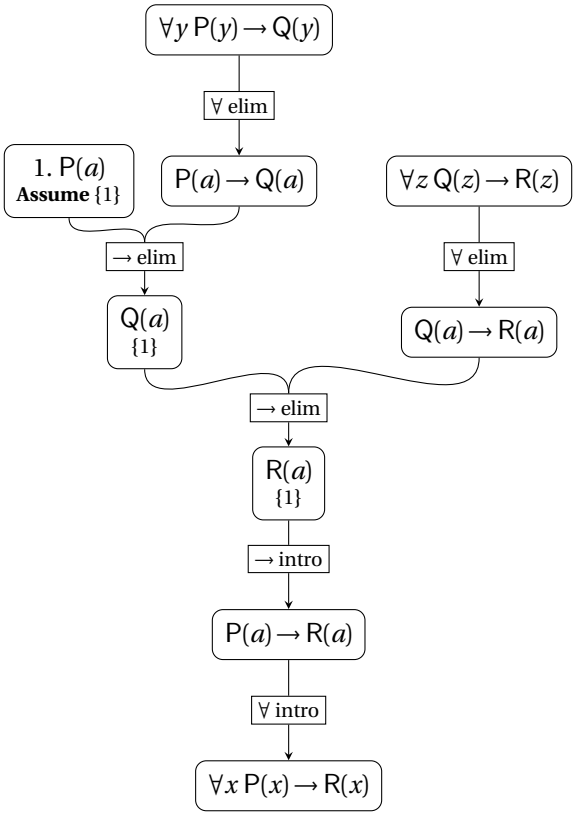



provided that $a$ does not appear free in any in-scope assumption supporting $\phi$, and that no occurrence of $a$ appears in the inferred $\forall x\,\phi$ (3.16)

For instance, in the proof shown in (3.17), **universal introduction** could not be applied to $\mathsf{P}(a)$, $\mathsf{Q}(a)$ or $\mathsf{R}(a)$ (with a variable substituted for $a$ and a quantifier introduced to bind that variable), as all three of these formulae are derived within the scope of the assumption $\mathsf{P}(a)$. Once that assumption is discharged to produce a conditional, however, $a$ no longer appears free in any in-scope assumption, and so it's valid to proceed from $\mathsf{P}(a) \rightarrow \mathsf{R}(a)$ by universal introduction to $\forall x\,\mathsf{P}(x) \rightarrow \mathsf{R}(x)$. Note as well that in (3.17) the formula that is the end result, as required, contains

[4] If you are enrolled in a course that uses the present textbook, your instructor may prefer not to make use of, or even to allow, free variables.

no remaining occurrence of $a$.

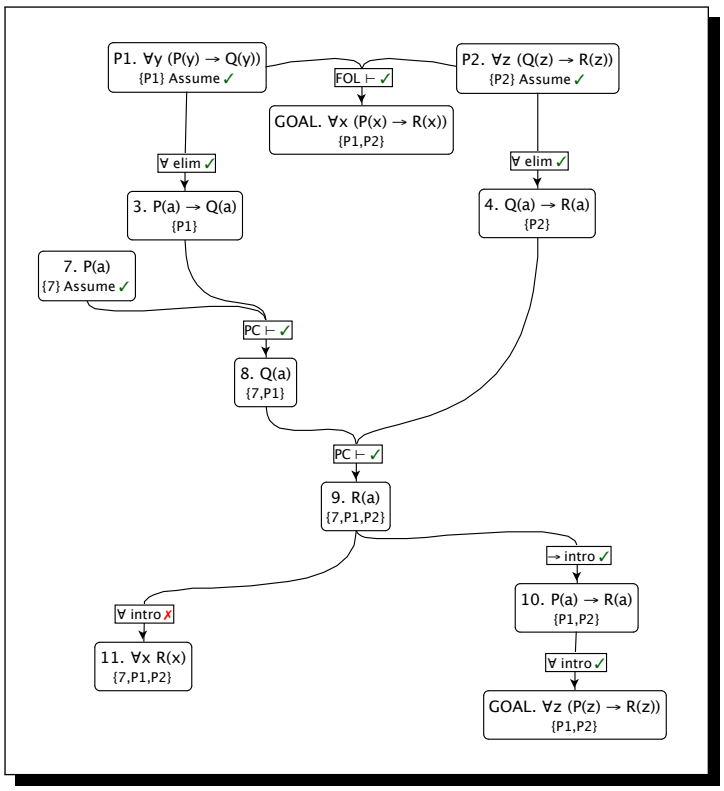



(3.17)

To hammer home that a correct use of universal introduction requires that variables or constants replaced with new variables within the scope of a new universal quantifier not be present in an undischarged assumption above, look at Figure 3.6, which shows a Slate screenshot of a slight variant of the proof shown in (3.17). On the bottom right, we see that success has been secured: the desired goal has been correctly obtained. The bottom left tells a very different story. There, an attempt has been made to deduce that everything is an R by universal introduction — a move which Slate declares to be in error. Do you see the problem? Look carefully at Figure 3.6. If you inspect the node that Slate prohibits, and look up "the chain" on which is rests, you will find that, sure enough, the constant a appears in an undischarged assumption, a violation of the condition that when a universally quantified variable is introduced in place of a constant (or a free variable), that constant cannot appear in any undischarged assumption that supports the intended inference. Notice that in the list of assumptions of the node in question, the numeral '7' appears, and that node with that label hasn't been discharged.

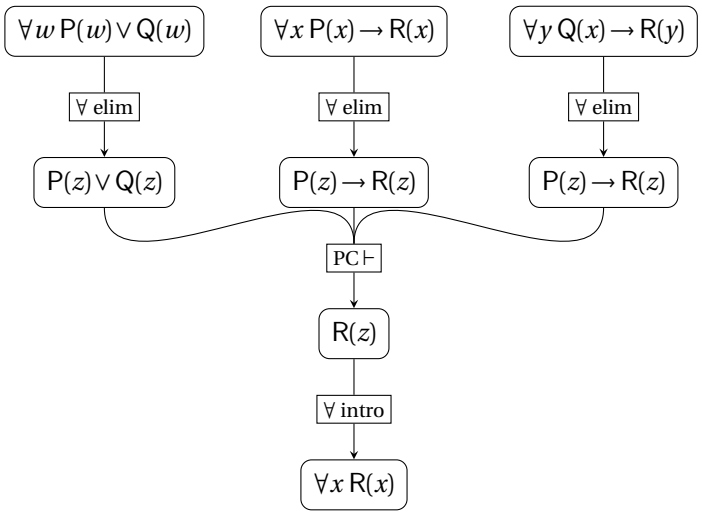
In the proof shown in (3.17), the arbitrary name was introduced in the assumption. Proofs of this form will tend to have results which are universal formulae whose quantified formulae are conditionals. However, we may also derive universals the

Figure 3.6: Universal Introduction: A Right Way and a Wrong Way




quantified formulae of which are not conditionals. For instance, consider the proof in (3.18), in which there are no assumptions with free variables at all, and whose

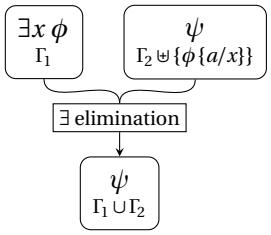
conclusion is a universal quantification over $x$.




(3.18)

### 3.5.7 Existential Elimination

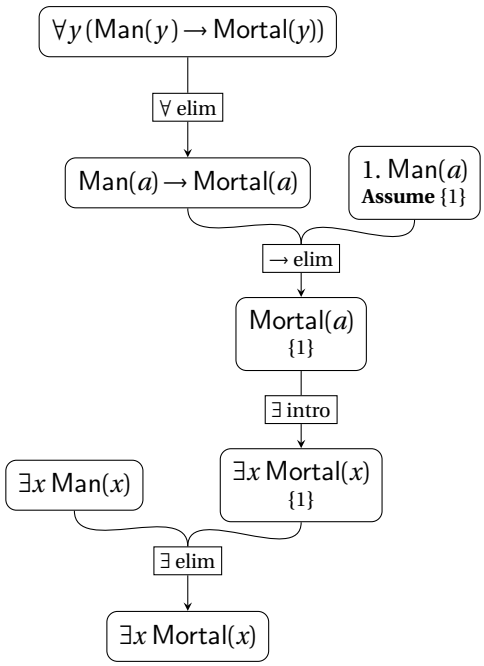
The **existential elimination** rule allows us to infer formulae from others that assert the existence of some objects. The technique that existential elimination regiments is the following: Knowing that there exists an individual with some property, pick an arbitrary name and assume that the individual denoted by that name has the particular property in question. Any formulae which follow from this assumption and do not contain the arbitrary name are valid conclusions of the existential claim. The justification for this is that while the name chosen was arbitrary, the existence claim guarantees that there is at least one individual with the given property—as it's often said, the "witness" for the existence claim in question; and since the conclusion itself does not reference this name, the arbitrary name is not used past the point where it is taken as denoting an individual with the particular property. This rule is given schematically in (3.19).




provided that $a$ does not appear free in $\Gamma_1$, $\Gamma_2$, or $\psi$ (3.19)

(3.20) shows how we could proceed from the premises $\exists y\,\mathsf{Man}(y)$ and $\forall y(\mathsf{Man}(y) \rightarrow \mathsf{Mortal}(y))$ to the intuitive conclusion, $\exists y\,\mathsf{Mortal}(y)$. As with universal introduction, we find that while an intuitive description of the rule involves a notion of arbitrary name, to apply existential elimination to an existential $\exists x\,\phi$, the choice is fixed: if

$a$ is the name we choose for the arbitrary individual, $\phi\{a/x\}$ is the formula which must be assumed.




(3.20)

We might render this as an informal proof couched for the most part in English as follows:

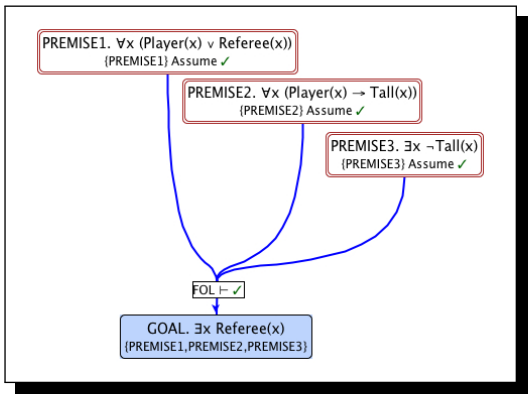
> "There is an individual which is a man. Let's call that individual, for the moment, *a*. Since all men are mortal, and since *a* is a man, *a* is mortal. Since *a* is mortal, there exists some mortal (viz., *a*). Thus, we have inferred from our premises, along with our temporarily naming an individual, that there is a something which is mortal. This conclusion does not depend on the particular name *a* that we chose, since that selection was wholly arbitrary (e.g., we could have just as well have selected *b*), and thus we conclude that there is, in fact, an individual which is mortal."

Note that while in reading this informal proof, we understand that at least one individual is both mortal and a man, but this is not immediately apparent from the premise $\exists x\,\mathsf{Man}(x)$ and the conclusion $\exists y\,\mathsf{Mortal}(y)$. We could have inferred a stronger conclusion, namely $\exists y\,\mathsf{Man}(y) \wedge \mathsf{Mortal}(y)$, which affirms that there is an individual who is both a man and mortal. Still, it's important to note that while the English rendering of $\exists x\,\mathsf{Man}(x) \wedge \exists y\,\mathsf{Mortal}(y)$ as "there is an individual which is a man, and there is an individual which is both a man and mortal" might suggest two distinct individuals, no such claim is made by the formula.

To ensure that you have a good understanding of existential elimination, let's go through a proof in Slate that exploits this rule. To begin, the challenge is issued as shown in Figure 3.7. The backstory is that we are reasoning about a world circumscribed by a basketball court, the only denizens of which are either players

or referees. We have three premises to leverage. The first expresses the idea that everything on the court is either a player or a referee; the second that all players are tall; and finally the third that there exists a person on the court who isn't tall. Notice (Fig. 3.7) that Slate has no problem finding a proof, as is indicated by the green arrow that's returned. But how are *we* to produce a proof?

Figure 3.7: Starting Position in a Proof Challenge




The key inference in the proof is going to be existential elimination. The core idea is this: If we invoke an arbitrary object to be a witness to the assertion, made in PREMISE3, that there's a player on the court who isn't tall, we should be able to show, with relative ease, that that person, call him $a$, is a referee—for after all, we're told that players are tall, and that everyone on the court is either a player or a referee. In order to implement this plan, our first step is announcing the selection of our witness, which is shonwn in the assumption highlighted in Figure 3.8. We use '$a$' as the arbitrary name of our witness.

Following on this, we suitably eliminate the universal quantifiers, and instantiate the variable $x$ to $a$ in both PREMISE1 and PREMISE2. We thus arrive at the graph shown in Figure 3.9.

At this point we are ready to capitalize on the fact that we learned how to reason in the propositional calculus, for in one step, we can use Slate to certify for us that we can deduce that $a$ must be a referee. The situation is now as depicted in Figure 3.10.

Our next move is simple: We infer by existential introduction from the fact that our witness $a$ is a referee that there exists some $x$ which is a referee; see Figure 3.11. Notice that at this stage our reasoning still depends upon the assumption that $a$ isn't tall (node 5). At this point we complete the proof by deducing, using existential elimination, that there's some $x$ which is a referee. This discharges the assumption made in order to set up use of existential elimination (node 5), and we're done; see Figure 3.12.

Figure 3.8: First Step Toward Meeting the Challenge

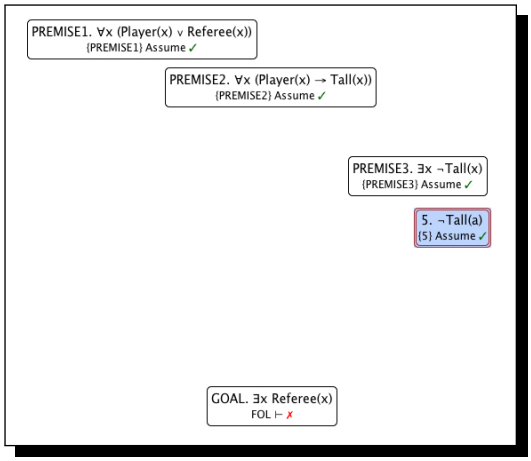
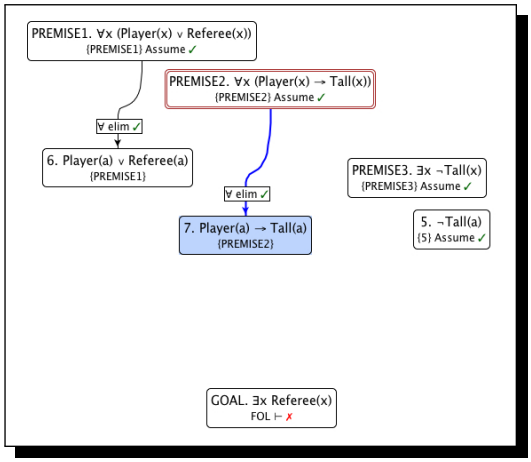



Figure 3.9: The Situation After Second and Third Steps




## 3.6 Disproofs; Modern Model Finders

Consider the following argument:

**A Seductive Syllogism** (SS)

(1) All whales are ocean dwellers.
(2) Some ocean dwellers are orcas.
∴ (3) Some whales are orcas.

The vast majority of those who haven't learned beginning deductive logic regard this to be a valid argument: they believe that (3) follows logically from the combination

Figure 3.10: We Rely on Slate *qua* Oracle at the Level of the Prop. Calc.

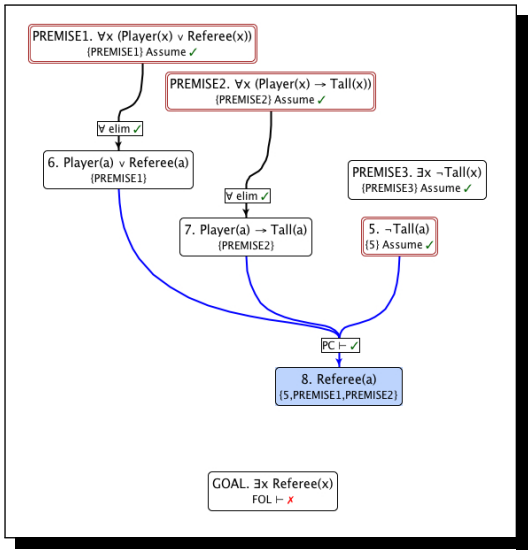
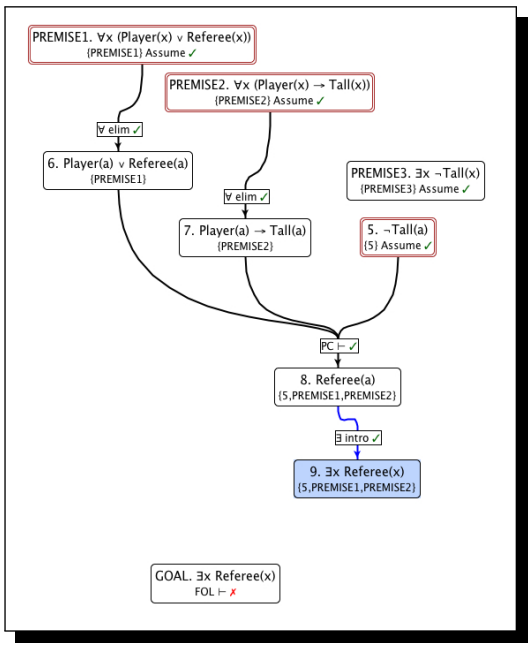


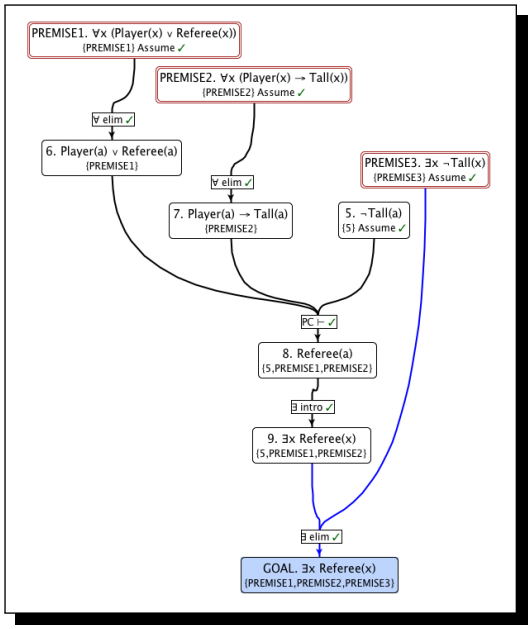

Figure 3.11: We Rely on Slate *qua* Oracle at the Level of the Prop. Calc.




of (1) and (2).[5] But of course, courtesy of the training you have already received by

[5]This holds whether or not, by the way, these poor souls have advanced degrees; *what* one studies

Figure 3.12: Finished!




way of the present volume and its concomitant software, you see that this argument is invalid: the conclusion, (3), can't be deduced from the combination of (1) and (2). Expressed using the now-familiar concept of entailment, you see that (3) isn't entailed by {(1), (2)}; that is, you apprehend

$$\{(1),(2)\} \nvdash (3).$$

But do you see exactly *why* entailment fails here? Or, put another way and more concretely, can you *prove* that this entailment fails?

Of course, if you have Slate handy, a request to the FOL entailment oracle that it attempt to find a proof of (3) from {(1), (2)} returns the response that no such proof can be found. But what if one of the —- forgive us — benighted folks who are seduced by this syllogism, caught firmly in grip of the seduction, challenged you to *prove* that it's invalid? If you accepted the challenge, what would you proceed to do? Well, you would first cook up an example: an interpretation $\mathfrak{I}$ on which both (1) and (2) are true, but (3) is false. And then you would proceed to explain that $\mathfrak{I}$ satisfies (1) and (2), but dictates that (3) is false. Since any deduction of a statement $\sigma$ from a collection of statements $\Sigma$ ensures that there is no situation in which all of $\Sigma$ are true but $\sigma$ is false, you would be done — and hopefully the seduced skeptic would be enlightened.

This disproof points toward a general recipe $R$ for providing a proof of $\Phi \nvdash \phi$. $R$ has two main steps:

---

may well be exceed in importance how *much* one studies.

Step 1 Discover an interpretation $\mathfrak{I}$ on which all of $\Phi$ are true, but $\phi$ is false. If these two conditions are met, we say that $\mathfrak{I}$ is a **counter-example** of $\Phi \vdash \phi$.

Step 2 Explain that $\mathfrak{I}$ does this, and that in light of the fact that it does, $\Phi \nvdash \phi$ holds. (Sometimes this second step is so obvious that it's omitted.)

Fortunately, the first step in *R* can nowadays often be carried out automatically by intelligent software; by what are called **model finders**. Such machine intelligence can allow you to quickly enlighten those who are duped by arguments that seem valid, but aren't. We can show this in the case at hand, by using the Mace4 model finder, which is free and readily available.[6] When Mace4 receives a request to search for a counter-example of SS, here's what it instantly returns:[7]

```
% Interpretation of size 2

c1 : 0

OceanDweller :
        0 1
   --------
        T F

Orca :
        0 1
   --------
        T F

Whale :
        0 1
   --------
        F F
```

What does this output mean? It's really quite simple to understand what Mace4 has cooked up and presented, once one takes note of a key fact: Counter-examples devised by Mace4 invariably have as their domains some subset of the natural numbers, $\mathscr{N}$. In the present case, Mace4 tells us that an "Interpretation of size 2" has been used, which means that the domain is specifically the set {0, 1}. In addition to this information, Mace4 also informs us how it's interpreting each of the three relations found in SS. It does this by providing, for each relation, a simple table, which as you'll note has a top and a bottom in each case. The top just lists each member of the domain, starting from 0, and then incrementing by one until the final member of the domain is reached. In the domain in the counter-example before us, there are, as we've noted, just two members. Let's zoom in on the table Mace4 gave for Whale:

[6] Mace4 can currently be otained free of charge by going to http://www.cs.unm.edu/~mccune/prover9.

[7] We have taken the liberty of changing how Mace4 expresses TRUE and FALSE, by using 0 for the former and 1 for the latter.

```
Whale :
        0 1
   --------
        F F
```

This tells us, specifically, that member 0 of the domain is not a whale (since right below 0 appears 'F' for FALSE), and that neither is 1 (since right below it is also the assignment of 'F' for FALSE). The same parse can be applied to the other two tables, and you should not proceed to the next sentence until you have studied them, and understand precisely what they communicate. ... Now, with the example Mace4 has supplied labeled $C_1$, we can follow recipe $R$ in order to produce an outright disproof:

> **Disproof**: We are to show that SS is invalid; i.e., that (3) cannot be deduced from the combination of (1) and (2). We appeal to $C_1$, and need only show that on $C_1$ both premises hold while the conclusion, (3), is false. It's clear that (3) is false on $C$, for the simple reason that in $C$ — as the table for the relation Whale regiments — nothing is a whale; therefore it's not true that something is both a whale and an orca. As to the verification of (1), since there are no whales, it's vacuously true that all whale are ocean-dwellers. (More precisely, since neither 0 nor 1 is a whale on $C_1$, for all assignments of the universally quantified variable $x$ in the FOL-formalization of (1), the antecdent of (1) is false, and since a conditional with a false antecedent is true, (1) is true on $C$.) Premise (2) is satisfied by $C_1$, because 0 is both an ocean-dweller and an orca; hence some ocean-dwellers are orcas. **QED**

Let's try our disproving skill on something a bit more challenging. Consider the following two premises:

(1′) $\forall x\, \forall y\, \big[\big(\mathsf{Aardvark}(x) \wedge \mathsf{Camel}\big(y\big)\big) \Rightarrow \mathsf{Larger}\big(x, y\big)\big]$

(2′) $\forall x\, \forall y\, \big[\big(\mathsf{Camel}(x) \wedge \mathsf{Llama}\big(y\big)\big) \Rightarrow \mathsf{Larger}\big(x, y\big)\big]$

Does it follow from these two formulae that the following formula must hold?

(3′) $\forall x\, \forall y\, \big[\big(\mathsf{Aardvark}(x) \wedge \mathsf{Llama}\big(y\big)\big) \Rightarrow \mathsf{Larger}\big(x, y\big)\big]$

We can once again ask Mace4 if it can find a counter-example. When we do this, we get back a counter-example:

```
% Interpretation of size 2

c1 : 0

c2 : 0

Aardvark :
        0 1
   --------
```

```
        1 0

Camel :
        0 1
   --------
        0 0

Llama :
        0 1
   --------
        1 0

Larger :
      | 0 1
   ---+----
    0 | 0 0
    1 | 0 0
```

As you can see, this counter-example, which we'll call $C_2$, says that nothing in the domain (which is once again $\{0,1\}$) is larger than anything in the domain, fact that can anchor a second disproof: a proof that $\{(1'),(2')\} \nvdash (3')\}$:

> **Disproof**: In $C_2$, regardless of what members of the domain $x$ and $y$ are assigned to, $\mathsf{Larger}(x,y)$ will be false. Given this, if for any assignment to these two variables it's the case that what $x$ is assigned to is classified as an aardvark, and what $y$ is assigned to is classified as a llama, $(3')$ will be false. With both $x$ and $y$ assigned to 0, as the relevant two tables in $C_2$ show, this is what we have. If now the truth of $(1')$ and $(2')$ are ensured, we have shown that $C_2$ gives us what we need to establish that $(3')$ isn't entailed by $\{(1'),(2')\}$. Since neither 0 nor 1 are ever classified as camels, the antecedent in $(1')$, and the antecedent in $(2')$, will always both be false — which means that $(1')$ and $(2')$ will always both be true. **QED**

## 3.7 Review & Self-Assessment via Selecting Axioms

Now let's allow you to see how much of the foregoing you have learned, by pulling a page from polymath and pedagogical pioneer Patrick Suppes.[8] Your challenge, in general, is to "select axioms." More specifically: You will be given a finite set $\mathscr{F}$ of $n$ first-order formulae. Your task is to select at most $n-1$ formulae from $\mathscr{F}$ as "axioms" ($\mathscr{A}$), from which you must prove the formulae remaining from $\mathscr{F}$. We denote these remaining formulas by '$\mathscr{F}'$.' We can modify[9] notation used above to sum up your goal in each finding-axiom exercise as:

$$\mathscr{A} \vdash \mathscr{F}'.$$

We write '$|\mathscr{X}|$' to indicate the number of formulae in $\mathscr{X}$. And now we can turn to your first challenge.

### 3.7.1 Selecting Axioms Challenge 1

The set $\mathscr{F}$ is given by the list of formulae immediately below. Note that $|\mathscr{F}| = 8$. Your challenge is to select four axioms from $\mathscr{F}$ in order to stock $\mathscr{A}$. Of course, this implies that $|\mathscr{A}| = 4$ and $|\mathscr{F}'| = 4$. Show in a single Slate workspace that:

$$\mathscr{A} \vdash \mathscr{F}'.$$

To save you some keystroke tedium, and to minimize error in data entry, we have provided you with a Slate file in which the following octet has been pre-encoded: `selecting_axioms_challenge1.slt`.

$F_1$ $\forall x, y, z\, [(x > y \wedge y > z) \rightarrow x > z]$

$F_2$ $\forall x, y\ x > y \rightarrow \neg(y > x)$

$F_3$ $\forall x\, \neg(x > x)$

$F_4$ $\forall x, y\, [(x \geq y \rightarrow \neg(y > x)]$

$F_5$ $\forall x, y\, [\neg(x > y) \rightarrow (y \geq x)]$

$F_6$ $\forall x, y\, (x \geq y \vee y \geq x)$

$F_7$ $\forall x, y, z\, [(x \geq y \wedge y \geq z) \rightarrow x \geq z]$

$F_8$ $\forall x\ x \geq x$

[8] The present section draws specifically from (Goldberg & Suppes 1972). Suppes declares therein that he in turn is pulling a page from the great math educator R.L. Moore (who was also a great mathematician in his own right). Moore's method for teaching math was one he encapsulated in the dictum: "The student who is taught the best is taught the least." It should be noted that Suppes refers to the problems introduced in this section as *finding* axioms exercises. But since the formulae from which your axioms come are supplied at the outset, it's perhaps a tad more accurate to say that you *select* axioms.

[9] Recall that normally the right side of the single turnstyle $\vdash$ is only allowed to denote an individual formula, not a set of formulae. What we write here is hence short for

$$\mathscr{A} \vdash F_1' \text{ and } \mathscr{A} \vdash F_2' \text{ and } \ldots \mathscr{A} \vdash F_k',$$

where the $F_i'$ exhaust the formulae in $\mathscr{F}'$.

### 3.7.2 Selecting Axioms Challenge 2

For our second selecting-axioms challenge, the set $\mathscr{F}$ is given by the list of formulae immediately below. Note that this time $|\mathscr{F}| = 14$. Your challenge is to select nine axioms from $\mathscr{F}$ in order to stock $\mathscr{A}$, and derive the remaining five. Hence $|\mathscr{A}| = 9$ and $|\mathscr{F}'| = 5$. Try now to show in a single Slate workspace that:

$$\mathscr{A} \vdash \mathscr{F}'.$$

To save you once again from keystroke boredom, and to minimize error in data entry, you can find a Slate file in which the following list of formulae has been pre-encoded: `selecting_axioms_challenge2.slt`.

$F_1$ $\forall x, y\ (x+1 = y+1 \rightarrow x = y$

$F_2$ $\forall x\ x+1 \neq 0$

$F_3$ $\forall x, y\ x \neq y \rightarrow x+1 \neq y+1$

$F_4$ $\forall x\ x+0 = x$

$F_5$ $\forall x\ x+1 \neq x$

$F_6$ $\forall x, y\ x+(y+1) = (x+y)+1$

$F_7$ $\forall x, y\ x+y = y+x$

$F_8$ $\forall x\ 0+x = x$

$F_9$ $\forall x\ x \times 0 = 0$

$F_{10}$ $\forall x, y\ x \times (y+1) = (x \times y)+x$

$F_{11}$ $\forall x, y\ x \times y = y \times x$

$F_{12}$ $\forall x\ 0 \times x = 0$

$F_{13}$ $\forall x\ x \times 1 = x$

$F_{14}$ $0 \neq 1$

### 3.7.3 Selecting Axioms Challenge 3

For our third selecting-axioms challenge, the set $\mathscr{F}$ is given by the list of formulae immediately below. Note that this time $|\mathscr{F}| = 25$. Your challenge is to select eight axioms from $\mathscr{F}$ in order to stock $\mathscr{A}$, and derive the remaining 17. Hence $|\mathscr{A}| = 8$ and $|\mathscr{F}'| = 17$. Try now to show in a single Slate workspace that:

$$\mathscr{A} \vdash \mathscr{F}'.$$

To save you once again from keystroke boredom, and to minimize error in data entry, you should obtain from your instructor a Slate file (`selecting_axioms_challenge3.slt`) in which the following list of formulae has been pre-encoded:

$F_1$ $\forall x, y, z\ (x \subseteq y \wedge y \subseteq z) \rightarrow x \subseteq z$

$F_2$ $\forall x, y\ (x \subseteq y \wedge y \subseteq x) \rightarrow x = y$

$F_3$ $\forall x\ x \subseteq x$

$F_4$ $\forall x, y\ x \subseteq y \rightarrow x \cap y = x$

$F_5$ $\forall x, y\ x \cap y = x \rightarrow x \subseteq y$

$F_6$ $\forall x\ x \cap x = x$

$F_7$ $\forall x\ x \cup x = x$

$F_8$ $\forall x, y\ x \cap y = y \cap x$

$F_9$ $\forall x, y\ x \cup y = y \cup x$

$F_{10}$ $\forall x, y, z\ x \cap (y \cap z) = (x \cap y) \cap z$

$F_{11}$ $\forall x, y, z\ x \cup (y \cup z) = (x \cup y) \cup z$

$F_{12}$ $\forall x, y\ x \cap (x \cup y) = x$

$F_{13}$ $\forall x, y\ x \cup (x \cap y) = x$

$F_{14}$ $\forall x, y, z\ x \subseteq y \wedge x \subseteq z \rightarrow x \cap y = x \cap z$

$F_{15}$ $\forall x, y\ x \subseteq y \rightarrow x \cup y = y$

$F_{16}$ $\forall x, y\ x \cup y = y \rightarrow x \subseteq y$

$F_{17}$ $\forall x, y\ x \cap y \subseteq x$

$F_{18}$ $\forall x, y\ x \subseteq x \cup y$

$F_{19}$ $\forall x, y, z\ x \subseteq z \wedge y \subseteq z \rightarrow x \cup y \subseteq z$

$F_{20}$ $\forall x, y, z\ x \subseteq y \wedge x \subseteq z \rightarrow x \subseteq y \cap z$

$F_{21}$ $\forall x, y\ x \cap y \subseteq x \cap y$

$F_{22}$ $\forall x, y\ x \cap (x \cap y) = x \cap y$

$F_{23}$ $\forall x, y, z\ x \cap (y \cap z) \subseteq y$

$F_{24}$ $\forall x, y, z\ x \cap y \subseteq x \cup z$

$F_{25}$ $\forall x, y, z\ x \subseteq (y \cap z) \cup x$

### 3.7.4 Selecting Axioms Challenge 4

In our fourth selecting-axioms challenge, $\mathscr{F}$ is the list of formulae immediately below. This time $|\mathscr{F}| = 11$, and your challenge is to select five axioms from $\mathscr{F}$ to constitute $\mathscr{A}$, and derive the remaining six. Therefore $|\mathscr{A}| = 5$ and $|\mathscr{F}'| = 6$. Try now to show in a single Slate workspace that:

$$\mathscr{A} \vdash \mathscr{F}'.$$

To advance your cause, specifically to facilitate the data-entry side, you should ask your instructor for a pre-engineered Slate file (`selecting_axioms_challenge4.slt`) in which the following list of formulae has been pre-encoded:

$F_1$ $\forall x$ $\mathsf{Between}(x, x, x)$

$F_2$ $\forall x, y$ $\mathsf{Between}(x, y, x) \rightarrow x = y$

$F_3$ $\forall x, y, z$ Between$(x, y, z) \rightarrow$ Between$(z, y, x)$

$F_4$ $\forall x, y, z$ $x = y \rightarrow$ Between$(x, y, z)$

$F_5$ $\forall x, y, z, w$ (Between$(x, y, w) \wedge$ Between$(y, z, w)) \rightarrow$ Between$(x, y, z)$

$F_6$ $\forall x, y, z, w$ $(\neg(y = z) \wedge$ Between$(x, y, z) \wedge$ Between$(y, z, w)) \rightarrow$ Between$(x, y, w)$

$F_7$ $\forall x, y, z, w$ (Between$(x, y, z) \wedge$ Between$(x, w, z)) \rightarrow$ (Between$(y, w, z) \vee$ Between$(w, y, z)$)

$F_8$ $\forall x, y, z$ (Between$(x, y, z) \wedge$ Between$(y, x, z)) \rightarrow x = y$

$F_9$ $\forall x, y, z$ (Between$(x, y, z) \vee$ Between$(y, z, x)) \vee$ Between$(z, x, y)$

$F_{10}$ $\forall x, y, z, w$ $(\neg(y = z) \wedge$ Between$(x, y, z) \wedge$Between$(y, z, w)) \rightarrow$ Between$(x, z, w)$

$F_{11}$ $\forall x, y, z$ Between$(x, y, x) \rightarrow$ Between$(z, y, x)$

## 3.8 Additional Topics

### 3.8.1 First Order Logic Subsumes Propositional Calculus

When we defined the syntax of first-order logic, we made a point to retain the truth-functional connectives, and in defining the proof rules, we kept the rules we had defined for the propositional calculus. It is reasonable to ask, then, whether everything that we did in the propositional calculus can also be done in first-order logic. The motivation to move to first-order logic was to use a more expressive language that would capture quantification, but we never formally guaranteed that we did not lose anything along the way. The definition of consequence remained the same, in that $\phi$ is a consequence of a set $\Phi$ if and only if every way of satisfying $\Phi$ is also a way to satisfy $\phi$, but we changed what satisfaction means. In the propositional calculus, satisfaction was a relation between a truth assignments and formulae; in first-order logic, satisfaction is a rather more tricky three-way relation between interpretations, variable mappings, and formulae.

Despite the difference in semantics between the propositional calculus and first-order logic, the former can indeed be reduced to first-order logic. That is, we can map the formulae of the propositional calculus to formulae of first-order logic such that: (i) consequence and nonconsequence among the first-order formulae implies consequence and nonconsequence among the propositional formulae, and furthermore that (ii) given an interpretation satisfying the first-order formulae, we can construct a truth assignment satisfying the propositional formulae.

The intuition behind the mapping is that propositional symbols are mapped to nullary relation symbols (that is, zero-arity relation symbols), boolean connectives will be preserved, and that an interpretation's mapping of relation symbols to truth functions will be used to construct truth assignments for the propositional formulae. To formalize this intuition, we will define a mapping from the formulae of the propositional calculus to sentences of first-order logic, and then show that for each sentence in the propositional calculus, there is a truth assignment satisfying the formula if and only if there is an interpretation satisfying the first-order sentence to which the propositional sentence is mapped.

The mapping $\mathscr{M}$ is defined as follows. Each propositional symbol $\mathsf{P}_i$ is mapped to the nullary relation symbol $\mathsf{R}_i^0$; that is, $\mathscr{M}(\mathsf{P}_i) = \mathsf{R}_i^0$. Any compound propositional sentence is either a negation, conjunction, disjunction, conditional, or biconditional. $\mathscr{M}$ maps these recursively; that is, $\mathscr{M}(\neg\phi) = \neg(\mathscr{M}(\phi))$, $\mathscr{M}(\phi \wedge \psi) = \mathscr{M}(\phi) \wedge \mathscr{M}(\psi)$, and so on. Thus $\mathscr{M}$ maps every sentence of the propositional calculus to a sentence of first-order logic. In fact, the range of $\mathscr{M}$ is a subset of the sentences of first-order logic; $\mathscr{M}$'s range is exactly the sentences of first-order logic which contain only nullary relation symbols and boolean connectives.

Now we show that there is a truth assignment for the propositional calculus that satisfies a sentence $\phi$ of the propositional calculus if and only if there is an interpretation that satisfies the first-order sentence $\mathscr{M}(\phi)$. Suppose $\pi$ is a truth assignment. Then for each propositional symbol $\mathsf{P}_i$, the value of $\pi(\mathsf{P}_i)$ is either true or false. Any interpretation $\mathfrak{I} = \langle \mathscr{D}, \mathscr{I} \rangle$ in which $\mathscr{I}$ maps $\mathscr{M}(\mathsf{P}_i)$, that is $\mathsf{R}_i^0$, to the empty nullary relation if $\pi$ assigns true to $\mathsf{P}_i$ and to the total nullary relation

if $\pi$ assigns false to $\mathsf{P}_i$, will satisfy $\mathscr{M}(\phi)$ if and only if $\pi$ satisfies $\phi$. The proof is inductive over the grammar of the propositional calculus, and we present only the base case and one inductive case.

The base case of the induction is propositional sentences which are single propositional symbols. Consider a propositional sentence which is a single propositional symbol $\mathsf{P}_i$. The corresponding first-order sentence is $\mathscr{M}(\mathsf{P}_i) = \mathsf{R}_i^0$. If there is a truth assignment $\pi$ which satisfies $\mathsf{P}_i$, then $\pi$ must map $\mathsf{P}_i$ to true. Any interpretation that maps $\mathsf{R}_i^0$ to the total nullary function (and such interpretations do, in fact, exist), satisfies the sentence $\mathsf{R}_i^0$, which is precisely $\mathscr{M}(\mathsf{P}_i)$. In the other direction, suppose an interpretation satisfies the sentence $\mathsf{R}_i^0$, that is, $\mathscr{M}(\mathsf{P}_i)$. Such an interpretation maps $\mathsf{R}_i^0$ to the total nullary relation. Any truth assignment $\pi$ that assigns true to $\mathsf{P}_i$ (and such truth assignments do, in fact, exist), satisfies the sentence $\mathsf{P}_i$. Thus, for a propositional symbol $\mathsf{P}_i$, a truth assignment $\pi$ satisfying $\mathsf{P}_i$ exists if and only if an interpretation $\mathfrak{I}$ exists satisfying $\mathscr{M}$.

We consider one inductive case, that for conditionals. Consider a propositional sentence $\phi \Rightarrow \psi$. A truth assignment satisfying $\phi \Rightarrow \psi$ exists if and only if an interpretation exists satisfying $\mathscr{M}(\phi \Rightarrow \psi)$. Suppose there is a truth assignment $\pi$ satisfying $\phi \Rightarrow \psi$. By the semantics of the propositional calculus, either $\pi$ satisfies $\psi$ or $\pi$ does not satisfy $\phi$. By the inductive hypothesis, then, there an interpretation $\mathfrak{I}$ that either satisfies $\mathscr{M}(\psi)$ or that does not satisfy $\mathscr{M}(\phi)$. Then $\mathfrak{I}$ satisfies the first-order sentence $\mathscr{M}(\phi) \Rightarrow \mathscr{M}(\psi)$, which is exactly $\mathscr{M}(\phi \Rightarrow \psi)$. In the other direction, suppose that there is an interpretation $\mathfrak{I}$ that satisfies $\mathscr{M}(\phi \Rightarrow \psi)$, that is $\mathscr{M}(\phi) \Rightarrow \mathscr{M}(\psi)$. Then $\mathfrak{I}$ either satisfies $\mathscr{M}(\psi)$ or does not satisfy $\mathscr{M}(\phi)$, and by the inductive hypothesis there is in the first case a truth assignment that satisfies $\psi$ and so satisfies $\phi \Rightarrow \psi$, and in the second case a truth assignment that does not satisfy $\phi$ and so satisfies $\phi \Rightarrow \psi$; and so in either case, there is a truth assignment satisfying $\phi \Rightarrow \psi$. Thus, there is a truth assignment satisfying the propositional sentence $\phi \Rightarrow \psi$ if and only if there is an interpretation satisfying the first-order sentence $\mathscr{M}(\phi \Rightarrow \psi)$.

A full proof would include an inductive case for each boolean connective, and concludes that $\mathscr{M}$ is such that for every sentence $\phi$ of the propositional calculus, there is a truth assignment satisfying $\phi$ if and only if there is an interpretation satisfying $\mathscr{M}(\phi)$.

Since we can reduce questions about consequence to questions of satisfiability ($\Phi \models \phi$ if and only if there is no truth assignment satisfying $\Phi \cup \neg\phi$), we have shown that satisfiability and consequence in the propositional calculus are subsumed by satisfiability and consequence in first-order logic. A similar proof shows that for any set of sentences $\Phi$ and sentence $\phi$, if $\Phi \vdash \phi$ in the propositional calculus, $\mathscr{M}(\Phi) \vdash \mathscr{M}(\phi)$ in first-order logic. Thus, the propositional calculus is subsumed by first-order logic; anything that can be expressed using the propositional calculus can be expressed in first-order logic.

### 3.8.2 Russell's (Barber) Paradox

In June of 1902, logician Bertrand Russell penned a short letter to Gottlob Frege that changed the course of logic and mathematics. We haven't the time here to delve into

the details surrounding this letter, including the epistulary response it prompted from Frege; we shall content ourselves with an understanding of the letter's main highlight: a short proof that in one shot turned a large part of Frege's attempt to erect a firm foundation for all of mathematics to turmoil.[10] In this short section, we first point toward a colorful form of this proof that we expect you to find, and then, after presenting the axiom that was a casuality of Russell's letter, sketch a version of the proof closer to Russell's original version. Here again we expect you to find the formal proof, and to encode and confirm it in Slate.

So, for the colorful form, we assume that there's a certain town in which lives a certain (male) barber who shaves all and only the men in that town who don't shave themselves. How would you represent this assumption in FOL? We suggest that you put aside your reading, and attempt now to construct a first-order formula that does the trick. ...

Finished? Okay, now compare your proposal to ($\mathscr{B}$):

$$\exists x\, z[\mathit{BarberOf}(x,z) \wedge \mathit{Town}(z) \wedge \mathit{ManOf}(x,z) \wedge \forall y(\mathit{ManOf}(y,z) \rightarrow \mathit{Shaves}(x,y) \longleftrightarrow \neg \mathit{Shaves}(y,y))]$$

You may of course have something different, but your proposal is correct if and only it's logically equivalent to ours. Formally establishing such logical equivalence could get a bit tedious, because you would need to emend your formula to make sure that either it uses exactly the same relation symbols as ours, or your proofs (that ours can be derived from yours in Slate in FOL, and *vice versa* as well) make use of "bridge laws."[11] For a timesaver, you can run a different check on your work: you can show that from your formula (encoded in Slate) you can derive an outright contradiction $\phi \wedge \neg\phi$, and you can show that the same holds for our $\mathscr{B}$. So that you don't have to key in $\mathscr{B}$, we have provided a pre-prepared file for you to use: `barber_exploration1.slt`.

We said that a particular axiom was a casualty of Russell's letter. What was it? Some notation first: Where $\phi$ is some arbitary first-order formula, we write $\phi(\nu)$ to express that the only free variable in $\phi$ is $\nu$. Now, here's the Fregean axiom targeted by Russell:

**Axiom V** $\exists y \forall x[x \in y \longleftrightarrow \phi(x)]$

[10] For Russell's letter, in English (it was originally written in Frege's native German), along with some elegant and informative commentary and context, see (Russell 1967). For the superhumanly gracious reply from Frege, see (Frege 1967).

[11] What are they? In the present situation, a bridge law would link the relations you used to ours. For example, we use *MathOf* in $\mathscr{B}$, but perhaps you used *ManIn*. A bridge law in this case would be

$$\lambda \quad \forall x, y(\mathit{ManOf}(x,y) \longleftrightarrow \mathit{ManIn}(x,y)).$$

If then instead of our $\mathscr{B}$ you went with ($\mathscr{B}'$)

$$\exists x\, z[\mathit{BarberOf}(x,z) \wedge \mathit{Town}(z) \wedge \mathit{ManIn}(x,z) \wedge \forall y(\mathit{ManIn}(y,z) \rightarrow \mathit{Shaves}(x,y) \longleftrightarrow \neg \mathit{Shaves}(y,y))],$$

then while these two formulas aren't inter-derivable, we do have:

$$\{\mathscr{B}, \lambda\} \vdash \mathscr{B}' \text{ and } \{\mathscr{B}', \lambda\} \vdash \mathscr{B}.$$

### 3.8.3 Guarded Quantification

When we translated English sentences like "All men are mortal" or the equivalent "Every man is mortal" we began by expanding into a more precise (though also more verbose) form, namely, "for every $x$, if $x$ is a man then $x$ is a mortal." When we spoke of every individual within some limited portion of our domain (e.g., "all men") we ended up rendering our sentence as a universally quantified conditional, where the antecedent of the conditional guaranteed that our variable in fact referred to some element in the limited portion of the domain. However, we would make the existential claim "Some men are courageous" more precise by rendering it as "There is an individual $x$ such that $x$ is a man and $x$ is courageous." This latter sentence is an existentially quantified conjunction, in which one of the conjuncts ("$x$ is a man") guarantees that $x$ in fact refers to some element in the limited portion of the domain.

This pattern is so useful that it is abbreviated in many contexts. For instance, "There is a number less than or equal to 5 which is not equal to 3" would be formalized as $\exists x\ (x \leq 5 \wedge x \neq 3)$. This can be abbreviated as $\exists x \leq 5\ x \neq 3$. "Every number greater than or equal to 10 is the sum of 7 and some number" is formalized $\forall x\ \big((x \geq 10) \Rightarrow \exists y\ \big(x = +(7, y)\big)\big)$. This can be similarly abbreviated as

$$\forall x \geq 10\ \exists y\ \big(y = +(7, y)\big).$$

These types of abbreviations make use of what is often called **guarded quantification**. (A synonym used by many to refer to the same technique is **bounded quantification**.) In a mathematical domain, common bounds for quantifiers will be inequalities (e.g., as in the examples just presented) or properties of mathematical objects (e.g., evenness, oddness, primality). Of course, any expression that the quantified variable could be "plugged into" would work. For instance, "Some men are courageous" can be formalized with as $\exists x \mathsf{Man}\ \mathsf{Courageous}(x)$, and "All men are mortal" as $\forall x \mathsf{Man}\ \mathsf{Mortal}(x)$. These sentences could be expanded as $\exists x\ (\mathsf{Man}(x) \wedge \mathsf{Courageous}(x))$ and $\forall x\ (\mathsf{Man}(x) \Rightarrow \mathsf{Courageous}(x))$, respectively.

### 3.8.4 Enforcing Domain Size

As we've seen, whether an interpretation $\mathfrak{I} = \langle \mathscr{D}, \mathscr{I} \rangle$ satisfies a sentence $\phi$ depends on the interpretation's mapping of function and relation symbols to functions and relations on $\mathscr{D}$; and these symbols vary based on the circumstances at hand. For instance, if we desire to reason in FOL about the student body at a particular college or university, it's a good bet that we would want to make use of the unary relation symbol `Student`, so that if for instance Cindy and Sally are students, this could be represented by the conjunction of `Student(cindy)` and `Student(sally)`. It shouldn't escape our notice, however, that the the equality relation $=$ must *always* be available when the relation symbols and function symbols are tailor-made for a particular application. Not only that, but we should take note of the fact that $=$ isn't really anything exotic: it simply denotes the standard, well-known, and used-since-kindergarten equality relation. You've long known for example that $3 + 4 = 7$, and the role of $=$ in this simple equation carries directly over to its application to


Figure 3.13: Two Interpretations Satisfying the Sentence Likes(a, b).

the elements of $\mathscr{D}$. A simple FOL formula such as `cindy` = `sally`, for example, will be true just in case that which `cindy` denotes in $\mathscr{D}$ is identical to that which `sally` denotes in $\mathscr{D}$. For another example, one that relates to the literary sphere, since the famous spy novelist John LeCarré is the pen name of David Cornwall, the following (with suitable choices for constants) holds:

$$\texttt{dcornwall} = \texttt{jlecarre}.$$

Now, passing to the point of the present section, it turns out that the rather unassuming relation symbol = allows us to dictate the *size* of a given domain $\mathscr{D}$. This is so, in general, because a bit of ingenuity will allow us to build formulae which revolve around the use of =; and these formulae will in turn determine how many objects must be in the domain of an interpretation satisfies them. The basic trick, informally put, is just this: If a formula $\phi$ says "There are exactly three things," then any situation in which $\phi$ is true have exactly three things.

We now consider some interesting properties of domain sizes of interpretations satisfying various sentences. The semantics of first-order logic guarantees that the domain of every interpretation is non-empty, and thus every domain has at least one element. We have seen that it is possible for sentences to be satisfied by interpretations with a single element, even if the most likely reading (to a human reasoner) might involve domains of greater sizes. For instance, the sentence Likes(a, b) probably evokes an image of two distinct individuals such that the Likes relationship holds between one individual, denoted by a, and the other, denoted by b. But this needn't be the case: a and b could denote the very same entity, and that entity could be a self-liker. This shows that the truth of Likes(a, b) doesn't guarantee that there are at least two things.

Let's now plumb the potential of = a bit more thoroughly. How might we guarantee that an interpretation satisfying a sentence must have at least two elements in its domain? Formulae involving equality are useful in this endeavor. For instance, to satisfy the sentence $\mathsf{a} \neq \mathsf{b}$, an interpretation must map a and b to different elements of the domain, and to be able to do that, there obviously must be at least two elements in the domain. To be a bit more general, we can achieve similar results without using any particular constant symbols: $\exists x\, \exists y\, (x \neq y)$. To ensure that there are at least three elements in a domain, we add a third existentially quantified variable and some more inequalities:

$$\exists x\, \exists y\, \exists z\, (x \neq y \wedge x \neq z \wedge y \neq z).$$

In general, a sentence of the form

$$\exists x_1 \ldots \exists x_n\, (x_1 \neq x_2 \wedge x_1 \neq x_2 \wedge \ldots \wedge x_{n-1} \neq x_n)$$

will only be satisfied by interpretations whose domains have at least $n$ elements. For each $n$, let $\underline{\phi_n}$ be the sentence all of whose interpretations must have domains of size at least $n$. Hence, we can easily enough construct sentences for which satisfying interpretations must have domains whose sizes are at least some minimum value. But can we also construct sentences such that any satisfying interpretation's domain will have at *most* some particular number of elements? Certainly.

To see how, let's begin with the smallest domain size possible, 1. $\exists x\, \forall y\, (x = y)$ is readily seen to be satisfied only by interpretations whose domains are singleton sets. The domain of any interpretation satisfying $\exists x\, \exists y\, \forall z\, (z = x \vee z = y)$ can have at most two elements. The two existentially quantified variables can denote at most two distinct elements of the domain, and if there were another element in the domain, then it would not be the case that $z$ could range over this third element, for it would not be the case that either $z = x$ or $z = y$. Thus the domain of any satisfying interpretation has at most two elements. We can continue this pattern as well to construct for any $n$ a sentence which can be satisfied only by interpretations whose domains have at most $n$ elements. For each $n$, let $\overline{\phi_n}$ be the sentence all of whose interpretations must have domains of size at most $n$.

With these two patterns in hand, we can easily construct sentences such that any satisfying interpretations must have domains of exactly some particular size. For instance, the sentence $\underline{\phi_5} \wedge \overline{\phi_5}$ is only satisfied by interpretations whose domains have exactly five elements. Using these forms, for any $n$, we can construct sentences all of whose satisfying interpretations' domains have at most $n$ elements, sentences whose satisfying interpretations' domains have at least $n$ elements, and sentences whose satisfying interpretations' domains have exactly $n$ elements. For the latter case, we can avail ourselves of a shorthand, by stipulating that $\phi^{=n}$ is a formula that asserts that there exist exactly $n$ things.

#### 3.8.4.1 Can FOL Capture Infinitude and Finitude?

Does the machinery introduced in the previous section enable us to show that the concepts of finitude and infinitude can be captured by suitable use of first-order logic? If so, how? We should first immediately sharpen this question, which as it stands is somewhat unclear. Let's first target the capturing of infinitude in FOL. With this aim in mind, our initial sharpening move is to stipulate that we are interested specifically in figuring out how we might use FOL to express that a set is countably infinite.[12] (Recall that we defined what it is for a set to be countably infinite in §1.5.3.) In further sharpening of the intuitively expressed question that kicked off the present section, what we shall be specifically hunting for is how to specify a set $\Phi$ that is such that a given interpretation

$$\text{(inf)} \quad \mathfrak{I} \models \Phi \text{ iff domain } \mathscr{D} \text{ in } \mathfrak{I} \text{ is countably infinite}$$

where the set $\Phi$ contains only formulae in FOL. If we can somehow obtain such a set $\Phi$, then we will have found a way to capture countable infinitude.

[12] There is no reason in principle why we shouldn't investigate how FOL might be used to capture *un*countable infinitude, and indeed that time will arrive later in the present volume. But for now we shall be prudently humble in our ambitions.

So can we meet this challenge, by drawing upon what was done in the previous section? Indeed we can. Consider the following set of first-order formulae:

$$\text{(def)} \quad \Phi := \{\underline{\phi_k} : k \in \mathbb{Z}^+\}$$

This set does the trick! We can easily prove that it does:

> **Proof**: Our target is (inf). For left-to-right, suppose that some $\mathfrak{I}^*$ satifies $\Phi$, and that $\mathscr{D}^*$, this interpretations domain, is only finite. $\mathscr{D}^*$ must be a particular size; assume without loss of generality that it's size (= cardinality) is $m \in \mathbb{Z}^+$. But by the definition (def) of $\Phi$ we infer there is some $\underline{\phi_{m+1}} \in \Phi$ that is true on $\mathfrak{I}^*$. But that in turn implies that $\mathscr{D}^*$ has at least $m+1$ members — contradiction! The consequent is thus reached. (We leave the converse to you.) **QED**

Now, what about finitude? Can it be captured by a set of formulae in FOL? The question here can be taken to consist in the challenge to find a set $\Psi$ of first-order formulae of such that a given interpretation

$$\text{(fin)} \quad \mathfrak{I} \models \Psi \text{ iff domain } \mathscr{D} \text{ in } \mathfrak{I} \text{ is finite}$$

where, again, the set in question once again contains only formulae in FOL. Can you find such a set? Alas, you can't: the task is impossible. To begin to learn why, we strongly suggest that you nonetheless try for a while, in earnest. It turns out, somewhat ironically, that an *infinitary* version of first-order logic, $\mathscr{L}_{\omega_1\omega}$, can, with great ease, capture finitude, using a single formula. For how this works, see section 3.8.8.

### 3.8.5 The Logic of Country Music

In the 1979 hit, "Coward of the County," Kenny Rogers asserts in song that "there's someone for everyone and Tommy's love was Becky." Assuming that "Tommy's love was Becky" is formalized as $\mathsf{Loves}(\mathsf{Tommy}, \mathsf{Becky})$, how could "there's someone for everyone" be formalized? What do you think is the intended interpretation? What differences are there between different formalizations?

*A formalization that puts the universal quantification within the scope of the existential, e.g.,*

$$\exists x \forall y \ \mathsf{Loves}(x, y)$$

*, interprets "there's someone for everyone" to mean "there is someone who* is *for everyone," that is, that there is someone who is loved by everyone. Alternatively, giving the universal quantifier wider scope, e.g. as in*

$$\forall x \exists y \ \mathsf{Loves}(x, y)$$

*, gives "there's someone for everyone" the meaning "there is someone specific to each person," that is, that for each person, there is some person to be loved by the first. The songwriter's intent is almost certainly the second of these.*

*An alternate formalization using functions might be*

$$\forall x \exists y \ \ y = \mathsf{Lovee}(x)$$

*, where* Lovee$(x)$ *denotes the "someone" for* $x$*. Under this formalization, Kenny Rogers' claim is actually a theorem (provided that* Lovee *is a function symbol in our language); a proof is given in Fig. 3.14.*

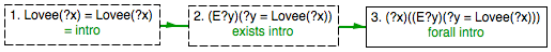



Figure 3.14: A proof that "there's someone for everyone."

### 3.8.6 Concerning Existence

The semantics of first-order logic require an interpretation to map *every* constant symbol to some element of the domain. This fact, along with the inference rules that we have described, leads to some interesting and, perhaps, counter-intuitive results. For instance, we can easily prove the existence of Santa Claus: By equality introduction we obtain SantaClaus = SantaClaus. From this equation, by existential introduction, we derive $\exists x\ x = \mathsf{SantaClaus}$. What is going on here? Any interpretation must map the constant symbol SantaClaus to some element of its domain, and so it is, in fact, the case, there is always some element in the domain which is equal to the element of the domain to which the interpretation maps SantaClaus.

This sort of proof highlights one of the differences between natural language and the formal language of first-order logic: In English, a noun phrase might not denote a real thing, while every term in first-order logic does denote some element of the domain of some interpretation. In English we may use the phrase "the greatest prime number," even though there is no such number, for instance in this sentence: "The greatest prime number has twenty-three digits." Such a sentence is bizarre, for its subject does not actually exist. (We do not have such a great problem with other subjects that do not exist. For instance, we have no problem with "Santa Claus lives at the North Pole.") However, such a sentence in first-order logic will necessarily include a claim of the existence of such a number, and will therefore be contradictory. For instance, we might formalize the claim as $\exists x\ \big[\mathsf{Prime}(x) \wedge \forall y\ \big[\mathsf{Prime}(y) \Rightarrow y \leq x\big] \wedge \mathsf{Has23Digits}(x)\big]$. Such a sentence, however, implies the sentence $\exists x\ \big[\mathsf{Prime}(x) \wedge \forall y\ \big[\mathsf{Prime}(y) \Rightarrow y \leq x\big]\big]$, which is contradictory.

In sum, every term in first-order logic must be mapped by an interpretation to some element of the interpretation's domain, and this can occasionally lead to surprising results, for we are accustomed to and comfortable with using phrases whose referents do not actually exist.

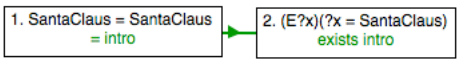



Figure 3.15: Proof that Santa Claus exists.

### 3.8.7 Formalizing Some English Sentences

Imagine classroom in which there are four students sitting in a row. Consider the sentence, "For all students $x$ and $y$, either $x$ is to the left of $y$ or $x$ is to the right of $y$." Is the following a good formalization of this English sentence? (The phrases "left of" and "right of" are to be interpreted generally, and not as "immediately to the left of" and "immediately to the right of.")

$$\forall x\ \forall y\ \big(\mathsf{Student}(x)\wedge\mathsf{Student}(y)\big)\Rightarrow\big(\mathsf{LeftOf}(x,y)\vee\mathsf{RightOf}(x,y)\big) \tag{3.21}$$

How would we formalize "Every male student is to the left of every female student"? This has the now-familiar general form "all $A$s are $B$s," where $A$ stands for *being a male student* and $B$ for the property *being to the left of every female*. We may naturally formalize that $x$ is an $A$ by $\mathsf{Male}(x)\wedge\mathsf{Student}(x)$. But how shall we treat the property picked out here by $B$? We can reword the property and say that such an $x$ has the property provided that "every female student is to the right of $x$;" and we can formalize this using the predicates Student, Female, and LeftOf, so long as we take care that the terms related by LeftOf are in the proper order. Do you agree that the following constitutes a good formalization, then?

$$\forall x\ \Big[(\mathsf{Male}(x)\wedge\mathsf{Student}(x))\Rightarrow\big(\forall y\ \big(\mathsf{Female}(y)\wedge\mathsf{Student}(y)\big)\Rightarrow\big(\mathsf{LeftOf}(x,y)\big)\big)\Big] \tag{3.22}$$

How about "Every bearded student is in back of a clean-shaven student"? Here again the underlying form is "all $A$s are $B$s," and so we can start with this template as a guide. Can you finish the process and supply a formula in FOL that captures the English sentence in question?

And consider this challenge: "Some male student is in front of every female student." Do you agree that the following formula does the trick?

$$\exists x\ \Big[(\mathsf{Student}(x)\wedge\mathsf{Male}(x))\wedge\forall y\ \Big[\big(\mathsf{Student}(y)\wedge\mathsf{Female}(y)\big)\Rightarrow\mathsf{FrontOf}(x,y)\Big]\Big] \tag{3.23}$$

### 3.8.8 Infinitary First-order Logic: $\mathscr{L}_{\omega_1\omega}$

First-order logic permits arbitrarily long formulae, but each formula must be only finitely long. $\mathscr{L}_{\omega_1\omega}$ is based on a relaxation of this constraint, and a rather radical relaxation at that: $\mathscr{L}_{\omega_1\omega}$ permits disjunctions and conjunctions to be countably infinitely long. For example, we know that the number zero is less than every positive integer. Where $s$ is the successor function on $\mathbb{N}$ that, when given some natural number $n$, returns $n+1$, we thus know that $0<s(0)$, that $0<s(s(0))$, that $0<s(s(s(0)))$, *ad infinitum*. What we know here can be expressed as a perfectly legal and conceptually simple formula in $\mathscr{L}_{\omega_1\omega}$, to wit:

$$(\text{f})\quad 0<s(0)\wedge 0<s(s(0))\wedge 0<s(s(s(0)))\wedge 0<s(s(s(s(0))))\ldots$$

This is a conjunction that never ends; the formula in question is infinitely long. Such a thing violates the grammar of $\mathscr{L}_1$, but is perfectly acceptable in the grammar for $\mathscr{L}_{\omega_1\omega}$. Of course, this formula is a bit cumbersome. Formulae of this form in

$\mathscr{L}_{\omega_1\omega}$ are written more elegantly, using "big" disjunction ($\bigvee$) and "big" conjunction ($\bigwedge$). Our sample formula (ƒ) from above, for instance, becomes this rather neater formula:

$$\bigwedge \{0 < \overbrace{s(\cdots 0 \cdots)}^{s \text{ } k \text{ times}} : k \in \mathbb{Z}^+\}$$

A final remark, one about some of the expressive power of $\mathscr{L}_{\omega_1\omega}$. We said in section 3.8.4.1 that first-order logic can't capture finitude, but that $\mathscr{L}_{\omega_1\omega}$ effortlessly can, via a single formula no less! Here is the formula in question:

$$(\infty) \quad \bigvee \{\underline{\phi_k} \wedge \overline{\phi_k} : k \in \mathbb{Z}^+\}$$

You should take the time, now, to make sure you see why any interpretation that satisfies ($\infty$) must have a finite domain, and *vice versa*.

# Chapter 4

# Higher-Order Logic

## 4.1 Introduction and Overview

As an alert and genuinely engaged reader, you will doubtless have wondered about the use of 'first' in 'first-order logic' (and correspondingly about 'F' in 'FOL'). For common-sense tells us that since there's first-order logic (FOL), presumably there is at least *second*-order logic (SOL), and there may well be a *third*-order case (TOL), and perhaps even something beyond that. The present chapter introduces you not only to SOL and TOL, but in general to a form of **higher-order logic** which encompasses all finite-order logics in one fell swoop. The first step in doing so is to provide an informal, intuitive description of the progression by which we systematically move from FOL to SOL to TOL. Let's take this step now.

## 4.2 Intuitive Encapsulation: ZOL-to-TOL Progression

Perhaps the best way to approach the progresion to which we have alluded is to treat English declarative sentences as input to an algorithm $\mathscr{R}$ that outputs the formal symbolization of the input sentences, in the form of representations of those sentences, where these representations are expressed as formulae in the formal language of some logic within the progression.[1] Our progression will actually start with a step-back from FOL to what we have called before the **pure predicate calculus**, which we now refer to as **zero-order logic** (ZOL), and will then move to first-order logic (FOL), then to second-order logic (SOL), and then to third-order (TOL); each of these is more expressive than its predecessors[2] We can then view $\mathscr{R}$

[1] The idea here is that the algorithm $\mathscr{R}$ can be applied to any declarative sentence, irrespective of the role that that sentence might have. Some declarative sentences that appear in the practice of mathematics as carried out in papers, articles, books etc. are largely ignored; others receive a lot of attention. Yet $\mathscr{R}$ can be applied to 1+1=2 just the same as to some great unsolved conjecture, such as that **P≠NP**. For an overview of this conjecture/problem, visit https://en.wikipedia.org/wiki/P_versus_NP_problem.

[2] The continuum is e.g. presented cogently in (Andrews 2002). Indeed, this is the book we recommend for ambitious autodidacts who finish the present book, and wishy to learn more about higher-order logic.

as computing a function, and accordingly write $\mathscr{R} : \sigma \mapsto \phi_{\sigma\mathscr{L}}$, where $\sigma$ is an English declarative sentence, and $\phi_{\sigma\mathscr{L}}$ is a formula in the particular logic $\mathscr{L}$ that's in play. Even for readers unfamiliar with the continuum we have alluded to, it's easy to get the hang of it in general, and easy to get the hang of how it works in connection with $\mathscr{R}$. Here's how.

Consider first the following sentence, which is true in the screenplay *Double-Minded Man*, the main characters of which are Harriet and Joseph.[3]

(A) Harriet loves Joseph.

As you will agree given your mastery of preceding material, We can easily represent this sentence in ZOL.[4] You'll recall specifically that ZOL allows us to use **relation symbols** to represent relations (or properties), **constants** to represent individual objects, and to then build formulae that ascribe these relation symbols to these constants. This logic also includes the familiar quintet of **truth-functional connectives**: $\wedge$ ('and'), $\vee$ ('or'), $\rightarrow$ ('if _ then _'), and $\leftrightarrow$ ('_ if and only if _'). No quantification is available in ZOL, and there are no variables (but identity, $=$, is included — though we don't use it in the preent section; with $=$ comes two familar inference schemas for reasoning with identity[5]). For handling (A), we employ a relation symbol to represent the two-place property of one thing loving another, and we need two constants, one to denote Harriet and one to denote Joseph. With these ingredients and some straightforward punctuation, we can specify $\mathscr{R}[(A)]$:

(A′) $\mathsf{L}(\mathsf{h},\mathsf{j})$

Now here's an example of a more expressive English sentence that is also true with respect to the story *Double-Minded Man*:

(B) Joseph owns a BMW sportbike.

To represent (B), we can use the following formula in FOL:[6]

(B′) $\exists x(\mathsf{B}(x) \wedge \mathsf{S}(x) \wedge \mathsf{O}(j, x))$

The symbol '$\exists$' is the now-well-understood **existential quantifier**, and when paired with variable $x$, as in '$\exists x$,' is of course read as 'there exists an $x$ such that.' As to $\mathsf{B}$, $\mathsf{S}$, and $\mathsf{O}$ in (A′), they of course are relation symbols that represent, resp., the properies *being BMW-manufactured*, *being a sportbike*, and *being a thing _ that owns a thing _*. FOL, as you'll doubtless recall, also incudes the **universal quantifier**, $\forall$, which allows us to capture such sentences as

(C) Every spouse of Joseph owns more than one Bible.

[3] The gist of *D-MM* is given in the appendix at the conclusion of the present chapter.

[4] ZOL, remember, subsumes the propositional calculus.

[5] I.e. `identity introduction` and `identity elimination`. See §3.5.3.

[6] Perhaps we'd better admit at this point that cognoscenti in pursuit of building all of $\mathscr{R}$ would call for pairing FOL with e.g. the additional machinery of the $\lambda$-calculus, as e.g. in (Blackburn & Bos 2005). But we can leave this aside. Readers interested in the formal tools in linguistics needed for intelligent pursuit of $\mathscr{R}$ should begin their study with (Partee, Meulen & Wall 1990).

which is also true of Harriet in *Double-Minded Man*,[7] and would by $\mathscr{R}$ yield

(C′) $\forall x[\mathsf{Sp}(x, j) \rightarrow (\exists x \exists y \exists z(\mathsf{Bi}(x) \wedge \mathsf{Bi}(y) \wedge \mathsf{O}(z, x) \wedge \mathsf{O}(z, y)))]$

To be clear about the aforementioned continuum: We invoked ZOL by regimenting the ascription of properties to particular individual things picked out by constants, then moved from ZOL to FOL by allowing quantification over lower-case variables $x, y, \ldots$ that are placeholders for individual objects. To next move from FOL to SOL, we simply extend quantification so that it ranges over not only individual objects, but over *relation symbols as well*. To do this, we simply introduce a new collection of majuscule variables — $X, Y, Z, \ldots$ — that range over not individual objects, but over relation symbols (and hence ultimately over properties). To quickly get a sense of what this specifically buys us, consider an English sentence that communicates another truth in *Double-Minded Man*:

(D) There are attributes Joseph has that Harriet's mental model of him doesn't.

Let's denote Harriet's mental model of Joseph with the constant symbol $\mathsf{m}$. Then we can represent (D) in SOL as follows:

(D′) $\exists X(X(\mathsf{j}) \wedge \neg X(\mathsf{m}))$

A move, at this point, to TOL would consist in a direct analogue of what got us started in the first place when we built ZOL: viz., we allow ascription of a new class of relation symbols to our original set of relation symbols. So far, relation symbols have been pulled from the set of upper-case Roman letters, and applied to individual objects only. For our new class of relation symbols, which allow us to ascribe properties to properties, we can avail ourselves for instance of $\mathscr{F}, \mathscr{R}, \mathscr{S}, \ldots$. We return to TOL, and indeed fourth-order logic and beyond, a bit later in the current chapter.

## 4.3 Second-Order Logic, More Precisely Put

In Chapter 3 we set out the grammar of FOL rigorously. How, precisely, do we explain that new wffs (well-formed formulae) are now allowed in SOL? That's easy, and has been directly suggested by the examples given in the previous section: First, we simply augment the set of symbols used to construct wffs. The augmentation consists first in introducing new upper-case variables $X, Y, Z$, and allowing them to range over relations (in FOL, as you'll recall, our variables, all lower-case, ranged over only over particular objects). Our second move is to then include two new ways of building wffs of three new kinds:

R1 If $X$ is an $n$-ary relation variable and $t_1, t_2, \ldots, t_n$ are terms, then $X(t_1, t_2, \ldots, t_n)$ is a wff (of SOL).

[7] This fact isn't *revealed* in *D-MM*, but is something which, in keeping with the authorial craft of creating robust characters, is in the author's mind when writing. For more on such techniques and Ibsen, see (Bringsjord 1995).

R2 If $\phi$ is a wff (of SOL), then $\exists X\phi$ and $\forall X\phi$ are wffs.

Notice that the three new kinds of formulae,

$$X(t_1, t_2, \ldots, t_n) \quad \exists X\phi \quad \forall X\phi$$

are each distinguished by use of our new collection of available variables, which range not over individuals, but instead over relation symbols.
Using this pair of new rules and the three new kinds of formulae obtainable therefrom, we can see how easy it is to mechanically construct the formula (D′) exhibited above in connection with *Double-Minded Man*. We need but three steps. Step 1 is to take the relation symbol $X$ (which we assume to be unary), along with the constants $\mathsf{j}$ and $\mathsf{m}$, and build the two following wffs by use of R1:

$$X(\mathsf{j}) \qquad X(\mathsf{m})$$

In Step 2, we employ a formation rule from FOL, which allows us to take these two formulae and conjoin them, yielding:

$$X(\mathsf{j}) \wedge \neg X(\mathsf{m})$$

In our third and final step, we use R2 to produce our finished product:

(D′) $\exists X(X(\mathsf{j}) \wedge \neg X(\mathsf{m}))$

In the next set of exercises, you will have opportunity to try your hand at simulating $\mathscr{R}$ on some additional English sentences that naturally call for SOL in order to be formalized.

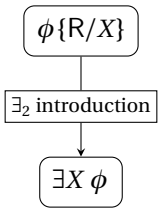
### 4.3.1 Exercises

1. Symbolize the following English sentences in SOL:
   (a) "Joseph and Harriet and have a property in common."
   (b) "Two things $a$ and $b$ having exactly exactly the same properties are identical."
   (c) Now let's turn to an example in elementary number theory. Suppose that the objects we're talking about are natural numbers. And suppose we wish to symbolize this statement: "If zero has a certain property, and if whenever $n$ has this property so does the successor of $n$, then all numbers have this property." Give a correct symbolization of this statement as a formula in SOL.
2. How exactly would Table 3.1 be adjusted to accomodate SOL? Specify the new parts that would need to be added in order to produce a corresponding table for SOL.

### 4.3.2 Formal Proofs in SOL: How?

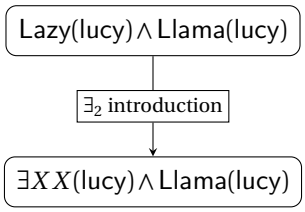
Now, what about formal proofs in SOL? How does that work? We're pretty such that you will be able to anticipate at least the general shape of the answer, which is simply that just as we have in our proof theory for FOL a pair of introduction rules, one for each of the two quantifiers (viz. `existential introduction` and `universal introduction`), so we need a new pair of quantifier-introduction rules

for SOL; and just as we have in our proof theory for FOL a pair of elimination rules, one for each of the two quantifiers (viz. `existential elimination` and `universal elimination`), so we need a new pair of quantifier-elimination rules for SOL. Let's start with our second-order version of `existential introduction`, which sanctions the deduction of a formula asserting the existence of some property (more precisely put: some relation symbol) from an assertion that a *particular* property is in play. For instance, given that "Lucy is a lazy llama" (which as a matter of fact is actually false as can be: Lucy is quite sedulous, esp. when it comes to things cyber), we are permitted to deduce that "there exists at least one property that Lucy has." Formally, from a formula $\phi\{\mathsf{R}/X\}$ we may infer $\exists X\ \phi$, and this is given schematically in (4.1).




(4.1)

To put this new inference schema to work, let's implement the deduction about Lucy that we just described in English. Here's what we get:




(4.2)

Notice that in (4.2) we decided to leave the relation symbol Llama untouched in the conjunction from which deduction proceeded, but we could've used the very same inference rule to produce

$$\exists X\, X\mathsf{(lucy)} \wedge X\mathsf{(lucy)}.$$

#### 4.3.2.1 Exercises

1. Specify a Slate-style inference schema that defines the second-order `universal`$_2$ `elimination`.
2. Specify a Slate-style inference schema that defines the second-order `existential`$_2$ `elimination`.
3. Specify a Slate-style inference schema that defines the second-order `universal`$_2$ `introduction`.
4. We could expand the grammar we've given in order to specify second-order wffs so that quantification over not only relation symbols is allowed, but in addition quantification over *function* symbols is as well. (How, exactly, would this be done?) However, since any use of (a) function(s) in any wff $\phi$ of SOL could be translated without any impact on meaning into a corresponding formula $\phi'$ having no function symbols, a cleaner presentation of SOL is obtained by leaving function symbols to the side, at least in the

minds of most writers. Yet, here's a formula $\sigma$ in SOL that makes use of the function symbol $f$, and includes quantification over it:

$$\forall f(\forall x \forall y[(f(x) = f(y) \rightarrow x = y) \rightarrow \forall x \exists y(x = f(y))]$$

What does this formula express? Put your answer in clear English. Provide a second-order formula that is equivalent to $\sigma$ but doesn't contain any function symbols.

## 4.4 Third-Order Logic, Fourth-Order Logic ...

We have referred to a progression/continuum of logics. So far we have described zero-order logic, first-order logic, and second-order logic, and have alluded to the next step in the progression: third-order logic. Do you perceive the pattern, and the principle by which it continues? This should help, in case the answer is negative: In the third-order case, we allow properties to have properties, and hence relation symbols to apply to relation symbols. This is the direct analogue to the move we made in creating zero-order logic, for to do that we allowed properties to be applied to individuals — but we didn't introduce any new quantificational machinery. (Zero-order logic, recall, in point of fact hasn't *any* quantifiers.) So these new relation symbols can't in third-order logic be quantified over, but we capture such English sentences as "Joseph and Harriet share a property that's praiseworthy" via for instance a construction such as this:

$$\exists X(X(\mathsf{j}) \wedge X(\mathsf{h}) \wedge \mathscr{P}(X)).$$

And what if we wished to quantify over properties of properties? Then we would be moving to fourth-order logic. For instance, consider the English assertion that "All the properties that Joseph and Harriet have in common each have a property." We can capture this with this formula:

$$\forall X((X(\mathsf{j}) \wedge X(\mathsf{h})) \rightarrow \exists \mathscr{X}(\mathscr{X}(X))).$$

## 4.5 Higher-Order Logic

We provide you a taste of higher-order logic by introducing **finite type theory**, a system of higher-order logic which is quite close to a system of higher-order logic given by Andrews (2002). We'll call our system $\mathbf{F}^\omega$. $\mathbf{F}^\omega$ subsumes ZOL, FOL, SOL, TOL, and indeed subsumes $n$-order logic, for *every* $n \in \mathbb{N}$. So, what are types? There's nothing wrong, at least initially, with conceiving of types as springing out of the brute, sobering fact that were we to continue to add to the progression of kinds of relation symbols and the variables that abstract them, we would quickly run out of earthly alphabets in which to expression the ever-growing progression, and even if we could valiantly keep up the fight for a good while, confusion would abound, since inevitably the differences between various $n$-order logics would be maddenly hard to wrap one's head around and keep straight. To refresh your understanding of the present situation, here are the step we've taken so far in our progression:

Step 1 individual variables and individual constants: $x, y, \ldots$ and $\mathsf{a}, \mathsf{b}, \ldots$

**Note** Truth be told, we have included function symbols, and allowed them to be applied to individual variables and constants. But we leave aside function symbols in our presentation of higher-order logic.

Step 2 propositional variables: $\mathsf{P}, \mathsf{Q}, \ldots$

Step 3 relation symbols in FOL to apply to the variables and constants invoked in Step 1: $\mathsf{Happy}, \mathsf{A}, \mathsf{B}, \ldots$

Step 4 variables to range over the relation symbols invoked in Step 3: $X, Y, \ldots$

Step 5 relation symbols in TOL to apply to the relation symbols and relation variables invoked in Steps 3 and 4: $\mathscr{R}, \mathscr{F}, \ldots$

Step 6 $\vdots$

Instead now of striving to cook up yet another alphabet on which to base Step 6, Step 7, and so on, we can be more systematic and invoke **type symbols** to indicate the types of variables, and deploy subscripts on these symbols. Here's the inductive definition of the types and orders of $F^{\omega}$:

T1 $\iota$ is a type symbol that denotes the type of *individuals*, and the order of $\iota$ is 0.

T2 $o$ is a type symbol that denotes the type of *propositional variables* and wffs in the ZOL, and the order of $o$ is 1.

T3 Let $\tau_1, \tau_2, \ldots, \tau_n$, where $i \in \mathbf{N}$, be type symbols. Then:

- i $\langle \tau_1, \tau_2, \ldots, \tau_n \rangle$ is a type symbol;
- ii the type symbol $\langle \tau_1, \tau_2, \ldots, \tau_n \rangle$ denotes the type of $n$-ary relation symbols whose sequence of terms $t_1, t_2, \ldots, t_n$ match up sequentially with $\langle \tau_1, \tau_2, \ldots, \tau_n \rangle$; and
- iii the order of type symbol $\langle \tau_1, \tau_2, \ldots, \tau_n \rangle$ is 1 plus the maximum of the orders of $\tau_1, \tau_2, \ldots, \tau_n$.

So for instance the type of the unary relation $\mathsf{Happy}$ in the formula $\mathsf{Llama}(\mathsf{larry})$ is $o$, and the order of this relation symbol is 1, since in this formula the type of $\mathsf{larry}$ is $\iota$ with an order of zero, and we increment by 1 (in accordance with rule T3.iii given just above).

But what are the wffs of $\mathbf{F}^{\omega}$, exactly? Given our considerable experience with the counterpart of this question for some of the simpler finite-order logics (e.g. FOL), the answer is easy to anticipate, and when given momentarily should be easy enough to grasp. Here then are the rules that define the wffs of $F^{\omega}$:

R1 Propositional variables and wffs from the pure predicate calculus are wffs. (Their order, recall, is 1.)

R2 If $v_{\tau_1}, v_{\tau_2}, \ldots, v_{\tau_n}$ are variables or constants of the types indicated by their subscripts, and $u_{\langle \tau_1 \tau_2 \ldots \tau_n \rangle}$ is a variable or constant too (with its type indicated by its subscript as well), then

$$u_{\langle \tau_1 \tau_2 \ldots \tau_n \rangle} v_{\tau_1} v_{\tau_2} \ldots v_{\tau_n}$$

is a wff.

R3 If $\phi$ is a wff, so is $\neg\phi$. (This is of course exactly in conformity with the same formation rule for negations we have e.g. seen in the case of the propositional calculus, ZOL, and FOL.)

R4 If $\phi$ and $\psi$ are wffs, so is $\phi \star \psi$, where $\star$ here represents any of the familiar four binary truth-functional connectives.

R5 If $\phi$ is a wff, and $v$ a variable of any type, then both $\exists v\phi$ and $\forall v\phi$ are wffs.

### 4.5.1 Exercises

1. Return to the SOL and TOL formulae we created in connection with Joseph and Harriet from *Double-Minded Man*, and determine their orders.

## Appendix

*Double-Minded Man* is a two-character, three-scene short-short film script written by S. Bringsjord and A. Bringsjord in and through *Movie Outline 3*, in a way that follows the approach to creative writing described in (Bringsjord 1992). The term 'Double-Minded' in the title refers to the mind of Joseph, by all accounts for years a rather boring and morally upright financial advisor working out of his own small suburban firm, but who is in reality a hard-drinking, cocaine-and-heroin-using, sportbike-driving, decidedly non-monogamous chap. Here's the current version of the first scene:

```
TWIRL - DAY

68-year-old Harriet Smith sits with two wrinkled hands firmly on the
wheel of her rust-eaten Subaru wagon, staring straight ahead through
the top level of bifocals as she waits serenely at a red light.

Harriet is alone in the car.  To her right is another vehicle, also
waiting, in this case to make a right turn; it's a sleek, low-slung,
black Camaro.

We are inside the cabin with Harriet.  The Subaru's sound system
softly plays choral music.  Harriet's lips move slightly as she
internally sings along, mouthing a slow aria.  Her head weaves
slightly side to side, in rhythm with the music.

Things are calm as can be here inside the car with Harriet.  There are
a pair of well-worn Bibles on the empty passenger seat beside her,
one with a gold-lettered 'Harriet' on its leather front cover, the
other with a matching 'Joseph' on its front cover.

Harriet's eyes swivel up to the light: still red.  We wait with her.

Suddenly there is a piercing SCREECH outside.  Harriet jerks her head
to the right and we follow her line of sight.
```

> A sleek motorcycle has swerved out of its lane and is now streaking straight for the right side of the Camaro beside Harriet's car.
>
> The bike slams with CLANG into the side of the Camaro. Its rider is flung up and forward into the air, twirling passed Harriet's windshield.
>
> We now watch from Harriet's POV, in slow motion. The black-leather-clad motorcyclist sails by Harriet's windshield, airborne. We see a man's face, clearly: His elephant-hide skin tells us that he is well beyond middle-age. Yet thick, black curls of youthful hair emerge from under his helmet. The rider has only one half of a black, bushy, swept-out, waxed mustache. His eyes are weary and grey, and appear to lock with Harriet's for an instant.
>
> We return to normal speed. The body is now lying on the incoming lane to the left of Harriet's Subaru, perfectly still on the blacktop, the head twisted into an impossible angle. Blood seeps from a nostril. Beside the lifeless head, a BMW medallion lies on the pavement, glinting in the sunlight.

Joseph, Harriet's 69-year-old husband, is thus dead on the spot. As we learn in the next scene, set in a morgue, Joseph's real hair consists of but anemic, white fuzz, and the second half of his fake mustache has been removed. The third and final scene is Harriet's visit to Joseph's secret lair, a small and dingy storage unit where he parked his sleek and super-fast motorcycle in secret. She there comes upon a number of things that reveal the dark aspects of Joseph's secret life, of which, hitherto, she had not the slightest inkling.[8]

[8] A recent draft of the entire screenplay is availale here.

# Chapter 5

# Propositional Modal Logic

## 5.1 Why Another Family of Logics?

We introduced first-order logic (FOL) when it became apparent that the propositional calculus was insufficiently expressive to represent certain kinds of perfectly meaningful declarative sentences — for example, even one as simple as "All whales are clever" — and to represent and certify certain kinds of obviously valid[1] deductive arguments, such as:

**Argument 1**

(1) All whales are clever.
(2) All clever things are fish.
∴ (3) All whales are fish.

To represent such sentences and arguments, we introduced the language of FOL, which as you'll recall includes quantification, variables, and names (= constants), along with a way of systematically interpreting the formulae these more demanding sentences and arguments give rise to; and we introduced and mastered — with help from Slate — a set of inference rules (= schemata) appropriate for reasoning deductively about sentences involving quantification. These innovations explain why at this point you are able to quickly encode the argument immediately above in an FOL workspace in Slate, and prove that (3) follows deductively from premises (1) and (2). You also know that the propositional calculus is too inexpressive to formalize Argument 1, since for example

$$\{W, C\} \nvdash F,$$

where the propositional constants symbolize the declarative statements in Argument 1. This trio is of course picked at random. We could instead use '$A$' for (1), '$B$'

[1] For those knowledgeable about cetacean zoology: Yes, whales aren't fish, but rather are mammals; but remember that deductive validity is entirely separate from truth and falsity. The argument here is — using vocabulary introduced earlier — *formally valid.* But the argument isn't *veracious* (since its premises aren't all true), and therefore it's not *sound.*

for (2), and '$C$' for (3) and the same negative result obtains; i.e.

$$\{A, B\} \nvdash C.$$

But the demands for additional expressivity in a logic don't stop with specimens like those immediately above involving whales, nor with other specimens provided earlier in the book; FOL is hardly the end even of our relatively short journey into beginning deductive logic. To see this, consider the following argument, which should stir memories of our motivation of FOL in Chapter 3, and which we give in the presumed context of Euclidean 2-space.[2]

**Argument 2**

(4) Necessarily, the interior angles of a triangle sum to 180 degrees.
∴ (5) The interior angles of a triangle sum to 180 degrees.

What is meant be 'Necessarily' here? The intuitive idea is straightforward, and you can probably anticipate it; the idea is that (4) expresses the claim that it absolutely, positively *must* be the case, mathematically or logically speaking, that any standard triangle's interior angles, when added together degree-wise, yield 180. And then the inference is made to (5), which asserts simply that the proposition to which 'Necessarily' is applied in (4) is in fact true.[3]

Clearly, Argument 2, like its predecessor, is deductively valid, given the concept of 'necessarily' used here. However, this argument can't be accurately represented in the propositional calculus; nor can it be satisfactorily captured in FOL. To see this, we have only to give it the old college try. In the case, first, of the propositional calculus, we obviously fail miserably, since (5), a declarative sentence, is symbolized by a propositional constant, say S, and likewise (4), as a *different* declarative sentence, would be represented by a different propositional constant, for example N. As we can all agree, there is no way to deductively derive N in the propositional calculus from S; this can of course be confirmed by a truth-value assignment in which **??** is assigned TRUE and **??** FALSE.[4] Alert readers will remember that when we motivated FOL by way of another argument involving triangles and their angles, we encountered

[2] As some readers will know, there are alternative geometries in which the three interior angles of a triangle don't add up to 180 degrees. You can verify this for yourself by carrying out a simple experiment: Draw a "triangle" on the surface of a sphere (a simple, sizable ball will do just fine), measure the three angles, and add the three numbers together.

[3] Once the reader has assimilated Chapter 6, "Theories," he or she will be in position to appreciate a more determinate definition of the 'Necessarily' used here, for we can say that, given the formal theory ($G$) of the geometry of Euclidean 2-space, some formula $\phi$ is necessarily the case provided that $\mathscr{G} \vdash \phi$.

[4] You can of course turn to a PC workspace in Slate, encode these atomic formulae in a pair of nodes, and then check whether the one for S follows by PC $\models$ from the one for N. This of course relies on the oracle for $\mathscr{L}_{\text{PC}}$. A path to take for those not wanting to rely on the oracle is to build a truth-tree in HyperSlate, which would be simply this stark trunk:


Obviously, this isn't a closed trunk/branch, so the truth-tree method confirms the deductive invalidity of Argument 2.

similar problems. Nonetheless, let's try to model Argument 2 in FOL, and see what happens.

To give FOL a shot, we can take advantage of work carried out on the argument involving triangles at the outset of Chapter 3. But we must confront an immediate problem, one that closely parallels the problem that arose when we tried to use the propositional calculus. To see it, suppose that (5) is some suitable formula in FOL; for example,

$$\forall x\ \mathsf{Triangle}(x) \Rightarrow \mathsf{IntSumTo180}(x).$$

If we label this formula with the meta-variable $\phi$, then we can try to represent (4) by invoking a new relation symbol in FOL to represent 'Necessarily'; let's use $\mathsf{N}$. We then have as a symbolization for the sentence (4):

$$\mathsf{N}(\phi).$$

But hold on! Nothing like this is possible in FOL, as can be verified by inspecting the grammar for FOL given in Chapter 3. The problem is that in FOL, relation symbols can only be applied to constants or variables; but $\phi$ is of course standing in for a formula!

The solution is to turn to brand-new machinery in a logic: the concept of an **operator**, which can be applied to entire formulae. Specifically, we now invoke two operators, one for 'Necessarily,' and one for 'Possibly.' The former will be expressed by the symbol $\Box$, and the latter by $\Diamond$. These are the traditional symbols used for the two "modes" of necessity and possibility.[5]

But before turning to the development of **propositional modal logic**, which is essentially produced by adding our two new operators to the propositional calculus, we make the point that attempting to squeeze FOL so as to enable it to handle natural-language sentences that make use of operators is truly unworkable.

Consider the argument (Argument 3) that from the statement that, necessarily, the Milky Way includes the planet Pluto, one can deduce that the Milky Way includes Pluto.[6] In FOL, about the best stab at capturing this would be something like this:

$$\begin{array}{ll} & 6'.\ \ \mathsf{Necessarily}(\mathsf{Includes}(\mathsf{milky\_way}, \mathsf{pluto})) \\ \therefore & 7'.\ \ \mathsf{Includes}(\mathsf{milky\_way}, \mathsf{pluto}) \end{array}$$

This is an interesting specimen. The fact is, the formula shown in $6'$ will be judged well-formed by Slate. But note that this is only because Includes is interpreted by Slate as a *function* symbol,[7] but clearly in Argument 3 the idea is that 'includes' is supposed to be a *property*, and hence must be represented in FOL as a relation symbol. If we then try to forge ahead with FOL by representing 'necessarily' as a relation symbol, and 'includes' as a function symbol, we see that Slate will accept

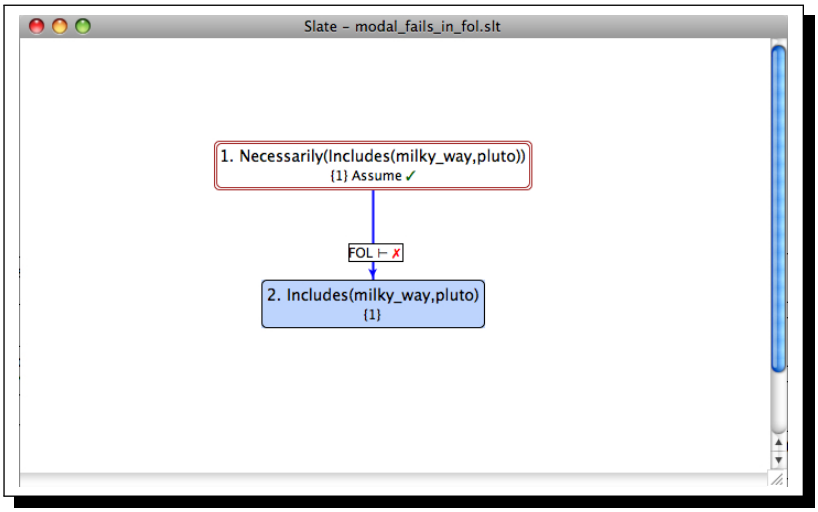
[5] Other symbols have, as you can doubtless guess, been used (e.g., 'L' for $\Box$ and 'M' for $\Diamond$), but nowadays the diamond and box, as they are often called, dominate usage.

[6] You may know that scientists have debated whether or not Pluto is a planet. We can leave this aside by simply stipulating for present purposes that Pluto is a planet, but an idiosyncratic one. We further assume that all idiosyncratic planets are planets.

[7] If you don't see exactly why this is the case, this would be a good spot to return to the grammar for the wffs of FOL given in Chapter 3. This grammar doesn't allow more than one relation symbol per atomic formula.

the formula as well-formed, but there is no way in FOL to deduce the formula in 7′ from the formula in 6′. This state-of-affairs is shown in Figure 5.1.

Figure 5.1: The Oracle for FOL Declares the Deduction Impossible in FOL




What happens if we try to apply Necessarily as a relation symbol to a propositional variable? We can attempt this in the present case by representing the declarative statement 'The Milky Way includes the planet Pluto' as simply P; this gives us the following argument as a target to capture in Slate.

$$\begin{array}{ll} & 6''. \quad \mathsf{Necessarily}(\mathsf{P}) \\ \therefore & 7''. \quad \mathsf{P} \end{array}$$

What happens if we try *this* in Slate, in a workspace for first-order logic? We leave it to you to carry out the attempt, and reflect upon what happens.

### 5.1.1 Epistemic Operators

There are other modal operators that we can (at least partially) capture with the propositional modal logics that we'll study in this chapter. For example, in addition to 'possibly' and 'necessarily,' we can capture 'knows.' To see this, and to convey the important point that operators can be "stacked," we now consider a simple parable.

Imagine that Rachel, a moneyed dowager, owns a fireproof combination safe in which she keeps some of her valuables. The safe is hidden behind a Degas in her dining room. Rachel knows the combination of the safe, and though in general she doesn't have this combination in mind, she can call it to mind whenever she moves the painting aside to either retrieve or store priceless items. Let's assume that the combination is composed of three numbers in the following sequence: 23-10-17. Now, specifically imagine that while Rachel is dressing upstairs in her bedroom

to attend a black-tie-and-gown ball she realizes that it would be a good idea to add some jewelry to the equation; specifically, Rachel wants to sport a diamond necklace for the evening. Given this, Rachel decides to head downstairs — but only after finishing her hair, which she calculates will require another 10 minutes. She makes a mental note to later retrieve the relevant jewelry from her safe, put it on down there, and walk outside to the waiting Rolls Royce.

At the moment of her decision to put on the necklace downstairs, Rachel doesn't bring to mind the combination of her safe: as we've already noted, she only brings the three numbers to mind once she is facing the dial, the painting moved aside. Nonetheless, at the moment of her decision to later descend and don her diamonds, she does clearly know that the combination of her safe is 23-10-17. For ease of reference, we will say that what Rachel knows is simply that $C$ is true, where $C$ symbolizes the proposition that the combination of her safe is 23-10-17.

But why is Rachel confident that she can jazz up her outfit once she gets downstairs, *even before she brings the trio of numbers to mind*? The answer is that at the time of her decision to add pizazz by donning her diamonds, she knows that she knows $C$. Now, how would we represent this fact in the logics we've studied so far? We simply can't. (You are encouraged to try!)

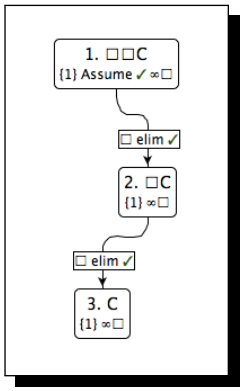
In order to handle the case of the sartorially splendid Rachel, we can simply use the box from above to represent not 'necessarily,' but 'knows.' Assuming then that we are talking only about one agent, Rachel, we can simply write the following into Slate to express that Rachel knows that Rachel knows $C$:

$$\Box\Box C.$$

In order to do this, you would open a new workspace in Slate, but instead of selecting a workspace for the propositional calculus or FOL, you would select one of the types of workspace that fall under the new family of logics introduced in the present chapter; for example, **S4**. To provide a peek ahead, we have done exactly this: We opened a new workspace in Slate, one of the **S4** variety, and we have proved that from the assumption that Rachel knows that Rachel knows that the combination is 23-10-17 (i.e., $\Box\Box C$), it follows that the combination is indeed 23-10-17. See Figure 5.2. This proof is very simple, but not insignificant; examine it now. Notice that we use the same rule of inference in each of the two steps: necessity elimination. This rule, in the current epistemic context, can be understood as allowing the Slate user to infer from someone's knowing $\phi$, that $\phi$ holds. Since everything that is known is true, the rule makes great sense.

The fact is, English (and of course other natural languages: Spanish, Hungarian, etc.), present the logician with the need to invoke *many* **modalities**. Modalities include, but are not limited to, notions of necessity and possibility (**alethic** modalities; these will be the foci in the present chaper), of knowledge and belief (**epistemic** and **doxastic** modalities), and of ethical obligation (**deontic** modalities), and time (**temporal** modalities). Most modal logics are designed to handle just one kind of modality, and then include the background machinery of the propositional calculus or first-order logic, but there are multi-operator modal logics that combine modalities. Multi-operator modal logics are beyond the scope of elementary deductive

Figure 5.2: Given that Rachel knows that she knows that the combo is 23-10-17, we can prove that that is in fact the combo.




logic, and you will not find an option in the version of Slate you have that allows multiple modalities in a single workspace. If you are lucky enough to have HyperSlate, a more advanced descendant of Slate, your instructor may have activated for your use of HyperSlate a multi-operator modal logic.

In the present version of this chapter, we briefly present: the most commonly studied propositional modal logics, those of necessity and possibility, the aforementioned alethic modalities; proof systems for these logics; and the distinctions between the major alethic modal logics.

## 5.2 The Language of Modal Logic

Now let's get down to business.

As the title of the present chapter indicates, we restrict ourselves in it to *propositional* modal logic (PML). In order to obtain this class of logics, we extend the language of propositional logic with the two afore-seen alethic modalities, necessity and possibility, each of which is represented by an operator that can be applied to formulae. To repeat, necessity we express using the symbol $\Box$, and possibility using $\Diamond$. Our modal language is an extension of the propositional logic, so if $\phi$ is any propositional formula, then so are $\Diamond\phi$ and $\Box\phi$, and of course, this holds if $\phi$ is a formula of the modal logic. For instance, $\Box P \Leftrightarrow \Diamond(\Box Q \vee \Diamond \Box R)$ and $\Box(P \Rightarrow (\Diamond P \vee \neg\Box\neg P))$ are wffs of propositional modal logic.

One might wonder why we chose to extend propositional logic with modal operators rather than extending first-order logic. Though first-order modal logics (i.e., modal logics that include quantification, equality, &c.) have been studied, they are significantly more complex, and are best approached after one has an understanding of how propositional modal logics work.

## 5.3 Inference Rules for Modal Logic

As in propositional and first-order logic, we follow the general trend of having an introduction and an elimination rule/schema for each operator in our logic. For PML, we take the same approach, hence we introduce introduction and elimination rules for both necessity ($\Box$) and possibility ($\Diamond$). Two of these rules are relatively simple; the other two are a bit more complex. But before we cover this quartet of rules, we introduce a helpful rule that simply reflects the fact that according to the standard interpretation of logical possibility and logical necessity, these two concepts are interchangeable — with a little help from negation.

### 5.3.1 Modal Dualities

Do you agree that from 'All birds can fly' one can deduce 'It's not the case that there's a bird who can't fly'? You should. For it's part of the very meaning of universal and existential quantification that the two are interchangeable — with a little help from negation. That is, $\forall$ can be replaced with $\neg\exists\neg$, and *vice versa*; and $\exists$ can be replaced with $\neg\forall\neg$, and *vice versa*. As astute readers will know, this is old hat for us, since it was presented in Chapter 3; indeed it was proved there. A perfectly analogous situation holds with respect to $\Box$ and $\Diamond$, where the former operator is thought of as the universal quantifier, and the latter as the existential. That is, $\Box\phi$ can be changed to $\neg\Diamond\neg\phi$, and $\Diamond\phi$ can be changed to $\neg\Box\neg\phi$.

The rule modal dualities allows one to switch between $\Diamond$ and $\Box$, and to "move" negations through modal operators left-to-right or right-to-left; in both cases of such movement, one simply switches the operator to the other. For example, if we have as a premise that

$$\neg\Diamond\phi,$$

we can move the negation sign through to obtain

$$\Box\neg\phi.$$

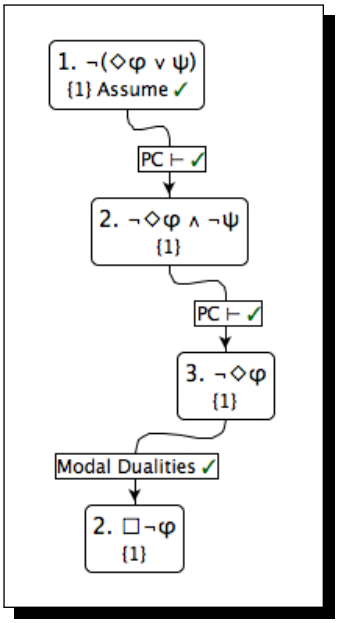
Figure 5.3 shows a simple proof that uses the rule modal dualities. Notice that this proof uses the oracle for propositional reasoning, and our first new inference schema for modal logic. This schema is valid for all the PML systems available in Slate; specifically, it's available for the four systems on which the current version of this chapter focuses: **T** (= **M**), **S4**, **S5**,and **D**. You should verify this now in Slate.[8]

### 5.3.2 Necessity Elimination

We next pass to perhaps the easiest and most intuitively plausible rule of inference in all of modal logic: necessity elimination. By necessity elimination we are entitled to deduce from $\Box P$ that $P$. That is, if $P$ is necessary, then $P$, in fact, is the case.

[8]The most efficient way to do that is to open an **S5** workspace, encode a relevant biconditional into a node (e.g., $\Box\phi \longleftrightarrow \neg\Diamond\neg\phi$), and check whether this a theorem according to the oracle for **S5**, then for **S4**, then for **D**, and finally for **T** (= **M**).

Figure 5.3: A Very Simple Proof Using Modal Dualities




Does this strike you as a valid rule? We hope so! It's this rule that allows you, with Slate, to quickly build a valid proof corresponding to the argument above regarding triangles. In short, that it's necessarily the case that the interior angles in a triangle sum to 180 degrees is captured by, for instance,

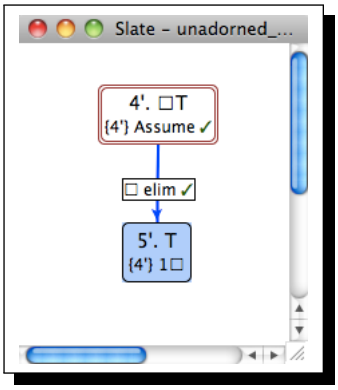
$$\Box T,$$

where $T$ is an atomic formula representing that the interior angles sum to 180 degrees. And then from this by necessity elimination we can immediately deduce $T$ on its own. This inference accomplished in Slate as a one-step proof is shown in Figure 5.4.

### 5.3.3 Possibility Introduction

Another straightforward rule is possibility introduction, which simply regiments the intuitively obvious fact that whatever is, is possible. For example, you exist, so clearly it's the case that it's logically possible that you exist. To many folks, possibility introduction is so obvious that they marvel over the fact that it can be useful, let alone indispensable. If you think back to FOL, and to the rule of existential introduction, you will have a sense of possibility introduction. In FOL, recall, we can infer by existential introduction from for example $\mathsf{R}(a)$ that $\exists x \mathsf{R}(x)$. Analogously, we can in our systems of PML infer from $\phi$ to $\Diamond \phi$ by possibility introduction. We encourage you to verify this in Slate.

Figure 5.4: A Simple, Unadorned Use of the Necessity Elimination Rule




### 5.3.4 Possibility Elimination

Now we make an immediate recommendation that may to a degree set us apart from established practice: The rule of possibility elimination is better thought of as possibility *propagation* or possibility *extension*, than by its standard name. The reason is that in an application of possibility elimination, one starts with a formula of the form $\Diamond\phi$, proceeds to show via a sub-proof that under the assumption that $\phi$ holds, it absolutely *must* be the case that $\psi$ holds, and then deduces that $\psi$ is itself possible, that is that $\Diamond\psi$. Do you see how this pattern of deduction can be thought of as extending the possibility of $\phi$?

An intuitive example may help, at least to a degree. Suppose that we know (on the basis of reasons that needn't detain us) that it's possible that there's life on the distant planet Kork. Suppose in addition that we know beyond a shadow of a doubt that it's necessary that life on *any* planet requires a certain life-supporting sub-atomic mechanism we can dub $\mathscr{M}$. Under these assumptions, can't we infer that $\mathscr{M}$ is itself quite possibly operating on Kork? Give this query some thought if your initial reaction is a negative one, or if you're just not sure. ... Again, under the assumptions in question, isn't it clear that $\mathscr{M}$ is possible? We expect that if you've pondered the case at hand, you will answer in the affirmative.

Of course, this is an informal example. What does using possibility elimination look like in Slate? Take a look at Figure 5.5. This figure shows an example of possibility elimination in action; specifically, it's established that if a conjunction is possible, then each conjunct itself is possible. The proof is overall a case of something that will be quite familiar to all but dozing readers (namely, conditional introduction), but the key to the proof is exploiting the fact that it's necessary that when a conjunction holds, the conjuncts within it absolutely must as well. This fact is what allows the two possibility-elimination inferences made in the graph. A second proof involving possibility elimination is shown in Figure 5.6. This is perhaps a more interesting result: we demonstrate that if a proposition is possible, then a disjunction with it and *any* proposition is possible.

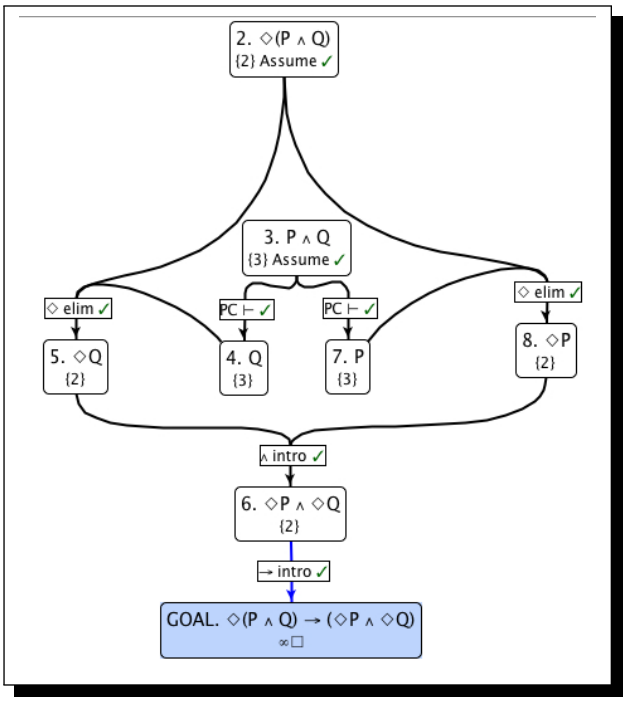



Figure 5.5: Our First Use of Possibility Elimination in Modal Logic

### 5.3.5 Necessity Introduction

We now come to the final rule in our quintet of new inference rules: necessity introduction. Mechanically speaking, this rule, as its name suggests, allows us, *under certain conditions*, to move from some formula $\phi$ to $\Box\phi$. The trickiness of the rule inheres in understanding when the conditions are right for adding a necessity operator, and this varies somewhat between the quartet **T**, **S4**, **S5**, **D** of systems. However, there is one condition under which necessity introduction is permitted in all three systems; and the condition is easily understood. That condition is theoremhood. More informatively, one can use necessity introduction on a formula $\phi$ whenever $\phi$ is a theorem. For example, here is a theorem in the propositional calculus:

$$(\mathsf{P} \wedge \mathsf{Q}) \rightarrow (\mathsf{P} \vee \mathsf{Q}).$$

Regardless of what PML workspace type you select in Slate, if you encode this formula, and encode the necessitation of it, which is

$$\Box[(\mathsf{P} \wedge \mathsf{Q}) \rightarrow (\mathsf{P} \vee \mathsf{Q})],$$

Slate will confirm that the latter node follows from the former by the rule of necessity introduction. We provide the remainder of an explanation of necessity introduction below, by explaining how the rule works in each of **T**, **S4**, **S5**, and **D**.

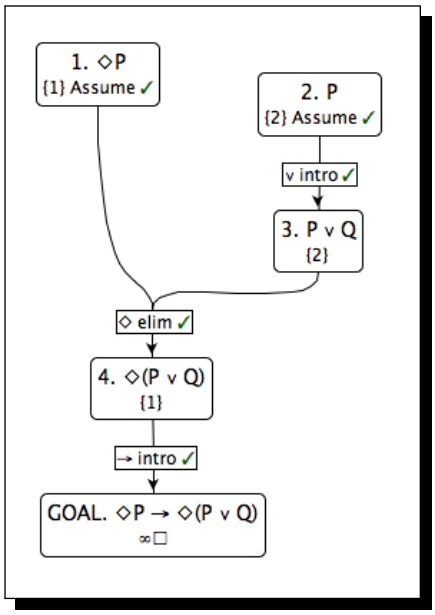



Figure 5.6: Our Second Use of Possibility Elimination in Modal Logic

## 5.4 Systems of Modal Logic

Each of **T** (= **M**), **S4**, **S5**, and **D** is available to the user of Slate once a new workspace in called for by the user.[9] The differences between these three systems are simple to state (as we show immediately below), but please realize at the outset that there are theorems provable in **S4** that cannot be proved in **T**, and theorems in **S5** which cannot be proved in **S4**. On the other hand, everything provable in **T** is provable in **S4**, and everything provable in **S4** is provable in **S5**. Because Slate offers an oracle for each logic available within its range of workspaces, the student is able to ask the system for its verdict whenever one is curious or puzzled. That is, for instance, if one is curious, while working in an **S4** workspace, as to whether some formula $\phi$ is **S4**-provable, one has only to invoke the oracle. Deep understanding of the differences between the three modal systems **T**, **S4**, and **S5**, at least to an appreciable degree, is arguably achieved when, for a given formula $\phi$, one can predict how Slate will respond when asked if $\phi$ is provable in each of these logics.

[9] Other systems are available as well, and you may have noticed that in your use and experimentation with Slate, but these systems aren't discussed in this textbook, and are there for your exploration, and for your instructor to discuss and demonstrate in class, if that is desired. The diligent student may for instance note that there is available a logic called '**K**.' Can you guess what the characteristic axiom schema of **K** is, after reading about each of **T**, **D**, **S4**, and **S5**? Are theorems in **T** also theorems in **K**?

### 5.4.1 T

The system T, which can be invoked in Slate by choosing 'T' from the available options for a workspace, allows the five inference rules discussed above.[10] By this time you should have reconstructed the proofs shown in earlier figures in this chapter. Specifically, you should know that in T, from a formula of the form $\Box \mathsf{P}$ one cannot directly add additional "boxes." That is, one cannot deduce from $\Box \mathsf{P}$ by necessity introduction that $\Box\Box \mathsf{P}$. On the other hand—and you should verify this now—, in T any theorem of the propositional calculus can be instantly declared necessarily true. For example, the PC oracle in Slate will agree that $P \rightarrow P$ is a theorem of the propositional calculus, and once the oracle certifies this in T, the system itself will tell you that you now can add an *infinite* number of boxes to the left of this formula; this is shown by the appearance of $\infty\Box$ in the node that holds $P \rightarrow P$. Please note that the system T is associated with a **characteristic axiom schema**, namely,

**T** $\Box\phi \rightarrow \phi$,

and that instances of this schema are theorems not only in T, but in S4 and S5, to which we now turn. We recommend that you verify now in a T workspace in Slate that the oracle for T agrees that **T** is a theorem, and that you construct your own proof of this schema.

#### 5.4.1.1 Exercises for System T

1. Prove that "modalized" *modus ponens* is valid in **T**. I.e., prove:

   $$\{\Box(\phi \rightarrow \psi), \Box\phi\} \vdash \Box\psi.$$

   To save you some encoding time, a file has been prepared for you ahead of time:

   `quantified_modus_ponens_in_T.slt`.

2. Prove:

   $$\vdash (\Box(\phi \rightarrow \psi) \wedge \neg\psi) \rightarrow \neg\phi.$$

   A file has been prepared for you ahead of time:

   `relative_of_mt.slt`.

[10] The system T is equivalent in all respects relative to our inquiry to M, and this fact may be reflected in your version of Slate. E.g., when you ask Slate to open a workspace, you may see that one of your options is labeled with *both* 'T' and 'M.' In addition, when, in such a workspace, you wish to appeal to the oracle at the level of T, you may be presented with an option that is the oracle at the level of M. Don't worry; just select the M-level oracle. The quirks of history are responsible for the use of both 'T' and 'M' to denote the same system: It was apparently Robert Feys who first referred to T by name (but not long thereafter folks had taken to regarding it to be logic within the great C.I. Lewis' family of modal logics, despite his not having treated it). It was Wright who called T 'M'—but he did so *after* Feys' use of 'T.' Have we confused you sufficiently? Fortunately, this is intellectual history, not the rather tidier field of logic, to which we now gladly return.

3. Prove:
$$\{\Box(\phi \vee \psi), \Box\neg\phi\} \vdash \Box\psi.$$
A file has been prepared for you ahead of time:

   `L-P_or_Q-LnotP_impliesLQ.slt`.

4. Now here is a theorem you should prove using possibility elimination. Remember that this inference rule, for reasons we've explained, can be regarded to *propagate* that which is possible. As a hint in this case, you want to propagate $\Diamond\psi$. Now here is what you are to prove:
$$\vdash (\Box\phi \wedge \Diamond\psi) \rightarrow \Diamond(\phi \wedge \psi).$$
Once again, a file has been prepared for you ahead of time:

   `Lp_and_Mp_then_M-p_and_q.slt`.

   Though this isn't an intrinsically hard problem, it can be a bit tricky because it's your first attempt to use possibility elimination, a rule that is *itself* tricky. If you feel you must look at the solution, it's

   `Lp_and_Mp_then_M-p_and_q_sol.slt`.

5. Prove:
$$\vdash \neg\Diamond\phi \rightarrow \neg\Box\phi.$$
The pre-prepared file in this case is:

   `notMp_then_notLp.slt`.

6. Prove:
$$\vdash (\Diamond\phi \wedge \neg\Diamond\psi) \rightarrow \Diamond(\phi \vee \psi).$$
The pre-prepared file is:

   `Mp_and_notMq_then_M-p_or_q.slt`.

### 5.4.2 S4

S4 is just like T, save for one important difference: Any time you have a formula of the form $\Box P$ in the workspace, you can add any number of boxes to the left. This is shown in the node for the formula in question; specifically by the now-familiar $\infty\Box$. Despite the fact that Slate will always notify you via the appearance of this information that you have the opportunity to add necessity operators, it's worth knowing that the characteristic axiom schema of the system S4 (which conveys the fact that in S4, whenever something is necessary, it's necessary that it's necessary, and so on *ad infinitum*) is:

**4** $\Box\phi \rightarrow \Box\Box\phi$

You should now verify in an S4 workspace in Slate that the oracle for S4 confirms **4** to be a theorem, and that you construct your own proof establishing this.

### 5.4.3 S5

S5 is just like S4, save for one important difference: Any time you have a formula of the form $\Diamond \mathrm{P}$ in an S5 workspace, you can add any number of boxes to the left. This is shown in the node for the formula in question; specifically by the now-ultra-familiar $\infty \Box$. Despite the fact that Slate will always notify you via the appearance of this information when you can add necessity operators, it's worth knowing that there is a characteristic axiom of the system S5 which conveys the fact that in this logic, whenever something is possible, it's necessary that it's possible, and so on *ad infinitum*. The characteristic axiom schema is:

**5** $\Diamond \phi \rightarrow \Box \Diamond \phi$

You should now use the oracle for **S5** to verify that Slate declares this to be a theorem. And then you should proceed to find a proof of the schema **5** on your own.

### 5.4.4 D = SDL (= 'Standard Deontic Logic')

We here introduce what is known as 'Standard Deontic Logic' (**SDL**), which in Slate is the system **D**. Deontic logic is the sub-branch of logic devoted to formalizing the fundamental concepts of morality; for example, the concepts of *obligation*, *permissibility*, and *forbiddenness*. The first of these three concepts can apparently serve as a cornerstone, since to say that $\phi$ (a formula representing some state-of-affairs) is permissible seems to amount to saying that it's not obligatory that it not be the case that $\phi$ (which shows permissibility can be defined in terms of obligation), and to say that $\phi$ is forbidden would seem to amount to it being obligatory that it not be the case that $\phi$ (which of course appears to show that forbiddenness is directly buildable from obligation). This interconnected trio of ethical concepts is a triad explicitly invoked and analyzed since the end of the $18^{th}$ century, and the importance of the triad even to modern deontic logic would be quite hard to exaggerate.[11]

SDL is traditionally axiomatized by the following:[12]

**SDL**

**TAUT** All theorems of the propositional calculus.

**OB-K** $\odot(\phi \rightarrow \psi) \rightarrow (\odot\phi \rightarrow \odot\psi)$

**OB-D′** $\odot\phi \rightarrow \neg \odot \neg\phi$

**MP** If $\vdash \phi$ and $\vdash \phi \rightarrow \psi$, then $\vdash \phi$

[11] Whether the triad is *adequate* as a foundation for formalizing ethics (as opposed to being central to such a project) is a very different matter. For a seminal analysis that, we believe, reveals the inadequacy of the triad, and points the way forward to the kind of beyond-the-triad-richness that is ultimately needed for the project, see (Chisholm 1982). The more recent (Bringsjord 2015*a*) builds upon Chisholm's critique in the service of engineering an ethical AI/robot.

[12] Here we draw directly from the entry "Deontic Logic" in the online *Stanford Encyclopedia of Philosophy*, by Paul McNamara; but we do make some changes to fit our purposes. See http://plato.stanford.edu/entries/logic-deontic.

**OB-NEC** If $\vdash \phi$ then $\vdash \odot\phi$

The symbol $\odot$ is new. It is intended to represent 'it is obligatory that.' Hence the formula $\odot\phi$ says that it's obligatory that $\phi$ be the case. Hence **OB-K** says that if it's obligatory that some conditional hold, then, if it's obligatory that the antecedent of this conditional hold, it's obligatory that the consequent of this conditional be the case. In Slate, when system **D** is selected, you will instantly find yourself in control of a workspace that enables you to prove everything that is provable from the list of axiom schemata and inference rules given immediately above. There is one twist, however: You may be using a version of Slate in which a **D** workspace doesn't offer you $\odot$, but rather $\Box$ instead, and doesn't offer you $\blacklozenge$, but rather $\Diamond$. You shouldn't hesitate to verify now on your own that you can obtain some classic theorems in **SDL**. To save you some time in doing so, open the file `SDL.slt`. What you see when you do so should correspond closely to what is shown in Figure 5.7; there you will see that the two axiom schemata **OB-K** and **OB-D**′ have been pre-installed. You will also see that four theorems have been pre-loaded into the workspace — but of course you will need to figure out how to prove them. In addition, make sure that you can create some proofs corresponding to:

**OB-RM** If $\vdash \phi \rightarrow \psi$ then $\vdash \odot\phi \rightarrow \odot\psi$.

**OB-RE** If $\vdash \phi \leftrightarrow \psi$, then $\vdash \odot\phi \leftrightarrow \odot\psi$.

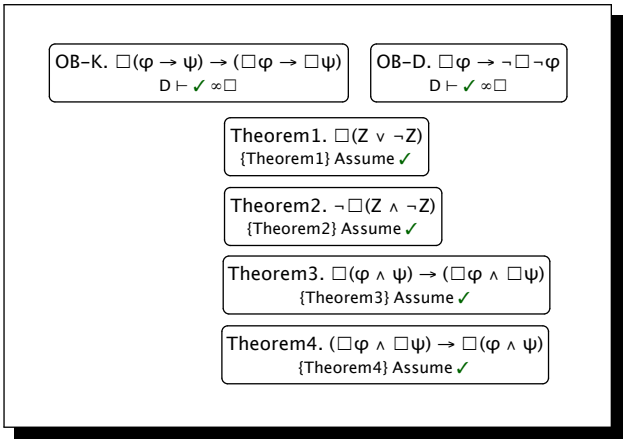



Figure 5.7: The Initial Configuration Upon Opening the File `SDL.slt`

We've also provided a Slate file that captures an augmentation of **SDL**: **SDL**$^+$; see `SDL+.slt`. **SDL**$^+$ includes a new axiom: viz., that it's obligatory that: if it's obligatory that $\phi$ then $\phi$. The reader should verify both that $\odot(\odot\phi \rightarrow \phi)$ is not provable in **SDL** (which of course immediately entails that the addition of this axiom constitutes a *bona fide* augmentation), and, secondly, that

$$\odot\odot\phi \rightarrow \odot\phi$$

is indeed considered a theorem of **SDL**$^+$ by Slate.

#### 5.4.4.1 Chisholm's Paradox and SDL

There are a host of problems that, together, constitute what is probably a fatal threat to **SDL** as a model of human-level ethical reasoning. We discuss in the present section the first of these problems to hit the "airwaves": Chisholm's Paradox (CP) (Chisholm 1963). CP can be generated in Slate, you we shall see. But before we get to the level of experimentation in Slate, let's understand the scenario that Chisholm's imagined.

Chisholm's clever scenario revolves around the character Jones.[13] It's given that Jones is obligated to go to assist his neighbors, in part because he has promised to do so. The second given fact is that it's obligatory that, if Jones goes to assist his neighbors, he tells them (in advance) that he is coming. In addiiton, and this is the third given, if Jones *doesn't* go to assist his neighbors, it's obligatory that he not tell them that he is coming. The fourth and final given fact is simply that Jones doesn't go to assist his neighbors. (On the way to do so, suppose he comes upon a serious vehicular accident, is proficient in emergency medicine, and (commendably!) seizes the opportunity to save the life (and subsequently monitor) of one of the victims in this accident.) These four givens have been represented in an obvious way within four formula nodes in a Slate file; see Figure 5.8. (Notice that $\Box$ is used in place of $\odot$.) The paradox arises from the fact that Chisholm's quartet of givens, which surely reflect situations that are common in everyday life, in conjunction with the axioms of **SDL**, entail outright contradictions (see Exercise 2 for **D** = **SDL**, in §5.4.4.2).

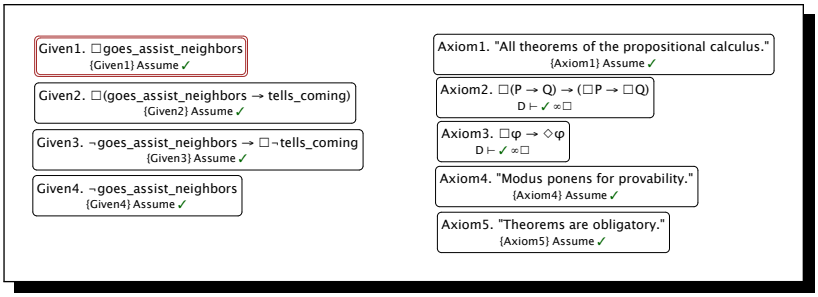



Figure 5.8: The Initial Configuration for Chisholm's Paradox (upon opening file `Chisholms_Paradox.slt`)

#### 5.4.4.2 Exercises for System D = SDL

1. Prove the following theorems in **D**:

   (a) $\odot\odot\phi \rightarrow \odot\blacklozenge\phi$

   (b) $\odot\odot\phi \rightarrow \blacklozenge\blacklozenge\phi$

   (c) $\odot\phi \rightarrow \blacklozenge(\psi \rightarrow \phi)$

[13] We change some particulars to ease exposition; generally, again, follow, the *SEP* entry on deontic logic (recall footnote 12). The core logic mirrors (Chisholm 1963), the original publication.

(d) $[\odot\phi \wedge \odot(\phi \rightarrow \psi)] \rightarrow \blacklozenge\psi$

(e) $\neg(\odot\phi \wedge \odot\neg\phi)$

(f) $\neg[\odot(\phi \wedge \psi) \wedge \odot(\phi \rightarrow \neg\psi)]$

(g) $(\blacklozenge\neg\phi \vee \blacklozenge\neg\psi) \vee \blacklozenge(\phi \wedge \psi)$

2. Prove that from Chisholm's four givens, and the axioms of **SDL**, a contradiction $\zeta \wedge \neg\zeta$ can be deduced. (We suggest that you open `Chisholms_Paradox.slt` and to save time proceed from what has been pre-installed for you therein.)

## 5.5 Semantics

When are formulae in the modal systems introduced above (i.e., **K**, **T**, **S4**, and **S5**) true, and when are they false? In the case of the propositional calculus, as we saw in Chapter 2, formulae are true or false on a given truth-value assignment, which is nothing more than a function from the set of all atomic propositional variables, to the set composed of TRUE and FALSE. Given such an assignment, with help from definitions of the truth-functional connectives (definitions which we gave in the form of truth-tables), it's a straightforward mechanical process to compute the truth-value of any wff, however complex. Assuming that some rust hasn't already set in, you can for instance quickly calculate the truth-value of $\mathsf{P} \rightarrow (\mathsf{Q} \vee \mathsf{R})$ if you are told that FALSE is assigned to $\mathsf{P}$.[14]

Unfortunately, this pleasantly simple process, now that modal formulas are in the picture, is insufficient; this is easy to see. For suppose that we know that $\mathbf{??}_1 9$ is a propositional variable, and that it has been assigned the value TRUE by some truth-value assignment $\nu$. What now is the truth-value of the formula $\Box \mathsf{P}_1 9$ on $\nu$? However long you may ponder this question (assuming that you haven't had any prior exposure to modal logic), and scour foregoing material in the present book, you will not find an answer. And relative to the semantics provided for the propositional calculus, this paralysis is a severe contrast. This is so because we now see that in the case of modal formulas, even if we know the truth-value of sub-formulas, we cannot work our way out mechanically to the value of the full, "outer" formula in question. In a word, we cannot rely upon a *compositional* process to get the job done. So, what do we do? We draw upon the remarkably fertile concept, introduced by the great universal genius Leibniz, of **possible worlds**.

Leibniz's conception was that before creating our world, God had before him an array of choices for what the actual world would be like. For instance, he passed over actualizing a possible world in which all the individual persons in it were incapable of learning any formal logic. He also decided against putting us on a planet without any mountains (which for Selmer, given that skiing requires mountains, is a world he's happy is non-actual!). And for Leibniz, God also decided against a world in which the only living creatures are human persons—in favor of ours, in which we can have pets who put their trust in us, and bring us considerable joy. You may not

[14] The answer is TRUE! If you didn't instantly realize this, we paternalistically suggest that you now read again section 2.5 before proceeding further in the present chapter.

buy into Leibniz's theology, of course; you may not even believe that God exists. What matters at the moment is that his seminal concept of a possible world as a comprehenisve, alternate reality provides a path toward an illuminating semantics for our new modal formulas $\Diamond\phi$ and $\Box\phi$.

To start down the Leibnizian path, we'll consider first how to systematically set out the conditions under which a formula in S5 is true or false. After we're clear on how these conditions for our possible-worlds semantics work, we'll generalize this semantics, and apply the generalized version to K, T, and S4—and then we'll return to S5. With the generalized form, we'll be able to precisely distinguish between these four propositional modal logics.

Our first move down the Leibnizian path is to harken back to truth-value assignments, and affirm a Leibnizian variant; we do as follows. We let all the atomic propositional variables be arranged in an infinite list $\mathscr{P}$:

$$\mathsf{P}_1, \mathsf{P}_2, \mathsf{P}_3, \ldots$$

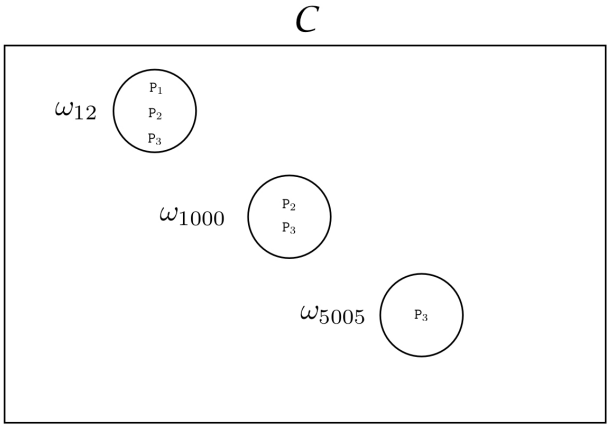
We next assume that there are an infinite number of possible worlds:

$$\mathscr{W} = \{\omega_1, \omega_2, \omega_3, \ldots\}.$$

Now let $W, W', W'', \ldots$ be sets of possible worlds such that each is a subset of $\mathscr{W}$. Next, we remind ourselves that $2^A$ is the power set of the set $AW$; that is, the set of all subsets of $A$. Given some $W \subseteq \mathscr{W}$, we define $\diamond$, a function from the set $\mathbf{Z}^+$ of positive integers to the power set of $W$.

> **Example**: Suppose that $W$ is $\{\omega_1 2, \omega_1 000, \omega_5 005\}$. Suppose in addition that $\Pi(1)$ is $\{\omega_{12}\}$, $\Pi(2)$ equals $\{\omega_{12}, \omega_{1000}\}$, $\Pi(3)$ equals $\{\omega_{12}, \omega_{1000}, \omega_{5005}\}$, and $\Pi(j)$, where $j$ is greater than 3, is the null set $\emptyset$. We have here defined a particular **canvass** $\mathscr{C}$; it is shown pictorially in Figure 5.9. The idea here is that $\mathsf{P}_1$ is true at world $\omega_{12}$ in this canvass $\mathscr{C}$, $\mathsf{P}_2$ is true at buth world $\omega_{12}$ and world $\omega_{1000}$ in this canvass, and $\mathsf{P}_3$ is true at all three possible worlds. Though it isn't shown in the figure, all the remaining propositional variables other than these three are false at each of the three possible worlds in the model $\mathscr{C}$. We introduce notation to say that a formula $\phi$ is true at a possible world $\omega$ in a canvass $\mathscr{C}$: $\models^{\mathscr{C}}_{\omega} \phi$. Hence we can (laboriously!) list out the information shown graphically in Figure 5.9 by writing the following statements:
>
> - $\models^{\mathscr{C}}_{\omega_{12}} \mathsf{P}_1$
> - $\models^{\mathscr{C}}_{\omega_{12}} \mathsf{P}_2$
> - $\models^{\mathscr{C}}_{\omega_{12}} \mathsf{P}_3$
> - $\models^{\mathscr{C}}_{\omega_{1000}} \mathsf{P}_2$
> - $\models^{\mathscr{C}}_{\omega_{1000}} \mathsf{P}_3$
> - $\models^{\mathscr{C}}_{\omega_{5005}} \mathsf{P}_3$

Figure 5.9: A Simple Canvass with Three Possible Worlds




$\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3, \ldots$, each member of which is, one, a set of possible worlds, and two, is such that $\mathbf{P}_j$ is the set of worlds at which the corresponding atomic propositional formula $\mathsf{P}_j$ is true. For example, the atomic propositional formula $\mathbf{??}_4 4$ would be true at all and only those worlds in $\mathbf{P}_4 4$.

We now imagine a thing that might aptly be called a "cosmic canvass," which is a pair composed of a set $W$ of possible worlds drawn from $\mathscr{W}$, and a particular $\mathbf{P}_j$. We shall use '$\mathscr{C}$' (and, if need be, $\mathscr{C}'$, etc.) to refer to such a canvass. Note that a canvass includes not just th

- $\models^{\mathscr{C}}_{\omega} \mathsf{P}_i$ iff $\omega \in \mathbf{P}_i$
- $\models^{\mathscr{C}}_{\omega} \neg\phi$ iff it's not the case that $\models^{\mathscr{C}}_{\omega} \phi$.
- $\models^{\mathscr{C}}_{\omega} \phi \wedge \psi$ iff both $\models^{\mathscr{C}}_{\omega} \phi$ and $\models^{\mathscr{C}}_{\omega} \psi$.
- $\models^{\mathscr{C}}_{\omega} \phi \vee \psi$ iff either $\models^{\mathscr{C}}_{\omega} \phi$ or $\models^{\mathscr{C}}_{\omega} \psi$.
- $\models^{\mathscr{C}}_{\omega} \phi \rightarrow \psi$ iff if $\models^{\mathscr{C}}_{\omega} \phi$, then $\models^{\mathscr{C}}_{\omega} \psi$.
- $\models^{\mathscr{C}}_{\omega} \phi \leftrightarrow \psi$ iff $\models^{\mathscr{C}}_{\omega} \phi$ iff $\models^{\mathscr{C}}_{\omega} \psi$.

The list of conditions immediately above, as no doubt you can see, is essentially a recasting of the truth-conditions for formulae in the propositional calculus into the Leibnizian scheme. You may therefore be wondering what the point is. Well, the scheme pays dividends when we come to modal formulas; specifically, Leibniz's idea is that $\Box\phi$ is true in a canvass exactly when $\phi$ is true in every possible world $\omega$ in that cosmic canvass; $\Diamond\phi$ is the case in a canvass provided that $\phi$ is true in at least one possible world. Here's the pair of conditions:

- $\models^{\mathscr{C}}_{\omega} \Box\phi$ iff for every world $\omega'$ in $\mathscr{C}$, $\models^{\mathscr{C}}_{\omega'} \phi$.
- $\models^{\mathscr{C}}_{\omega} \Diamond\phi$ iff there is at least one world $\omega'$ in $\mathscr{C}$ such that $\models^{\mathscr{C}}_{\omega'} \phi$.

# Chapter 6

# Theories

Most logic problems in most introductory logic courses and textbooks feature propositions bereft of long-term, real-world importance. The main reason for this is twofold: one, it makes great pedagogical sense to cut one's logical teeth on problems that have been engineered to do the cutting, and two, such problems aren't the focus of professionals who use logic. For example, we have seen problems, in the modal-logic case, that are crafted to highlight the different characteristic axioms of the propositional modal logic systems covered in Chapter 5. But command of these characteristic axioms is second nature to professionals who in the course of their jobs use modal logic. So far, confessedly, at least for the most part, pedagogical problems have been the focus of our attention in the present volume — but the time has come to shift gears so that we can reach some things that have long-term, real-world importance. Bhat do we mean by propositions that have no "long-term, real-world importance"? Well, consider the propositions expressed by these two statements, which you no doubt remember from earlier:

- Everyone likes anyone who likes someone.
- Larry likes Lucy.

As you should now be able to appreciate (and at this point outright prove in a snap!), from this pair it can be established that the following proposition follows deductively.

- Lucy likes herself.

So the third sentence follows deductively from the first two. For many people, seeing this consists in a minor revelation, because at first glance it appears that the first two propositions deductively imply only that everyone likes Larry. Yet, surely it must be admitted that in the grand scheme of things, this is a rather minor revelation. After all, Larry and Lucy are (here) imaginary llamas, and the first proposition, in real life, is false. (We all know there are not only agents who fail to like likers, but there are probably as well some agents who like absolutely no one. This seems true in the human sphere, if not in the animal one.) In short, even students who

have committed these three propositions (and representations of them in first-order logic) to memory for life will not thereby have done something particularly significant. To put the point another way, it would surely be an odd thing to assert that the first two of these propositions constitute a theory of liking between people.

Wouldn't it be nice if our study of logic could progress to the study of sets of formulae (and hence the propositions they encode) that *are* of lasting importance to matters that truly matter? You may not agree that the answer to this question is "Yes," but we do assume you agree it would at least be a nice change of pace to consider sets of formulas that are at least *candidates* for being of lasting importance.

## 6.1 What is a Theory, Exactly?

Such candidates for us will be called — following terminological tradition — **theories**. Doubtless you are in general familiar with the concept of a scientific theory. You may have heard of "the theory of evolution," or "the general theory of relativity," and so on. In the field of formal logic, a theory is generally a good deal more rigorous than what counts as a theory in the physical sciences. Once we have developed some understanding of the logician's sense of "theory," we shall return to investigate, albeit briefly, the relationship between this sense and those things called theories in the physical sciences. We shall see that in the case of physics, things, of late, are becoming more rigorous courtesy of the use of formal logic to express theories in this field.

So, what is a theory for us, here? A theory $\mathscr{T} = (L, \Phi, \mathscr{P})$ will be composed of the following three elements.

- A **language** L. As you know by now, we are talking here about a *formal* language, which means that an alphabet has been specified, and so has a formal grammar which generates well-formed formulas (wffs) on the basis of this alphabet.
- A set $\Phi$ of formulas usually referred to as "axioms." (This is the reason that the concept of a theory being presented here is often said to be the concept of an **axiomatic** theory.)
- A **proof theory** $\mathscr{P}$; e.g. the natural-deduction style proof theory of first-order logic embodied in the Slate system, with which you are now quite familiar.

Given this definition of a theory, one can immediately invent simple examples in order to cultivate some understanding of the definition. (Such examples, of course, will not yet take us to theories that are objects of study by professionals in the real world.) Here's one, a theory that is supposed to help a group of logic-minded humans figure out whether any or all of three llamas — Don, Ed, and Frank — they have at their disposal can carry a given load of a given weight from a given location $l$ to another location $l'$, in the Andes Mountains. However high the maximum altitude of every point along the route connecting every two locations, all three llamas will be fine. However, while llamas are high-altitude creatures conditioned by millennia of experience to shrug of the effects of altitude, the weight of their loads, and the distances of their travel with loads, are a concern. When a llama regards its load to simply be too heavy, it will simply lay down, and may spit in disgust to emphasize the point that it plans to go nowhere until its pack is lightened. The humans here are

clever enough to realize that such a disenchanted and motionless llama is very bad for the transport business. Their solution is to establish a theory, implement that theory in a computer, and query the computer for a check before loading up a llama, and setting out from the point of origin to the target destination. Our humans have managed to work out part of the theory they seek; that part follows immediately below. Since we don't have on hand a theory of arithmetic (yet; arithmetical theories are soon to be presented and discussed), we use a small range of symbolic values for both weight and distance.[1] Please study the partial theory **LLama Loading** now.

**Llama Loading**

| | | | |
|---|---|---|---|
| **A1** | maxWeight(don, light) | **A7** | dist(iquique, huavina, near) |
| **A2** | maxWeight(ed, medium) | **A8** | dist(huavina, chusmiza, near) |
| **A3** | maxWeight(frank, heavy) | **A9** | dist(chusmiza, cariquima, far) |
| **A4** | maxDist(don, medium) | **A10** | weight(pack1, heavy) |
| **A5** | maxDist(ed, near) | **A11** | weight(pack2, medium) |
| **A6** | maxDist(frank, far) | **A12** | weight(pack3, light) |

**A13** $\mathsf{Llama}(\mathsf{don}) \wedge \mathsf{Llama}(\mathsf{ed}) \wedge \mathsf{Llama}(\mathsf{frank})$

**A14** $\mathsf{Town}(\mathsf{iquique}) \wedge \mathsf{Town}(\mathsf{huavina}) \wedge \mathsf{Town}(\mathsf{chusmiza}) \wedge \mathsf{Town}(\mathsf{cariquima})$

**Note**: The axiom that you must complete follows (**A15**); '???' is of course the missing component. Once the completion is accomplished, and you have encoded the theory **Llama Loads** in Slate, you will be able to issue queries as to whether a given llama $x$ is okay to travel from town $y$ to town $z$ carrying a particular pack $u$.

**A15** $\forall x, y, z, u\,(\mathsf{OK}(x, y, z, u) \longleftrightarrow ???$

#### 6.1.0.1 Exercises

- Using the Slate file that encodes **Llama Loading**, verify that the following formula is provable from the theory.

  $\mathsf{OK}(\mathsf{don}, \mathsf{iquique}, \mathsf{huavina}, \mathsf{pack3})$

An **interpreted** theory is a theory accompanied by a fourth element: the standard interpretation $\mathscr{I}$ for this theory. So in the case of an interpreted theory $\mathscr{T}$ it is a quadruple $\mathscr{T} = (L, \Phi, \mathscr{P}, \mathscr{I})$. The interpretation $\mathscr{I}$, as you know by now, essentially says what the relation symbols and function symbols appearing in wffs mean.

[1] Generally speaking, full-grown llamas range in weight from about 250 pounds to 450 pounds, and can carry loads that are about 30% of body weight. The place-names given here are in fact real; the relative distances between them aren't.

## 6.2 Key Definitions Relating to Theories

Here are six key definitions relating to theories. The concepts defined here will allow us to speak informatively about theories.

**Theorem of a Theory** A formula $\phi \in L$ is a theorem of $\mathscr{T}$ iff $\mathscr{T} \vdash \phi$. (Technically speaking, we should write instead $\Phi \vdash \phi$, since inference can only be made from the axioms of a theory. But we shall for convenience make reference directly to the theory itself, and we trust the reader to know from context exactly what is meant.)

**Soundness of a Theory** A theory $\mathscr{T}$ is **sound** iff every theorem of $\mathscr{T}$ is true on the standard interpretation $\mathscr{I}$ of the theory. That is, using the notation we have at hand, for every wff $\phi$, if we have $\mathscr{T} \vdash \phi$, we have $\mathscr{I} \models \phi$. (Note that soundness here isn't the same concept as soundness of a logical system.)

**Decidability of a Theory** A theory $\mathscr{T}$ is **decidable** iff there is a Turing machine (or an equivalent) which when given any formula $\phi \in L$ can determine whether or not $\mathscr{T} \vdash \phi$.

**Theory Deciding a Sentence** A theory $\mathscr{T}$ *decides* a sentence $\phi \in L$ iff either $\mathscr{T} \vdash \phi$ or $\mathscr{T} \vdash \neg\phi$.

**Completeness of a Theory** A theory $\mathscr{T}$ is **complete** iff it decides every sentence in $L$, i.e., for every $\phi \in L$, either $\mathscr{T} \vdash \phi$ or $\mathscr{T} \vdash \neg\phi$. A theory is **semi-complete** if from it every true formula can be proved. Notice that every complete theory is semi-complete, but the converse doesn't hold.

**Inconsistency of a Theory** A theory $\mathscr{T}$ is **inconsistent** iff for some formula $\phi \in L$, both it and its negation are theorems of $\mathscr{T}$.

## 6.3 Theories of Arithmetic

### 6.3.1 Elementary Arithmetic (including Tarskian Elementary Arithmetic)

Our first theories of arithmetic are meant to roughly correspond to parts of what students in the first three grades of elementary school are early on expected to understand about addition, subtraction, and — at least for the best students — multiplication. Different descriptors are used by different authors to denote the first theory we shall consider. For example, that first theory, our $\mathbf{EA}^{+\times}_{\mathscr{N}}$, short for 'Elementary Arithmetic of Addition and Multiplication' is called 'Baby Arithmetic' by Smith (2007), who provides an elegant and enlightening presentation. We'll consider some expansions and modifications of $\mathbf{EA}^{+\times}$ in the present section, and when doing so will designate these variants with suitable superscripts.

Harkening back to the three components needed for a theory, we note that in order to specify $\mathbf{EA}^{+\times}_{\mathscr{N}}$ we need to define, one, the formal language of $\mathbf{EA}^{+\times}_{\mathscr{N}}$; two, its axioms; and three, the proof theory by which deduction can be carried out. Delivering this trio is quick and easy: For deduction, we allow our now-familiar inference rules of the propositional calculus, along with Identity Elimination and Identity Introduction. More precisely, we allow what is in Slate called the *pure predicate calculus* (see §3.5.3). As to the language of $\mathbf{EA}^{+\times}_{\mathscr{N}}$, we have first the now-second-nature-to-you symbols for each of the five truth-functional connectives; the familiar

identity symbol '='; a function symbol $\dot{s}$ intended to represent the successor function on symbols denoting members of the natural numbers $\mathscr{N} = \{0, 1, 2, \ldots\}$; and the function symbols $\dot{+}$ and $\dot{\times}$, which represent standard addition $+$ and standard multiplication $\times$, respectively. The purpose of the dots above the standard symbols for the successor function, addition, and multiplication is to convey that, in the language of $\mathbf{EA}^{+\times}_{\mathscr{N}}$, these are simply arbitrarily selected symbols without any intrinsic meaning, but we *interpret* these symbols to denote, respectively, the standard successor (or increment-by-one) function, standard addition, and standard multiplication. If we wanted to, we could use novel function symbols to denote these three familiar functions. For example, to pick some options at random, $\star$ could be used to pick out addition, and $\bowtie$ could be used to denote standard multiplication. Because we are confident that readers will keep in mind the fundamental difference between the symbols in the language of $\mathbf{EA}^{+\times}_{\mathscr{N}}$, versus what these symbols mean, we shall proceed to use symbols to denote addition and multiplication that correspond to those symbols used in elementary education; that is, we shall simply use $s$ for the successor function, $+$ for addition, and $\times$ for multiplication. That said, here, with $n$ and $m$ used to denote arbitrary numbers constructed by way of iterated use of the successor function $s$, are the six schemas that together define $\mathbf{EA}^{+\times}_{\mathscr{N}}$:

$$\mathbf{EA}^{+\times}_{\mathscr{N}}$$

**Schema 1** $0 \neq s(n)$

**Schema 2** $s(n) = s(m) \rightarrow n = m$

**Schema 3** $+(n, 0) = n$

We can also write this not in prefix notation, but in infix notation. In that case, **Schema 3** becomes:

$$n + 0 = n$$

**Schema 4** $+(n, s(m)) = s(+(n, m))$

Expressed in infix notation:

$$n + s(m) = s(n + m)$$

**Schema 5** $\times(n, 0) = 0$

Expressed in infix notation:

$$n \times 0 = 0$$

**Schema 6** $\times(n, s(m)) = +(\times(n, m), n$

Expressed in infix notation:

$$n \times s(m) = (n \times m) + n$$

Notice that there are no quantifiers present in any of these six expressions. In saying that each is a **schema**, we are saying that each can be instantiated in any of a number of infinite and obvious ways. For example, each of the following three atomic formulae are instances of **Schema 1**:

$$0 \neq s(0)$$

$$0 \neq s(s(s(0)))$$

$$0 \neq s(s(s(s(s(s(0))))))$$

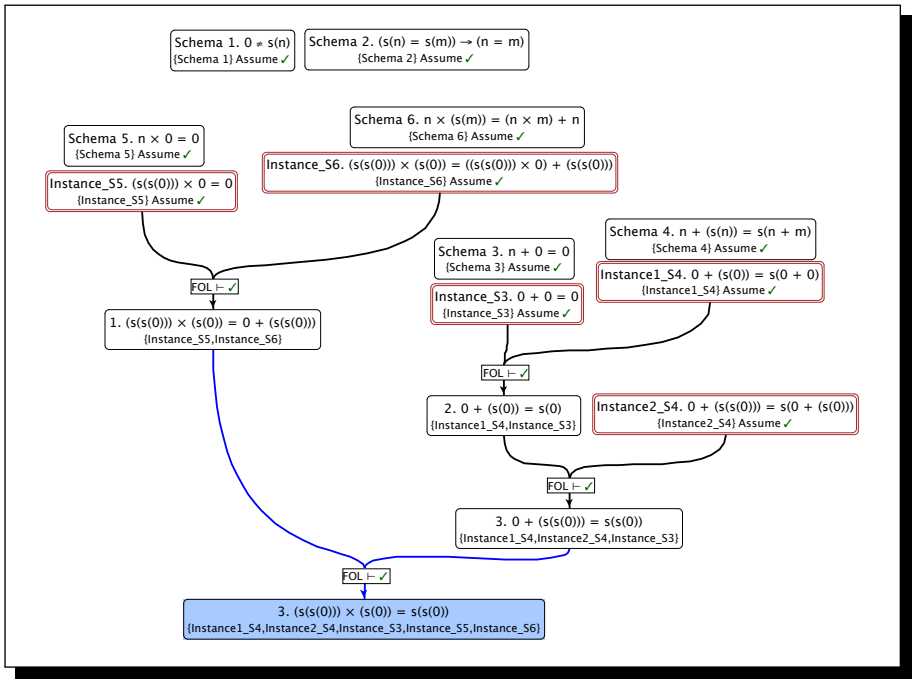
Respectively, this trio represents that 0 is other than 1, that 0 is other than 3, and that 0 is other than 6.

Now let's see if we can prove some theorems in Slate in $\mathbf{EA}^{+\times}_{\mathscr{N}}$. Take a look at Figure 6.1; there is shown a proof that $2 \times 1 = 2$. (Of course, this is technically a proof that $\dot{2} \times \dot{2} = \dot{4}$, or put at an even-lower level of abstraction, the level used in the Slate proof itself, the result is that

$$s(\dot{s}(\dot{0})) \times s(\dot{0}) = s(\dot{s}(\dot{0})).$$

Since Slate doesn't have a built-in rule for the heterodox inference of instantiating a schema (which is to be expected), to signal that an inference of this type has been made, we position the schema and its instantiation together in the workspace, as you can see.

Figure 6.1: Proof in **EA**/**BA** That $2 \times 1 = 2$




Note that we can, with the addition of three new schemata to the six that compose $\mathbf{EA}^{+\times}_{\mathscr{N}}$, move to including propositions that involve greater-than (>):

$$\mathbf{EA}^{+\times>}_{\mathscr{N}}$$

**Schemata 1–6**

**Schema 7** $s(m) > m$

**Schema 8** $m > n \rightarrow s(m) > n$

**Schema 9** $0 \not> s(m)$

Mathematics as taught in Grades 1 and 2, at least in the U.S., centers around addition and subtraction.[2] We have of course seen above some capture of addition (and multiplication, which jumps ahead grade-wise to Grade 3), but so far, nothing has been said about subtraction. Do you see what schemata should be added to capture subtraction? Deeper understanding would be greatly fostered by an attempt on your paper to answer this question before reading on ...

Here's a system that formalizes simple addition, multiplication, greater-than, *and* substraction:

$$\mathbf{EA}^{+\times>-}_{\mathscr{N}}$$

**Schemata 1–9**

**Schema 10** $m - 0 = m$

**Schema 11** $s(m) - s(n) = k - m$

#### 6.3.1.1 Exercises

1. In Slate, prove in $\mathbf{EA}^{+\times}_{\mathscr{N}}$ that:
   - $2 \times 2 = 4$
   - $2 + 3 = 5$
2. In Slate, prove in $\mathbf{EA}^{+\times>}_{\mathscr{N}}$ that:
   - $4 > 1$
   - $3 \not> 5$
3. We can move to a meta-logical perspective from which some deeper theorems involving elementary arithmetic are set as challenges, as follows. To fix matters, say that we are specifically talking about $\mathbf{EA}^{+\times}_{\mathscr{N}}$. Now, we can consider that challenge of proving that every true "real-life" arithmetic equation $e$ that can be expressed in $\mathbf{EA}^{+\times}_{\mathscr{N}}$ can be proved in this system as well (in Slate). For instance, we know that $5 \times 5 = 25$; do we also know that it can be proved in Slate that $\dot{5} \times \dot{5} = 25$? The correct answer is "Yes," but can you prove this?
4. Tarski (1995) specified and partially explored what we can label $\mathbf{TEA}^{<>}_{\mathscr{N}}$, which has the following five axioms.

   **Axiom 1** $\forall x, y(x = y \vee x < y \vee x > y)$

   **Axiom 2** $\forall x, y(x < y \rightarrow y \not< x)$

[2] The current Common-Core standards for mathematics, for K–12, are available at http://www.corestandards.org/Math

**Axiom 3** $\forall x, y(x > y \rightarrow y \not> x)$
**Axiom 4** $\forall x, y, z[(x < y \wedge y < z) \rightarrow x < z]$
**Axiom 5** $\forall x, y, z[(x > y \wedge y > z) \rightarrow x > z]$

When he presented this system, he proceeded to prove, over the course of a number of pages, six theorems, the sixth being what he called "The Law of Trichotomy." Figure 6.2 shows a Slate workspace that contains the axioms and theorems in question (and you'll notice that Slate has found proofs of the first and last theorems). Provide a proof for each of the six theorems, from the five axioms. Of course, while you can use oracular reasoning along the way, your finished products should make no use of any oracle. For you convenience, you can make use of the Slate file `Tarski_EA_lessthan_greaterthan.slt`. Here is what might be a handy list of the theorems to be demonstrated. Notice that the sixth theorem is called 'The Law of Trichotomy' (and here we follow Tarski's language).

**Theorem 1** $\neg\exists x(x < x)$
**Theorem 2** $\neg\exists x(x > x)$
**Theorem 3** $\forall x, y(x > y \rightarrow y < x)$
**Theorem 4** $\forall x, y[x \neq y \rightarrow (x < y \vee y < x)]$
**Theorem 5** $\forall x, y[x \neq y \rightarrow (x > y \vee y > x)]$
**Law of Trichotomy** $\forall x, y[(x = y \vee x < y \vee x > y) \wedge (\neg(x = y \wedge x < y)) \wedge (\neg(x = y \wedge x > y)) \wedge (\neg(x < y \wedge x > y)) \wedge (\neg(x = y \wedge x < y \wedge x > y))]$

5. Tarski (1995) augmented $\mathbf{TEA}^{<>}_{\mathscr{N}}$ with the relations of less-than-or-equal ($\leq$) and greater-than-or-equal ($\geq$), thereby giving rise to what we can dub $\mathbf{TEA}^{<>^{\leq\geq}}_{\mathscr{N}}$. This new theory of elementary arithmetic takes as axioms that five that anchor $\mathbf{TEA}^{<>}_{\mathscr{N}}$, plus this definition:

   **Definition1** $\forall x, y[x \leq y \longleftrightarrow (x = y \vee x < y)]$

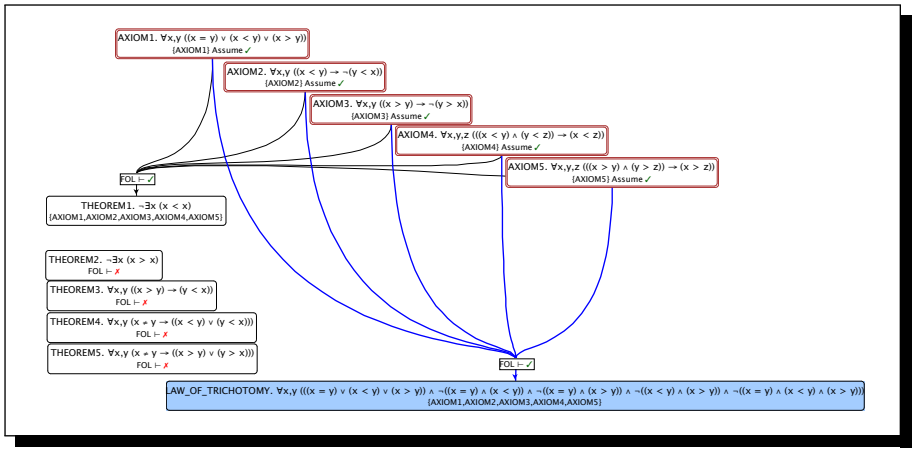
   Your challenge is to prove the following five theorems in $\mathbf{TEA}^{<>^{\leq\geq}}_{\mathscr{N}}$, the pre-stocked Slate file for which is `Tarsk_EA_lessthaneq_greaterthaneq.slt`.

   **Theorem 7** $\forall x, y(x \leq y \longleftrightarrow x \not> y)$
   **Theorem 8** $\forall x, y[x < y \longleftrightarrow (x \leq y \wedge x \neq y)]$
   **Theorem Ex8a** $\forall x, y, z[(x < y \wedge y \leq z) \rightarrow x < z]$
   **Theorem Ex8b** $\forall x, y, z[(x < y \wedge y < z) \rightarrow x < z]$
   **Theorem Ex8c** $\forall x, y, z, u[(x \leq y \wedge y < z \wedge x < u) \rightarrow x < u]$

### 6.3.2 Robinson Arithmetic ($\mathscr{Q}$)

**Axiom 1** $\forall x(0 \neq s(x))$
**Axiom 2** $\forall x \forall y(s(x) = s(y) \rightarrow x = y)$
**Axiom 3** $\forall x(x \neq 0 \rightarrow \exists y(x = s(y)))$
**Axiom 4** $\forall x(+(x, 0) = x)$

Figure 6.2: **TEA**$^{<>}_{\mathcal{N}}$ With Six Theorems




**Axiom 5** $\forall x \forall y(+(x, s(y)) = s(+(x, y)))$

**Axiom 6** $\forall x(\times(x, 0) = 0)$

**Axiom 7** $\forall x \forall y(\times(x, s(y)) = +(\times(x, y), x))$

### 6.3.3 Peano Arithmetic (PA)

**Axiom 1** $\forall x(0 \neq s(x))$

**Axiom 2** $\forall x \forall y(s(x) = s(y) \rightarrow x = y)$

**Axiom 3** $\forall x(+(x, 0) = x)$

**Axiom 4** $\forall x \forall y(+(x, s(y)) = s(+(x, y)))$

**Axiom 5** $\forall x(\times(x, 0) = 0)$

**Axiom 6** $\forall x \forall y(\times(x, s(y)) = +(\times(x, y), x))$

**Induction Schema** Every sentence that is the universal closure of an instance of this schema:

$$[\phi(0) \wedge \forall x(\phi(x) \rightarrow \phi(s(x))] \rightarrow \forall x \phi(x)$$

## 6.4 Theories of Sets (Axiomatic Set Theory)

In Chapter 1, "Preliminaries," specifically in the section 1.5.1, we supplied a serviceable summary of what we called "naïve" set theory. Why is this form of set theory said to be naïve? Do you remember? The answer, in a nutshell, is that as Russell's Paradox shows, such set theory is provably inconsistent: one can prove outright contradictions in it. Specifically, what we saw in §3.8.2 was that Russell was able to deduce a contradiction from Frege's (n.d.) Axiom V. In light of this we promised to provide, later, coverage of *axiomatic* set theory; the time to deliver on this promise has now come. Axiomatic set theory is decidedly non-naïve, since it is

formulated with an eyes-wide-open understanding that paradoxes can rise up and threaten unreflective use of set-theoretic concepts. There are a number of different possibilities for specifying an axiomatic set theory. We turn now to the dominant one, known by the label 'ZFC.'

### 6.4.1 ZFC

The Zermelo-Fraenkel Axioms for Set Theory, or just 'ZFC' for short, include the following nine axioms.[3][4]

**Axiom of Extensionality**

$$\forall x \forall y (\forall z (z \in x \longleftrightarrow z \in y) \rightarrow x = y)$$

**Axiom Schema of Separation**

$$\forall x_0 \ldots \forall x_{n-1} \forall x \exists y \forall z (z \in y \longleftrightarrow (z \in x \wedge \phi(z, x_0, \ldots, x_{n-1})))$$

**Pair Set Axiom**

$$\forall x \forall y \exists z \forall w (w \in z \longleftrightarrow (w = x \vee w = y))$$

**Sum Set Axiom**

$$\forall x \exists y \forall z (z \in y \longleftrightarrow \exists w (w \in x \wedge z \in w))$$

**Power Set Axiom**

$$\forall x \exists y \forall z (z \in y \longleftrightarrow \forall w (w \in z \rightarrow w \in x))$$

**Axiom of Infinity**

$$\exists x (\emptyset \in x \wedge \forall y (y \in x \rightarrow y \cup \{y\} \in x))$$

**Axiom Schema of Replacement**

$$\forall x_0 \ldots \forall x_{n-1} (\forall x \exists^{=1} y \phi(x, y, x_0, \ldots, x_{n-1}) \rightarrow \forall u \exists v \forall y (y \in v \longleftrightarrow \exists x (x \in u \wedge \phi(x, y, x_0, \ldots, x_{n-1}))))$$

**Axiom of Choice**

$$\forall x ((\emptyset \notin x \wedge \forall u \forall v ((u \in x \wedge v \in x \wedge u \neq v) \rightarrow u \cap v = \emptyset)) \rightarrow \exists y \forall w (w \in x \rightarrow \exists^{=1} z \, z \in w \cap y))$$

#### 6.4.1.1 Exercises

1. The Axiom Schema of Separation was the replacement for Axiom V. Show that Russell's reasoning fails when the attempt is made to apply it to the Axiom Schema of Separation.

2. Provide for each axiom of ZFC one clear English sentence that expresses the axiom.

[3] While it's obvious what the 'Z' and 'F' abbreviate in the label 'ZFC,' what about 'C'? This letter refers to one of the axioms that follow: the Axiom of **C**hoice. 'ZF' refers then to the following list of axioms, *without* the Axiom of Choice.

[4] Note that when we write '$\phi(x)$' we are saying that variable $x$ appears free in formula $\phi$. In the Axiom Schema of Separation, $y$ does not occur free in '$\phi(z, x_0, \ldots, x_{n-1})$.'

# Chapter 7

# Meta-Logic

If you have earnestly read the present book to this sentence, then you have spent considerable time working with Slate. In doing so, you have worked at what is often called the **object level**. One of the marks of work carried out at this level is that it involves the attempt to discover, build, and — with help from Slate's responses to the inferential links you proposed — verify proofs proceeding from some set $\Gamma$ of formulas in a formal language (e.g, the formal language of the propositional calculus) to a particular formula $\gamma$ (in this same language) that was your goal. We suspect that at some points during such activity you "stepped back," and asked yourself the question:

> Q1 "Is there a proof of $\gamma$ from $\Phi$, or could it be that I'm wasting my time and banging my head against the wall for so reason, because in fact there *isn't* such a proof?"

Probably soon afer posing this question you enlisted an oracle for help. The question, which we can of course compress to simply

$$\Phi \vdash \gamma?,$$

would probably have been answered by the oracle you selected, and that answer may have put you at least somewhat at ease. But how do you know you can trust the oracle? The only basis for deep trust of the oracle is an outright proof that it (where 'it' is suitably formalized) is indeed infallible when it comes to its tackling questions like the compressed version of Q1. In other words, you need a proof that the oracle can in fact infallibly decide whether the answer to Q1 is YES or NO. The discovery, generation, and checking of such a proof is a perfect example of meta-logic in action, because such activity is to use logic to obtain meta-theorems about a system of logic in which proofs (so-called **object-level** proofs) are obtained in order to establish object-level theorems.

Let's be a bit more careful about the situation. First, we note that our oracles are really just computing machines; and second, we remind ourselves that computing machines for us are MC-machines (see §1.6). Ergo, the question of whether an oracle in Slate can infallibly answer YES or NO to $\Phi \vdash \gamma$? becomes this question:

Q2 Is there an algorithm $\mathscr{M}$ that can be given to an MC-machine that enables that machine to infallibly answer YES or NO in response to $\Phi \vdash \gamma$??

## 7.1 Why *These* Meta-Properties of Logical Systems?

At at least one of us (hint: the older) used to be somewhat mystified as to why certain meta-properties are deemed worthy of investigation, while others are left aside. Why do logicians seem to for instance be obsessed with the soundness of a logical system? Or to put the query in the impressionistic terms used in the previous figure, on what criteria should our meta-thinker gauge the systems with which the object-level thinkers are engaged? The immediate cure for avoiding puzzlement in your own case is to think now, rather hard if need be, about what attributes *you* would like to see in a logical system. You should also give thought as to what attributes you might be willing to give up, at least in certain cases. We say no more on the issue here, for perhaps nothing sharpens the logical mind more than sustained reflection aimed at why certain logical systems may be better or worse than other logical systems, and at what metrics should be used to undergird such judgements.

## 7.2 Meta-Properties of the Propositional Calculus

In our coverage of the propositional calculus (Chapter 2), we discussed two ways of inferring formulas from other formulas. The first method, using the tabular approach, is a semantic approach, based on what particular forumulas mean; or more precisely, this approach arises from consideration of what conditions they must be assigned particular truth-values. It was through this method that we defined the **consequence** relation. Recall that a formula $\psi$ is said to be a consequence of a set of formulas $\Phi$ if and only if every truth-value-assignment that satisfies every formula in $\Phi$ also satisfies $\psi$; abbreviated courtesy of the customary notation: $\Phi \models \psi$. In the second method, we composed proofs in the propositional calculus incrementally, using 10 inference rules; five classified as introduction rules, and five classified as elimination rules. This second method gave us the **entailment** relation: A sentence $\psi$ is said to be entailed by a set of formulas $\Phi$ iff there is a proof $\rho$ of $\psi$ using only the assumptions $\Phi$ as a starting point; this relation is traditionally abbreviated $\Phi \vdash \psi$.

We determined that, in principle, we could mechanically ascertain, for any formula $\psi$ and any set $\Phi$ of formulas, whether $\Phi \models \psi$, through the use of the tabular method. But the tabular method, we saw, isn't just tedious: it's painfully unrealistic: it quickly becomes prohibitively expensive, given the space and time that would be required. (Time is money, after all; and real estate also has a price.) It is reasonable, then, to ask whether our proof calculi (and at the moment, specifically our proof calculus for the prop. calc.) can help us in determining consequence.

Two particular questions are relevant. The first question is whether entailment implies consequence; that is, if there is a proof of $\psi$ from $\Phi$ ($= \Phi \vdash \psi$), is it also the case that $\Phi \models \psi$? (Did you decide upon the reflection we urged above that this meta-logical question is important?) If our proof system has this property, it is said

to be **sound**. The second question is whether consequence implies entailment. That is, if $\psi$ is a consequence of $\Phi$ ($= \Phi \models \psi$), is there a proof in our system of $\psi$ from $\Phi$; that is, is it true that $\Phi \vdash \psi$? If our proof system has this property, it is said to be **complete**. Of course, in both cases, the system that anchors these queries is Slate.

The questions of soundness and completeness apply to any standard deductive proof calculus, not just the ones that we have developed earlier in the present book. Soundness and completeness are also defined *relative* to the particular consequence relation: A proof calculus may be sound or complete with respect to one consequence relation, but unsound or incomplete with respect to another. In addition, the properties of soundness and completeness are independent: a proof system may be sound but not complete, complete but not sound, neither sound nor complete, or sound and complete.

### 7.2.1 Soundness of the Propositional Calculus

How can we go about proving that our proof calculus for the propositional calculus is sound? First, we must be clear about the claims of consequence that we extract from our proofs. Each vertex in a proof contains a formula $\psi$ and a set of in-scope assumptions $\Phi$. Our proof system is sound just in case for every vertex with formula $\psi$ and in-scope assumptions $\Phi$, it is the case that $\psi$ is a consequence of $\Phi$ whenever $\psi$ is entailed by $\Phi$.

It's helpful to look at the situation mechanically, from the standpoint of a workspace in Slate that holds a fully valid, Slate-verified proof $\rho$. Here, $\rho$ is completely arbitrary, and for all we know might have more nodes that there are atoms in the physical universe, or it might be tiny; the hypergraph that is $\rho$, however, must be finite. Let's assume that $\rho$ has starting premises $\Delta$ (these are therefore top-level assumptions) and an eventual conclusion $\delta$. Inside $\rho$, we have individual inferences, and each of these inferences is associated with a particular rule of inference that has been named inside a vertex, and this same vertex contains the encouraging green check ✓. To look at the situation mechanically, we have only to note that soundness consists in the conditional fact that for every inference in $\rho$, if the Slate user switches the justification from the selected rule of inference from among the available 10, to the consequence oracle $\models$, another ✓will be declared by the system, *not* a ✗. Behind the scenes, of course, our infallible oracle is just using the tabular method to check whether the relevant conclusion is a consequence of the assumptions that are listed within the vertex that contains this conclusion.

At this point, then, we can state the soundness theorem for the propositional calculus in a traditional way, and then in more intuitive, mechanical way, to wit:

> **Soundness Theorem** (traditional): Let $\Delta$ be a set of formulas in the propositional caclulus, let $\delta$ be an individual formula from this same calculus, and let $\vdash$ refer here specifically to the proof calculus for this calculus given in Slate. Then:
>
> $$\text{If } \Delta \vdash \delta \text{ then } \Delta \models \delta$$
>
> **Soundness Theorem** (Slate-based): Let $\Delta$ be a set of formulas in the propositional caclulus, let $\delta$ be an individual formula from this same calculus. And,

> let $\rho$ be a finite but otherwise arbitrary proof in Slate, in which every justification (from among the 10 relevant rules) to a conclusion $\psi$, based on a set of assumptions $\Phi$ listed in the vertex containing $\psi$, is certified by a red ✓. Then:
>
> No justification, when switched to $\models$ will result in a red ✗.

To prove that the vertices in our proofs have this property, we will use a proof-by-cases technique to over each of the inference rules that are connected to vertices holding premises "above," and connected to vertices holding conclusions "below." Each vertex in a proof is justified by an inference rule and some (possibly empty) set of premises. Our basis case is to show that each formula in each vertex is justified by an inference rule, and no premise is a consequence of its in-scope assumptions. Our inductive step is to demonstrate that for each rule with a non-empty set of premises, if the formula constituting each premise is a consequence of its in-scope assumptions, then the formula constituting the conclusion of the rule is a consequence of its in-scope assumptions; this is our inductive step. We should only accept this inductive argument if we can be sure that moving 'backwards' to the premises of a vertex, and then to their premises, and so on will always terminate. To ensure this, there must be no infinite chains of inferences in our proofs, nor any cycles. Fortunately, our earlier definition of proof guarantees these properties.

#### 7.2.1.1 Basis Case

Above, we spoke very generally, and said that our basis case would be to check whether the formula of the conclusion of every inference rule with no premises is a consequence of its in-scope assumptions. However, this is a simple task as there is only one inference rule that has no premises: **Assume**. Every vertex justified by **Assume** is of the form:

$$\boxed{\begin{array}{c}\phi\ \{\phi\}\\ \textbf{Assume}\end{array}} \tag{7.1}$$

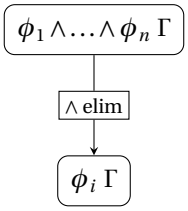
Regardless of what particular sentence $\phi$ is, $\phi$ is a consequence of itself, that is $\{\phi\} \models \phi$, and so our basis case is complete.

#### 7.2.1.2 Inductive Step

The basis case was straightforward, as there is only one rule with no premises. The inductive step has a much larger number of cases, as we have introduced a fairly large set of ten rules, two for each of five boolean connectives. We will consider a few rules to give the general idea behind the cases, but will leave a full proof to the ambitious reader.

**7.2.1.2.1 Conjunction Elimination** Recall the conjunction elimination rule, by which from a conjunction we may infer any one of its conjuncts under the same set

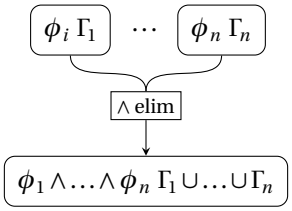
of in-scope assumptions.




(7.2)

By the inductive hypothesis, the premise is sound, so $\Gamma \models \phi_1 \wedge \ldots \wedge \phi_n$. This means that every truth-assignment which satisfies every formula in $\Gamma$ also satisfies $\phi_1 \wedge \ldots \wedge \phi_n$. A truth-assignment satisfies a conjunction if and only if it satisfies each conjunct of the conjunction. Since every truth assignment that satisfies $\Gamma$ satisfies $\phi_1 \wedge \ldots \wedge \phi_n$, it must also satisfy $\phi_i$. Then every truth assignment satisfying $\Gamma$ satisfies $\phi_i$; and this means that $\phi_i$ is a consequence of $\Gamma$; that is, $\Gamma \models \phi_i$. Hence conjunction elimination is sound.

**7.2.1.2.2 Conjunction Introduction** The rule conjunction introduction allows us to infer the conjunction $\phi_1 \wedge \ldots \wedge \phi_n$ from the premises $\phi_1, \ldots, \phi_n$. The conclusion, $\phi_1 \wedge \ldots \wedge \phi_n$, has the in-scope assumptions $\Gamma_1 \cup \ldots \cup \Gamma_n$, where each $\Gamma_i$ is the set of the respective set of in-scope assumptions of $\phi_i$.



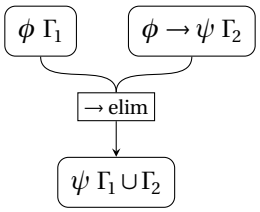

(7.3)

(Note that while we make the in-scope assumptions of the premises and conclusion explicit here, they follow the default pattern that the conclusion's in-scope assumptions are the union of the premises'.) By the inductive hypothesis, each $\phi_i$ is a consequence of $\Gamma_i$. It remains to be shown that $\phi_1 \wedge \ldots \wedge \phi_n$ is a consequence of $\Gamma 1 n$. In pursuit of this, suppose a truth-assignment satisfies each formula in $\Gamma_1 \cup \ldots \cup \Gamma_n$. Since $\Gamma_i \subseteq \Gamma_1 \cup \ldots \cup \Gamma_n$, the truth-assignment also satisfies each formula in $\Gamma_i$, and, since $\Gamma_i \models \phi_i$, we have that $\Gamma_1 \cup \ldots \cup \Gamma_n \models \phi_i$ for each $\phi_i$. The truth-assignment, therefore, satisfies each $\phi_i$, and so also satisfies $\phi_1 \wedge \ldots \wedge \phi_n$. Since the truth-assignment is arbitrary, we obtain $\Gamma_1 \cup \ldots \cup \Gamma_n \models \phi_1 \wedge \ldots \wedge \phi_n$. Therefore conjunction introduction is sound.

We continue by showing the soundness of two more inference rules, conditional elimination and conditional introduction. Conditional elimination is another rule with standard in-scope assumption properties (i.e., the in-scope assumptions of the conclusion are just the union of those of the premises), but conditional introduction requires the removal of an in-scope assumption. With these examples, the arguments for the other rules should not be too hard to derive. The details will differ, of course, for rules that have assumption-discharging and conditional-citing variants. The assumption-discharging versions will require arguments similar to

the one we will give for conditional introduction (which discharges an assumption), but the conditional-citing versions will have arguments similar to those for the rules we have already treated and that for conditional elimination, which we are about to give.

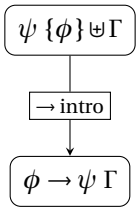
**7.2.1.2.3 Conditional Elimination** Conditional elimination, or *modus ponens*, as we first encountered it, allows us to infer the consequent of a conditional from the conditional and its antecedent.




(7.4)

We reason as follows. By the inductive hypothesis, $\Gamma_1 \models \phi$ and $\Gamma_2 \models \phi \rightarrow \psi$. Suppose a truth-assignment models $\Gamma_1 \cup \Gamma_2$. Then it must satisfy every sentence in $\Gamma_1$, as well as every formula in $\Gamma_2$. Since the truth-assignment satisfies $\Gamma_1$, and because $\Gamma_1 \models \phi$, it must satisfy $\phi$. The truth-assignment also satisfies $\Gamma_2$, and so, since $\Gamma_2 \models \phi \rightarrow \psi$, it must either satisfy $\psi$ or not satisfy $\phi$. It does, however, satisfy $\phi$, and so must satisfy $\psi$. Hence every truth-assignment satisfying $\Gamma_1 \cup \Gamma_2$ must also satisfy $\psi$, so $\Gamma_1 \cup \Gamma_2 \models \psi$, and so conditional elimination is sound.

**7.2.1.2.4 Conditional Introduction** Conditional introduction is the first rule we added to our proof system that had the feature of discharging assumptions. If we use only the conditional-citing versions of the rules that have assumption-discharging and conditional-citing variants, conditional introduction is the only rule in our proof system that discharges assumptions. We must pay careful attention to the in-scope assumptions while we demonstrate the soundness of conditional introduction.




(7.5)

By the inductive hypothesis, $\{\phi\} \uplus \Gamma \models \psi$. If we knew that $\Gamma \models \psi$, we could justify the conclusion by supposing that a truth-assignment satisfied $\Gamma$, and then reasoning that since it must satisfy $\psi$, it would also satisfy $\phi \rightarrow \psi$; and thus we would have $\Gamma \models \psi$. In general, however, we will not know that $\Gamma \models \psi$, and indeed, in many cases we will have $\Gamma \not\models \psi$. How, then, may we proceed?

Suppose a truth-assignment satisfies $\Gamma$. We do not know whether the truth-assignment satisfies $\phi$, but we do know that it either satisfies $\phi$ or that it does not satisfy $\phi$. We consider these cases in turn. If the truth-assignment satisfies $\phi$, then

it satisfies every sentence in $\{\phi\} \uplus \Gamma$, and by the inductive hypothesis must satisfy $\psi$. Since $\phi \rightarrow \psi$ is satisfied by any truth-assignment that satisfies $\psi$ or does not satisfy $\phi$, our truth-assignment satisfies $\phi \rightarrow \psi$. Alternatively, if the truth-assignment does not satisfy $\phi$, then it must satisfy $\phi \rightarrow \psi$ (as, again, $\phi \rightarrow \psi$ is satisfied by any truth assignment that satisfies $\psi$ or does not satisfy $\phi$). Then in either case the truth-assignment that satisfies $\Gamma$ also satisfies $\phi \rightarrow \psi$. Since the truth-assignment is arbitrary, we conclude that $\Gamma \models \phi \rightarrow \psi$. v

**7.2.1.2.5 Finishing the Inductive Proof** The basis case for the soundness of **Assume**, and the portion of the inductive step that we have worked out, should together give a clear idea of how to complete the proof. Once the soundness of the other six inference rules is demonstrated, we may conclude that for any vertex of any proof, that vertex has the property that its formula is a consequence of its in-scope assumptions.

Notice how the justification for the soundness of the various inference rules parallels the reasoning they are intended to capture. For instance, conjunction elimination is designed to capture the pattern of reasoning "if '*A* and *B* and *C*' is true, then each of '*A*', '*B*', and '*C*' must be true;" and the argument for the rule's soundness is, roughly, "if '*A* and *B* and *C*' is satisfied by a truth-assignment, then each of *A* and *B* and *C* is satisfied by that truth assignment." The justification of the soundness of each inference rule for certain types of sentences corresponds directly to using the pattern of reasoning that the rule captures with respect to the semantic structures relevant to those types of sentences. Try working out the soundness proofs for negation introduction or elimination and for disjunction elimination. These proofs depend, respectively, on proof by contradiction and proof by cases, the very patterns of reasoning that they are designed to capture!

## 7.2.2 Completeness of the Propositional Calculus

The theorem in this case, recall, is:

> **Completeness Theorem** (for the propositional calculus): Let $\Delta$ be a set of formulas in the propositional caclulus, let $\delta$ be an individual formula from this same calculus, and let $\vdash$ refer here specifically to the proof calculus for this calculus given in Slate. Then:
>
> $$\text{If } \Delta \models \delta \text{ then } \Delta \vdash \delta$$

There are a number of ways of understanding what this important theorem says. A fairly close-to-the-formal-version is this reading: "If a proposition $\delta$ must be true under the assumption that all members of a set $\Delta$ of propositions are true, then there is a proof of $\delta$ from $\Delta$. But we can be more economical in our reading of the Completeness Theorem. To do this, we can note, first, that if a formula $\phi$ is provable from a set $\Phi$ of formulae, that is, if

$$\Phi \vdash \phi,$$

then the folowing holds:

$$\bigwedge \Phi \rightarrow \phi .$$

In a second move, we simply modify the Completeness Theorem a bit to express it like this:

$$\text{If } \models \bigwedge \Delta \rightarrow \delta \text{ then } \vdash \bigwedge \Delta \rightarrow \delta$$

In order to prove this theorem, we need a rather easy proposition, and a rather deep lemma. We now immediately announce and prove the proposition; following on that, we prove completeness, under the assumption that the lemma holds. We then wrap up by tackling and taming the lemma itself.

> **Proposition 1**: Where $\Phi$ is set of formulas in the propositional caclulus, and $\psi$ is an individual formula from this same calculus, this holds:
>
> $$\text{If } \Phi \nvdash \psi \text{ then Con } \Phi \cup \{\neg\psi\}.$$
>
> **Proof**: Suppose that the hypothesis of the proposition holds ($= \Phi \nvdash \psi$), and further assume, for *reductio*, that Inc $\Phi \cup \{\neg\psi\}$. Given this context, we can easily show in Slate (Figure 7.1 conveys an abstract proof-sketch for the proof) that $\psi$ can be derived from $\Phi$, which gives us the desired contradiction. **QED**

As promised in the plan laid down above, here is the proof of completeness [where the aforementioned 'deep lemma' is (**)]:

> **Proof**: We target the contrapositive of the theorem; viz.,
>
> $$\text{If } \Delta \nvdash \delta \text{ then } \Delta \nvDash \delta$$
>
> Accordingly suppose $\Delta \nvdash \delta$. We obtain by Proposition 1
>
> $$\text{Con } \Delta \cup \{\neg\delta\}$$
>
> Given
>
> (**) Every consistent set of (prop.calc.) formulae is satisfiable.
>
> we have that there is a t.v.a. $\tau$ such that $\tau \models \Delta \cup \{\neg\delta\}$. But this $\tau$ establishes the consequent of our target. **QED**

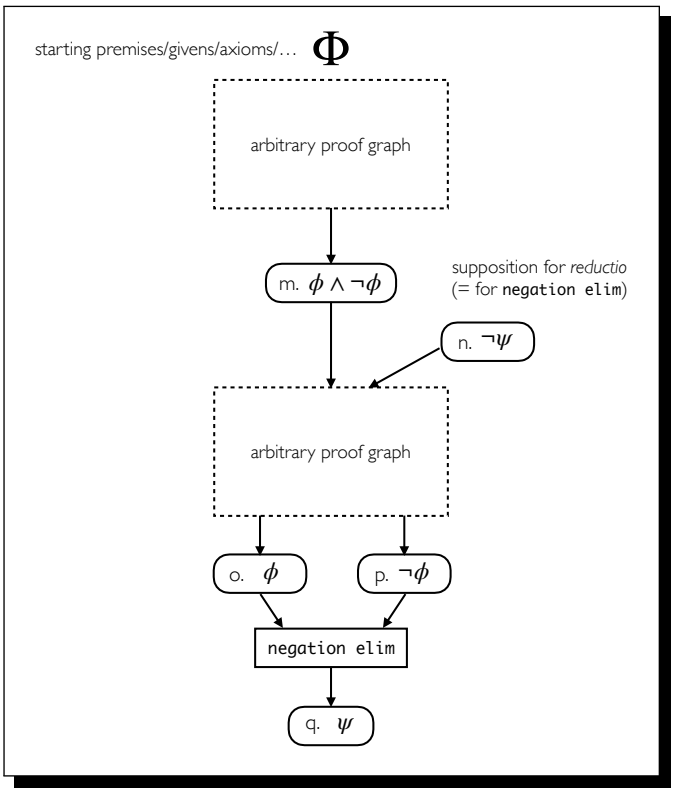
Per the plan we are following, this proof marks nice progress, but we still have before us the a loose end: viz., establishing (**). At present, this task is left as a challenge for the reader.

## 7.3 Meta-Properties of First-Order Logic

### 7.3.1 Completeness of First-Order Logic

The completeness of first-order logic was first established by Gödel, in his dissertation of 1929; this is the first of a long line of remarkable achievements in formal

Figure 7.1: Proof-Sketch for Proving Arbitrary Formula $\psi$ in a Slate Workspace Under Supposition for *Reductio*




logic over the course of a career that, at least in logic, is almost certainly peerless..[1] The classical expression of the completeness theorem for FOL is this statement:

**CCT** For every wff $\phi$ of FOL: If $\models \phi$, then $\vdash \phi$.

**CCT** can be read this way: Any formaula of FOL that's valid (i.e., true on all interpretations) can be proved. Often the completeness theorem is put in such a way that it makes reference to *sets* of formulae, as in for instance:

**CCT′** For every set $\Phi$ of wffs of FOL, and every particular $\phi$: If $\Phi \models \phi$, then $\Phi \vdash \phi$.

Obviously, **CCT** and **CCT′** amount to the same thing, since $\Phi \models \phi$ is equivalent to $\models \bigwedge \Phi \rightarrow \phi$ (where $\bigwedge \Phi$ is the conjunction of the formulae in the set $\Phi$), and any proof of a particular formula from a set of formulae can be easily extended to a

[1] The next major accomplishment by Gödel, reached the following year, was an *in*completeness result, his first incompleteness theorem: viz., in short, that any axiom system which includes basic arithmetic will be plagued by the existence of propositions $\gamma$ such that $\Phi \nvdash \gamma$ and $\Phi \nvdash \neg\gamma$! This is the most widely known result Gödel ever attained, about which many a book has been written. We cover Gödel's first and second incompleteness theorems later in the present chapter.

proof of the conditional that if the conjunction of the formulae in the set hold, so does the particular formula.

Actually, in his dissertation of 1929, Gödel didn't prove **CCT** or **CCT′**; instead, he showed that every formula of FOL is either satisfiable or refutable, which can be stated in this manner:

**GCT** For every $\phi$, either Sat $\phi$ or Ref $\phi$.

Here, Ref $\phi$ is an abbreviation for '$\phi$ is refutable,' which amounts formally to the statement that the negation of $\phi$ can be proved; so, to be explicit, **GCT** relies on this biconditional:

$$\text{Ref } \phi \text{ if and only if } \vdash \neg\phi.$$

# Chapter 8

# Heterogeneous Logic

Heterogeneous logic is a branch of logic in which reasoning can take place over not only symbolic expressions (as e.g. in the formulae we have now seen and worked with repeatedly in previous chapters), but over **pictorial** information as well.[1] In this brief chapter we confine our attention to only one kind of bearer of pictorial information: *diagrams*. We draw directly from the inaugural description of the pre-existing Vivid system for heterogenous reasoning, a description given in (Arkoudas & Bringsjord 2009), but adapt and then employ (in rather free-wheeling fashion) elements of Vivid to fit both the LAMA paradigm (encapsulated, recall, in the Preface) under which Slate falls, and Slate's hypergraphical nature as well.[2]

## 8.1 Symbolic Expressions versus Diagrams

Arithmetic supplies us with expressions that are quintessentially symbolic in nature, not pictorial (and therefore not diagrammatic). Consider for example the expression

E1: $3n = 12$,

the kind of thing that students see in about the 6th grade when first exposed to elementary algebra. Here, $n$ of course refers to some natural number such that, when multiplied by 3, yields 12. Now consider a second expression:

E2: $3 \cdot n = 12$,

E1 and E2 have exactly the same meaning, as we're sure you'll agree; put another way, both equations represent the same underlying assertion that there is at least

[1] Other authors sometimes use different adjectives to refer to the kind of representations in question. E.g., Sloman (1971) speaks of **analogical** representations, and Barwise & Etchemendy (1995) speak of **homomorphic** representations.

[2] A robust implementation of Vivid in Python, in an extended form, has been achieved by former RPI student Nick Marton (2016); we refer to this system as 'pyVivid.' If you are computationally adventurous, and are on top of pretty much all that has come earlier in the present textbook, feel free to experiment with pyVivid, available from the RAIR Lab, on its web page for the Lab's computational reasoners. If you have any trouble finding/acquiring pyVivid, contact S Bringsjord.

one natural number $n$ such that multiplying $n$ and 3 yields 12.[3] Likewise, here's a third equation that represents the same underlying assertion yet again:

E3: $3 \times n = 12$,

The fact that this trio of equations, despite significant differnces between them in the shape of the inscriptions used, is a dead giveaway that these equations are *symbolic* representations. The point is made even clearer if we shift out of base 10 to, say, base 2. For observe that, where we are using binary rather than decimal numbers, the equation

E4: $11 \times n = 1100$

represents precisely the same thing as is represented by E1–E3, and this despite the fact that the shape of the inscriptions used in E4 are dramatically different than those in the trio E1–E3.

Symbolic representations, in short, bear no direct resemblance to the thing they represesent. This phenomenon can of course be seen in any domain, not just arithmetic. Hence,

$$_{\text{IBM}}\,\text{IBM}\;\text{IBM}\;\text{IBM}\;\text{IBM}$$

each denote exactly the same thing: an American technology company famous (most recently) for engineering the *Jeopardy!*-skilled AI system known as Watson. In addition, 'Big Blue' is another way to refer to this company.[4]

Pictorial representations, in stark contrast, are such that very minor differences at the inscription level can have enormous representational differences. Two maps, for instance, or two portraits of soneone's face, differing only at the inscription level in tiny ways, can nonetheless represent entirely different things. Both portraits may be of Smith, but if portrait two, over and above portait one, has but a slight bit of red shading in the whites of Smith's eyes, and the slightest downturn of one corner of Smith's mouth, portait one may represent Smith in an emotionally neutral state, while portrait two may represent a sad Smith. To make the point more vivid, let's turn to one of our llamas: Larry. To represent the fact that at some time $t$ Larry is happy, one symbolic representation we might use is this: $\mathsf{Happy}(\mathsf{larry}, \mathsf{t})$. Here, there is no direct connection between the inscription `Happy` and what is being denoted. We might for example just as easily and reasonably use '`H`' to represent this state, and the shift from `Happy` to `H` wouldn't in the least compromise the validity or accuracy of the representation (just as long as it was known that `H` is stipulated to denote the state of being happy. The shape of `Happy` and `H` are different, but they denote the same thing (viz. the state of being happy), just as the shape of 3 (in base 10) versus that of 11 (in base 2) are different, yet they too both denote precisely the same thing (viz. the fourth natural number starting from and counting up from zero).

In the case of portraits of Larry, the situation is radically different, because potraits are pictorial, not symbolic. To make the distinction concrete, Figure 8.1

[3] Given what we have learned to this point in LAMA-BDLA, in particular about FOL, there is thus an "invisible" existential quantifier in equations E1, E2, and — to refer to a third equation to be soon set out in the main body of the present chapter — E3. E.g., E1 is really, when fleshed out, this: $\exists n(n \in \mathbb{N} \wedge (3n = 12))$.

[4] 'Big Blue' is a longstanding nickname that picks out the company.

Figure 8.1: Portrait of a Happy Larry at Time $t$ (by KB Foushée)

represents Larry's being happy at time $t$, and Figure 8.2 his being angry at a subsequent time $t'$. Here, obviously, differences at the inscription level cause the pair of representations to differ radically.

Figure 8.2: Portrait of an Angry Larry at Time $t'$ ($t < t'$) (by KB Foushée)

Applying heterogeneous logic to faces and elements thereof would be interesting, but we should start instead with something simpler. Accordingly, we turn to a class of puzzles that involve very straightforward diagrams.

## 8.2 Jumping In With Seating Puzzles

Specifically, we now get started in earnest with a class of diagrams that denote seating arrangements. Here's a simple diagram intended to denote five chairs in a row, each of which is either empty or filled with one of four possible people:

$$\underline{?} \quad \underline{?} \quad \underline{?} \quad \underline{?} \quad \underline{?}$$

To make it easier on ourselves, we label the seats from left to right, starting with 1:

$$\begin{array}{ccccc} \underline{?} & \underline{?} & \underline{?} & \underline{?} & \underline{?} \\ s1 & s2 & s3 & s4 & s5 \end{array}$$

In addition, assume that the only people who can occupy seats are Alvin ($a$), Billy ($b$), Cindy ($c$), and Dora ($d$). A seat can also be empty; we denote this condition by the constant $e$. (Here, obviously, these lower-case Roman letters are symbolic

representations of the four people, and emptiness is for convenience treated as a fifth "person.") Given this setup, we have a significant portion of what we need to solve in a Slate-style heterogenous logic so-called **seating puzzles**. Here's such a puzzle given by Barwise & Etchemendy (2008), which we adapt slightly for expository purposes:

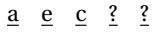
> Our four people are to be seated in a row of five seats. This seating arrangement must satisfy the following three conditions:
>
> C1 $a$ and $c$ should flank the empty seat.
>
> C2 $c$ should be closer to the middle seat than $b$.
>
> C3 $b$ and $d$ should be seated next to each other.
>
> Given this information, reach the following three goals:
>
> G1 prove that the empty seat can't be in the middle and can't be on either end;
>
> G2 settle whether it can be determined who must be seated in the middle seat;
>
> G3 settle whether it can be determined who is to be seated on the two ends.

As you may have noticed by now, This is fundamentally not a difficult problem. You can no doubt easily discover the following informal proof:

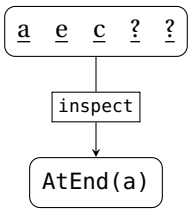
> **Proof**: From all the permutations in which our four characters are seated (with a remaining empty seat), which totals 5! = 120), we can eliminate as inconsistent with condition C1 all but these six possibilities:
>
> P1 a e c ? ?
> P2 ? a e c ?
> P3 ? ? a e c
> P4 c e a ? ?
> P5 ? c e a ?
> P6 ? ? c e a
>
> But possibilities P2–P5 each lead to absurdity when combined with the conjunction of C2 and C3. Hence by `disjunctive syllogism` over the disjunction of all six possibilities P1–P6 we deduce P1 $\vee$ P6, from which we in turn can deduce (i) G1, (ii) that $c$ is in the middle seat and hence an affirmative answer settles G2, and (iii) that any of $a$, $b$, $d$ can occupy an end seat, and from this a negative answer to the query presented in G3. **QED**

What would a corresponding *formal* proof in the Slate style be like? Our first provision for enabling such formal proofs is simply to make a distinction between formulae, which as the reader well knows have hitherto been designated at the meta-language level by lower-case Greek letters $\phi, \psi$ and so on, and diagrams, which for convenience we denote at this level by $\mathfrak{d}$ and slight variants (e.g. $\mathfrak{d}'$ and $\mathfrak{d}_1$). Our second move is to add to our deductive arsenal a new inference rule that allows us to deduce symbolic information $\phi$ that is directly expressed in a diagram $\mathfrak{d}$. The new inference rule is: `inspection`. This rule allows us to extract a piece of information

expressed by a given diagram, and to cast that information in the form of a standard, symbolic formula. For an example, note that each of P1 through P6 in the informal proof given above is a diagram. We set P$i$ to $\mathfrak{d}_i$. Now consider specifically $\mathfrak{d}_1$, that is:




We can deduce by `inspect`[5] that `AtEnd(a)`. We can also deduce by `inspect` that `At(a,s1)`. Here's the first of these inferences in Slate style:




(8.1)

For a formal proof of the solution to the seating puzzle above we need inferential machinery beyond the rule of `inspect`. In the informal proof of the solution given above, the first move is to zero in on the six cases P1–P6. Each of these is now understood to be a diagram. While the informal proof seeks to rule out cases, heterogeneous logic in Slate provides a better route, one directly inspired by Vivid. The route is that if in every single diagrammatic case arising from a starting diagram, conjoined with a key formula $\chi$, we are able to obtain some goal, and the cases are exhaustive, then the goal can be deduced, with the remaining assumption of $\chi$. The inference schema in question is `cases diag2diag`. Of course, certain givens may be needed — and in fact in the case before us, needed givens include the particular formula

$\sigma$ $\forall x \forall y(\texttt{SameSeat}(x,y) \rightarrow x = y)$

which expresses the needed background constraint that distinct people must if seated occupy distinct seats. For added convenience, we label the designated goal as follows:

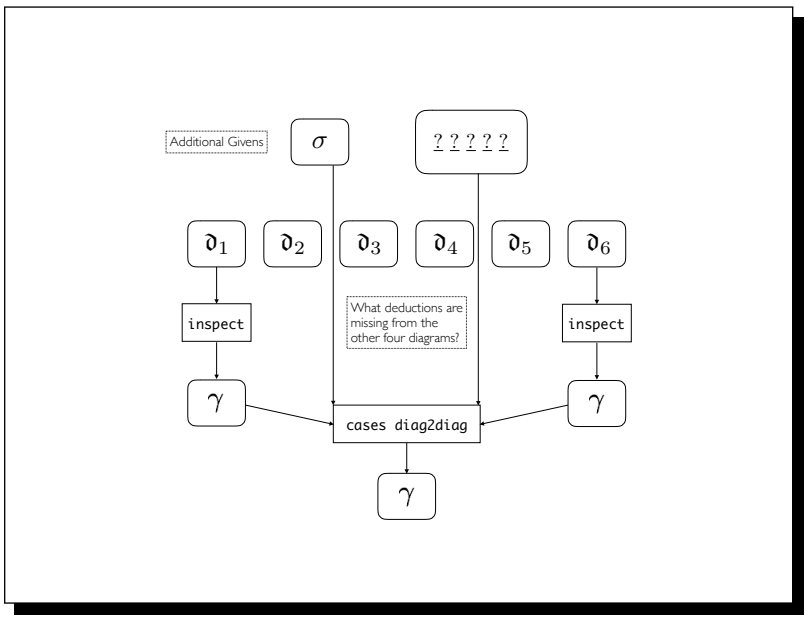
$\gamma$ $\neg\texttt{AtEnd}(e) \wedge \neg\texttt{Middle}(e) \wedge \texttt{Middle}(c)$

A Slate-style proof that makes use of `cases diag2diag` is shown in Figure 8.3.

Now let's return to portraits. We introduce with help from them another inference schema that is part of the available machinery in Slate for heterogeneous deduction: `thinning`. Consider the diagram ($\mathfrak{d}_?$) shown in Figure 8.4. As you can see, Larry's emotional state is a mystery. But under some circumstances, it's possible to deduce, from the diagram shown in this figure, in conjuction with helpful symbolic information, what emotional state Larry is in. Let's assume, for example, that the emotional state of Larry the Llama must be, at any one time, exactly one of five possible states: *being angry*, *being happy*, *being distrustful*, *being sad*, *being fearful*.

[5] This is the informal, "Slatean" correlate to the inference rule **observe** in (Arkoudas & Bringsjord 2009).

Figure 8.3: Hetergeneous Slate Proof that Solves the Seating Puzzle




Moreover, let's assume that corresponding to each of these states is a symbolic relation symbol (resp.,

$$\mathsf{Angry}, \mathsf{Happy}, \mathsf{Distrustful}, \mathsf{Sad}, \mathsf{Fearful}),$$

and that these symbols, along with an obvious and naïve use of constant symbols to denote Larry and the particular time in question, can be used to build formulae such as:

$$\mathsf{Happy}(\mathsf{larry}, \mathsf{t}) \quad \mathsf{Angry}(\mathsf{larry}, \mathsf{t}).$$

We have one such formula for each of our five possible emotional states.

Figure 8.4: Portrait of an "Emotionally Mysterious" Larry (diagram $\mathfrak{d}_?$; by KB Foushée)


Now, suppose that the following conjunction, $\phi$, holds:

$$\neg\mathsf{Happy}(\mathsf{larry}, \mathsf{t}) \wedge \neg\mathsf{Angry}(\mathsf{larry}, \mathsf{t}) \wedge \neg\mathsf{Distrustful}(\mathsf{larry}, \mathsf{t}) \wedge \neg\mathsf{Fearful}(\mathsf{larry}, \mathsf{t})$$

Given this conjunction, and the diagram $\mathfrak{d}_?$, we can clearly reduce (or "thin") the ways in which this diagram can be fleshed out. There are only give possible emotional states Larry can be in, and we know by $\phi$ that four of these possibilities, or cases, can be ruled out as inconsistent with $\phi$. Since only one case remains, and it's that Larry is sad, we can refine $\mathfrak{d}_?$ into the only remaining diagrammatic possibility that remains: that Larry is sad. This diagram is shown in Figure 8.5.

Figure 8.5: Result of Deduction by `thinning` Applied to Diagram $\mathfrak{d}_?$ and $\phi$ (by KB Foushée)

## 8.3 Exercises

1. Examine Figure 8.3 carefully, if you haven't done so already. Complete the proof, by specifying and deploying the missing givens, and by adding the deductions that are missing below each of the diagrams $\mathfrak{d}_2$–$\mathfrak{d}_4$.
2. What, exactly, is the general form of `cases diag2diag`? Write down the inference schema in graphical form. (We suggest that you generalize from the proof produced by an answer to the previous exercise.)
3. What, exactly, is the general form of `thinning`? Write down the inference schema in graphical form.
4. As you know, we have (informally) introduced the inference schema `thinning`. What do you think the inference schema called `widening` is, given that it is accurately said to be the inverse of `thinning`? Write down a specification of it, as your best hypothesis.

# Chapter 9

# Inductive Logic

## 9.1 Introduction

To this point, the present book has provided introductory coverage of formal *deductive* logic. This focus is of course conveyed by this phrase in its title: 'Beginning Deductive Logic.' From this title you have probably deduced, or at least been tempted to deduce, that therefore there must be more to logic than deductive logic; specifically, there must be a *non*-deductive form of logic. Indeed there is. Much of logic pertains to reasoning that isn't deductive, but which is still quite rigorous, and quite important; that is, much of logic is, to use the traditional label, **inductive** logic. While the hallmark of deductive logic is *proof*, the hallmark of inductive logic is the concept of an *argument*. An exceptionally strong kind of argument is a proof, but plenty of arguments fall short of being proofs — and yet still have considerable force.

We can turn to a certain longstanding and popular class of "card-logic puzzles" to elucidate our departure at this point from deductive logic to inductive logic.[1] Let's use a streamlined way to refer to playing cards found in the standard 52-card deck used for games like Poker, Bridge, and so on. For instance, to refer to the card known as 'Ace of Spades' we use


and for annother example, here's how we represent Two of Hearts:


Now, here's a little card-logic puzzle for you:

[1] It's standard practice to count puzzles that are in the realm of inductive logic, as opposed to deductive logic, as "logic puzzles" nonetheless. E.g., see (Danesi 2017), Chapter 1 of which, entitled "Card Logic Puzzles," provides a very nice set of puzzles of the sort that we now proceed to introduce. Of course, we demand a formal justification that the answers given *are* answers. Following standard practice in the popular literature, this Danesi (2017) doesn't do.

> Two rows of playing cards are laid out on a table in front of you, as shown immediately below. Which card is missing from the top row (= Row 1)? Prove that your answer is correct.
>
> Row 1: K♡ 5♣ 9♢ ?
>
> Row 2: Q♠ 9♢ 5♣ K♡

In this genre of logic puzzles, the answer, which presumably you had little trouble coming up with, is:

Q♠.

But of course you're not done yet, are you? There was a hitch: you were asked to prove that your answer is correct. Can you accomplish that? No — at least you can't unless you posit as assumptions that certain background propositions hold for this puzzle. It should be pretty obvious what these propositions would need to be. For example, for a proof that the Queen of Spades is the correct answer, you could start by assuming that (*) for every card $c_{R2}$ in Row 2 there must be a matching (in number/royalty and suit) card $c_{R1}$ in Row 1. Then, since Queen of Spades in Row 2 must match some card in Row 1, and it doesn't match any of the three shown in Row 1, Row 1 must have Queen of Spades. No doubt you see that with a bit of effort, an outright proof could be articulated. In fact it makes for a nice exercise to prove, in Slate, after suitable formulae with suitable relations are devised, that the answer holds.

Buf of course such a proof relies upon (*), and this statement isn't supplied in the puzzle; you invoked it. The invocation, whatever else it be, certainly isn't deduction. To hammer this home, note that there are any number of key propositions that would enable a proof that the answer is Queen of Spades. For instance, let's assign a location number to each entry in a row, starting from 1 and progressing to 4, left to right. (Hence e.g. location 2 in Row 2 is occupied by Ten of Diamonds.) This addition enables us to speak unambiguously of the reversal of the cards in a row. And, in turn, we can now posit this statement:

> (**) Row 1 is the reversal of Row 2.

Given (**), we can produce another proof that the correct answer is Queen of Spades. But of course we're in the same boat as before: Where does (**) come from? We don't quite know, but we *do* know that (**) doesn't come from deduction. For if it did, where is the declarative content from which it's deduced? The upshot is that we *hypothesize* statements like (*) and (**) in order to justify the answers that are given. To put the situation another way: Card-logic puzzles require puzzlers to cook up an implicit argument in support of some proposition which, when added to background knowledge that all puzzler are assumed to have,[2] allows the deduction of the correct answer.

We would be remiss if we didn't add here that card-logic puzzles standardly include a list of options, from which the one correct answer is to be selected. Such

[2] For instance, one such piece of knowledge is that there are no duplicate cards in the standard 52-card deck.

a list generally makes it easier to deduce, given the hypothesized principle(s), the answer. One reason is that if all but one option can be excluded by deduction, the remaining option must be the — to use the technical term used by test designers — **key**. In this case, we have a type of card-logic puzzle that *is* a matter of pure deduction.

Let's examine a second card-logic puzzle in order to firm things up a bit, and make sure we understand the situation. See Figure 9.1. The question here is of course: What playing card should go where the large question-mark appears? We'll give you a moment to think about it ...

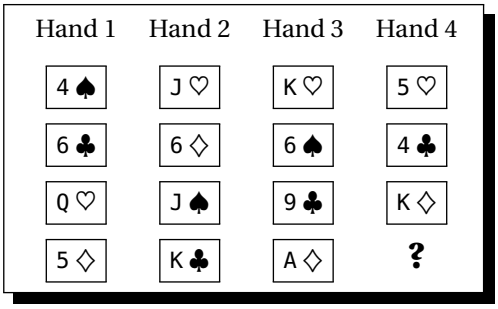



Figure 9.1: Card-Logic Puzzle #2

Hopefully you're not too angry with us. We asked you a trick question. We did so in the hopes that you would conclude that there is no one card that is the solution to the problem, as it stands. Clearly, there's no way to deduce what the mystery card is.[3] Now watch how a set of options can immediately change the situation. Suppose that the four options for what the missing card is are given in Figure 9.3; the correct answer must be one of these. Now, what is the mystery card, and can you prove that you're right? Look carefully ...

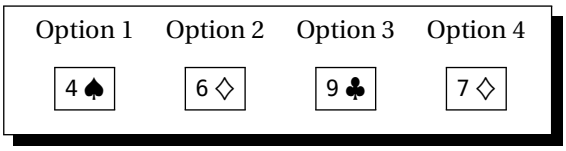



Figure 9.2: First Set of Options for Card-Logic Puzzle # 2

Absolutely! You're right! The correct answer is Option 4: They mystery card is 7 of Diamonds. Can you prove it? Easily:

> **Proof**: These hands are dealt from a standard 52-card deck, which entails that no two cards in the four hands match. We have four cases to consider (Option

[3] How would you prove that it can't be proved from the information supplied in Figure 9.1, along with basic background knowledge about standard card decks, but not other information, that a particular card is the mystery card? This is a challenging question, but one that will catalyze lots of learning!

1–Option 4). Options 1, 2, and 3, if assumed, lead to a contradiction. Ergo, Option 4 is correct. **QED**

Hence, we have here the special case of a card-logic puzzle that is deductive in nature, not inductive. But let's now associate with Card-Logic Puzzle #2 not the options shown in Figure 9.3, but instead the options shown in Figure **??**. Now what's the mystery card?

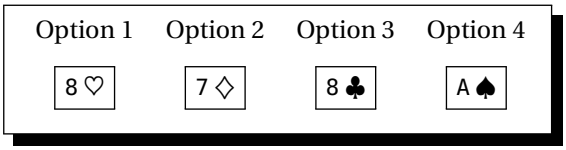



Figure 9.3: Second Set of Options for Card-Logic Puzzle # 2

The shift to the second option set has instantly thrown us out of pure deductive logic into the realm of inductive logic. Specifically, we have returned to the type of card-logic puzzle with which we began. In this type of puzzle, a solution requires three steps: generating an hypothesis $\mathscr{H}$ that should guide the search the mystery card; realizing that given the observed facts about the particular cards in view, in conjunction with $\mathscr{H}$ and background knowledge about standard card decks, one can prove the mystery card is $c^*$; and — this being a step that card-logic puzzles in the popular media leave aside — supplying the proof in question.

Turning from inductive playing-card puzzles to something more conventionally academic, consider the following argument $\alpha_1$:

| | | |
|---|---|---|
| | (1) | Tweety is bird. |
| | (2) | Most birds can fly. |
| ∴ | (3) | Tweety can fly. |

For start contrast, consider as well this argument ($\alpha_2$):

| | | |
|---|---|---|
| | (1′) | 3 is a positive integer. |
| | (2′) | All positive integers are greater than 0. |
| ∴ | (3′) | 3 is greater than 0. |

The second of these arguments qualifies as an outright proof. That is, using the notation much employed before the present chapter:

$$\{(1'),(2')\} \vdash (3')$$

But in stark contrast, argument $\alpha_1$ is not a proof that Tweety can fly. The reason is obvious: (3) isn't deduced from the combination of (1) and (2); that is,

$$\{(1),(2)\} \nvdash (3)$$

This says that there is no proof of (3) from (1) and (2) taken as assumptions/suppositions or premises/givens. Yet argument $\alpha_1$ does have considerable force. Put another way, there is certainly a non-zero "degree" to which statements (1) and (2) confirm

that Tweety can fly. In fact, intuitively, the degree of confirmation here isn't merely non-zero: at the very least, it's appreciable. Put yet another way: All things being equal, it wouldn't be irrational to bet, on the basis of (1) of and (2), that Tweety can fly (i.e. that (3) holds).

We shall speak of the degree $\delta$ of confirmation that an argument $\alpha$ provides for a conclusion $\chi$, given premises $\Pi$. We write this as:

$$\Pi \mid\!\!\vdash^{\delta}_{\alpha} \chi$$

Applied to the case of Tweety, we have the following:

$$\{(1), (2)\} \mid\!\!\vdash^{\delta_1}_{\alpha_1} (3)$$

In the case of argument $\alpha_2$ about positive integers, we have:

$$\{(1'), (2')\} \mid\!\!\vdash^{\delta_2}_{\alpha_2} (3')$$

In seems undeniable that $\delta_2$ must be greater than $\delta_1$; hence any inductive logic should reflect this fact, or at least be consistent with it. Of course, at this point, we have no rigorous way to determine what value $\delta_1$ takes on in the case of argument $\alpha_1$ about Tweety. On the other hand, perhaps you will agree with us that in the case of argument $\alpha_2$, the degree to which the combination of premises (1′) and (2′) confirm (3′) is maximally strong. Soon, we shall remedy, at least to a significant degree, the vagueness of the concept of degree of confirmation. But before we do, let's step back and consider the fundamentals of an inductive logic.

## 9.2 Requirements for an Inductive Logic

What do we need in order to have a "contender" inductive logic? Some things are, if you will, non-negotiable. For example, a logic, whether deductive or non-deductive, must be anchored by some alphabet and grammar that together precisely define the formulas that represent statements in some more informal language — for example, statements in English that we have just considered above in arguments $\alpha_1$ and $\alpha_2$. But let's leave such lower-level constituents aside for now; in due time we'll take them up. What *general* requirements must be satisfied by any inductive logic? Taking account of the long history of inductive logic, Fitelson (2005) lays down three desiderata that most modern inductive logicians have agreed must be satisfied in order for a reasoning system to qualify as an inductive logic:

> D1 An inductive logic should include deductive entailment and refutation as endpoints on a continuum that includes degrees of inductive entailment/refutation somewhere between these endpoints. Put in terms of the two examples visited above, arguments $\alpha_1$ and $\alpha_2$, deductive entailment, seen in $\alpha_2$, should lead to a degree of confirmation that is maximally high. Deductive refutation of a conclusion $\chi$ on the strength of premises $\Pi$ would corrrespond to $\Pi \vdash \neg\chi$.

D2 An inductive logic should use formal, classical probability theory as a key building block. In a modification that departs from Fitelson, we relax this requirement and stipulate only that an inductive logic should use measures of uncertainty that can be applied to statements, and possibly to inferential links as well.

D3 An inductive logic should be, to use the adjectives of (Fitelson 2005), "objective" and "formal." The idea here is that degree of confirmation of a conclusion $\chi$ provided by premises $\Pi$ shouldn't be merely a matter of subjective, personal preference. In the case of argument $\alpha_1$ about Tweety, a rash gambler might be willing to stake all his savings on a bet that Tweety is a bird, whereas an excessively conservative gambler might refuse to put down even a penny on this bet, complaining that it's not a sure thing that Tweety can fly. An inductive logic should presumably provide some objective machinery independent of both such minds that renders a verdict as to the degree to which, in light of premises (1) and (2), (3) is confirmed.

### 9.2.1 Naïve Inductive Logic (NIL)

NIL Where $\Pi \Vdash^{\delta}_{\alpha} \chi$, $\delta$ is high iff $prob(\bigwedge \Pi \rightarrow \chi)$ is high.

### 9.2.2 Received Inductive Logic (RIL)

RIL Where $\Pi \Vdash^{\delta}_{\alpha} \chi$, $\delta = prob(\chi \mid \bigwedge \Pi)$.

Here, conditional probability is defined in a way that follows Kolmogorov:

$$prob(\phi \mid \psi) = \frac{prob(\phi \wedge \psi)}{prob(\psi)}$$

What are logical interpretations of RIL? The bottom line is that conditional probability, $prob(\phi \mid \psi)$, must be defined in terms of the abstract structure or shape of $\phi$ and $\psi$, where these two objections are formulae in some abstract language.

## 9.3 Formal, Classical Probability Theory

Students studying beginning deductive logic often develop the impression that such notions as probability are utterly outside the realm of formal logic. Nothing could be further from the truth. The *opposite* is in fact the case: probability theory is *based* on formal deductive logic (as we shall see momentarily). The erroneous impression is due in majority measure not to the unreliable whims of novice students of logic, but rather to the distressing fact that the traditional teaching of introductory deductive logic is fundamentally deficient. If students are exposed only to systems in which every single formula is ultimately either true or false, with concepts such as *probable* and *unlikely* left aside, the ironic fact is that it is simply *logical* for such students to assume that such bivalence, bereft as it is of any semantic shades of grey, is the invariant landscape in which the logician moves.

Kolmogorov (1933) gave us the axioms of probability rather a long time ago. Where the underlying formal language is that of the propositional calculus (with which, at this point you are quite familiar), and where we use

$$\text{Inc}\ \Phi$$

to indicate that the set $\Phi$ (of formulae) is inconsistent,[4] they are the following four:[5]

**KPC**$_{\text{PC}}$

K1 $\forall \phi (0 \leq prob(\phi) \leq 1)$

K2 $\forall \phi$ if $\vdash \phi$, then $prob(\phi) = 1$.

K3 $\forall \phi, \psi$ if $\{\phi\} \vdash \psi$, then $prob(\phi) \leq prob(\psi)$.

K4 $\forall \phi, \psi$ if Inc $\{\phi, \psi\}$, then $p(\phi \vee \psi) = p(\phi) + p(\psi)$.

These axioms will reward those who think hard about them; they are remarkably fertile. You should ask yourself in each case why clever people would think that what is expressed is unshakably true. To pursue that question, you will need to create and reflect upon some examples.

While deep and sustained reflection upon Kolmogorov's quartet will indeed pay great dividends, it requires very little thought to see that this quartet is shot through with deductive logic. For example, K2 tells us that when we have as a theorem that $\phi$, we know that $\phi$ has a probability of 1: it is certain. K3 tells us that if $\psi$ can be deduced from $\phi$, then the probability of $\phi$ is less than, or equal to, $\psi$. And K4 informs us that one can deduce a contradiction from a set of formulae in the propositional calculus,

## 9.4 Blending Probability Theory with the Propositional Calculus

But these connections between Kolmogorov's probability calculus and deductive logic seem abstract. After all, we have seen no proofs in Slate that involved anything like **KPC**$_{\text{PC}}$.[6] Well, we haven't seen such things *yet*. But we will see them now, after making a series of moves to integrate the propositional calculus as we have used it in Slate, with the Kolmogorovian quartet of K1–K4. The end result of this series, quite a pleasant and promising one, is that we will obtain Slate workspaces that implement an "under one roof" integration of the propositional calculus with **KPC**$_{\text{PC}}$.

[4] Remember: Proof-theoretically this means that from $\Phi$ some contradiction $\phi \wedge \neg\phi$ can be deduced; model-theoretically this means that $\Phi$ is unsatisfiable. We are assuming a proof-theoretic interpretation of K4 here.

[5] We here closely follow that presenttion of Adams (1998). Note that we refer not to *events*, which is customary, but to formulae instead.

[6] We have selected the acronym '**KPC**$_{\text{PC}}$' to be mnemonic, as it denotes '**K**olmogorov's **P**robability **C**alculus (at the level of the Propositional Calculus)'. (Probability calculi pitched at the level of first-order logic are not only possible, but certainly eventually required (e.g., to formally model human probabilistic reasoning.) For coverage, see the appendix 'A7 Probabilistic Predicate Logic' in (Adams 1998).

Our first move is to harken back to the axiomatization of the propositional calculus given in section 2.7.1, and put it in view now. As we have seen, that axiomatization, **PM**, can be implemented in Slate, and if you've been energetic you have even proved some theorems on your own in **PM**.

Our second move is to see how we can take the axioms of **PM** and encode them as formulae in FOL. This move opens up new territory; make sure your thinking cap is firmly on. Perhaps an example is the best way to see how our encoding is going to work. Here's one of the axioms in **PM**:

A1 $(\phi \vee \phi) \rightarrow \phi$

## 9.5 Some Simple Inductive Logics in the LAMA Style

### 9.5.1 The Simple Inductive Logic $\mathscr{L}^{\text{IND}}$

The logic $\mathscr{L}^{\text{IND}}$ is based on a formal language that augments the syntax of FOL via three straightforward moves: In the first move, every well-formed formula $\phi$ from FOL can be paired with with one of eight integers that encode strength factors

$$S = \{4, 3, 2, 1, 0, -1, -2, -3, -4\}$$

to form a well-formed formula $\phi^n, n \in S$; that is, we write the pairings by simply putting a superscipt on a wff $\phi$ from FOL. Hence we can for instance write $\phi^4$. But what do these strength factors communicate? The nine integers constitute a continuum ranging from `certain`, 4, to `certainly false`, -4. Table 9.1 and Table 9.2) together provide an English label for each strength factor in the continuum. While there is no formal defintion of each label, they have been chosen to convey a general sense of the level of strength (or lack thereof, in the below-zero part of the continuum) to be captured by the integer in question. In addition, we provide some elaboration now.

Table 9.1: First Part of Strength-Factor Continuum

| 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| `certain` | `evident` | `beyond reasonable doubt` | `probable` | counter-balanced |

Table 9.2: First Part of Strength-Factor Continuum

| -1 | -2 | -3 | -4 |
|---|---|---|---|
| `improbable` | `beyond reasonable belief` | `evidently false` | `certainly false` |

Propositions that are `certain` include those expressed by axioms in established systems of logic and mathematics. Basic number theory provides an informative example. Readers will recall that in Chapter 6 we canvassed a number of theories of arithmetic. One of these theories is **Q**, and one of the axioms (Axiom 1) of **Q** is:

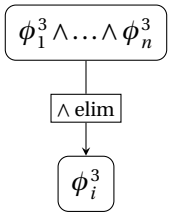
$$\forall x (0 \neq s(x))$$

where as you will recall the function symbol $s$ represents a function that maps a given $k \in \mathbb{N}$ to $k+1$. You should feel quite comfortable pairing the strength factor 4 with Axiom 1; in other words, you should feel quite comfortable embracing the claim that every natural number $k$ produced by adding 1 to another number is such that $k$ can't be 0. So, in the notation of $\mathscr{L}^{\text{IND}}$, we have

$$\text{Axiom } 1^4.$$

Given this, the other end of the continuum seems easy to understand. Propositions that are `certainly false` would for instance include the negation of any `certain` proposition. We thus know that anyone making the following claim is certainly wrong: there is a natural number $k$ such that, if you add 1 to $k$, the result is 0.

The second move is to explain how the familiar rules of inference in FOL, when adjusted to include strength factors, work. In the case of the rules with which we're already familiar, we stipulate that if each of the formulae in each of the premises has a strength of $n \in S$, then the conclusion deduced by one of these rules has a strength of $n$ as well. For example, suppose we know that $\phi^3 \wedge \psi^3$. Then we know that by `conjunction elimination` we can derive $\phi^3$ on its own, and can by the same rule derive $\psi^3$ on its own.




(9.1)

# Bibliography


Adams, E. (1998), *A Primer of Probability Logic*, CSLI, Stanford, CA.

Andréka, H., Madarász, J. X. & Németi, I. (2007), Logic of Space-Time and Relativity Theory, *in* M. Aiello, I. Pratt-Hartmann & J. V. Benthem, eds, 'Handbook of Spatial Logics', Springer, pp. 607–711.

Andréka, H., Madarász, J. X., Németi, I. & Székely, G. (2011), 'A Logic Road from Special Relativity to General Relativity', *Synthese* pp. 1–17.
**URL:** *http://dx.doi.org/10.1007/s11229-011-9914-8*

Andrews, P. (2002), *An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof*, Springer, New York, NY.

Arkoudas, K. & Bringsjord, S. (2007), 'Computers, Justification, and Mathematical Knowledge', *Minds and Machines* **17**(2), 185–202.
**URL:** *http://kryten.mm.rpi.edu/ka_sb_proofs_offprint.pdf*

Arkoudas, K. & Bringsjord, S. (2009), 'Vivid: An AI Framework for Heterogeneous Problem Solving', *Artificial Intelligence* **173**(15), 1367–1405.
**URL:** *http://kryten.mm.rpi.edu/KA_SB_Vivid_offprint_AIJ.pdf*

Baker, N. (2013), 'Wrong Answer: The Case Against Algebra II', *Harper's Magazine* pp. 31–38.
**URL:** *https://harpers.org/archive/2013/09/wrong-answer*

Barwise, J. & Etchemendy, J. (1995), Heterogeneous Logic, *in* J. Glasgow, N. Narayanan & B. Chandrasekaran, eds, 'Diagrammatic Reasoning: Cognitive and Computational Perspectives', MIT Press, Cambridge, MA, pp. 211–234.

Barwise, J. & Etchemendy, J. (2008), Information, infons, and inference, *in* 'Situation Theory and Its Applications', CSLI, Stanford, CA, pp. 33–78.

Bass, L. & Johnson, A. (2012), *Geometry: Common Core*, Pearson, Upper Saddle River, NJ. Series Authors: Charles, R., Kennedy, D. & Hall, B. Consulting Authors: Murphy, S. & Wiggins, G.

Bellman, A., Bragg, S. & Handlin, W. (2012), *Algebra 2: Common Core*, Pearson, Upper Saddle River, NJ. Series Authors: Charles, R., Kennedy, D. & Hall, B. Consulting Authors: Murphy, S. & Wiggins, G.

Blackburn, P. & Bos, J. (2005), *Representation and Inference for Natural Language: A First Course in Computational Semantics*, CSLI, Stanford, CA.

Boole, G. (2010), *An Investigation of the Laws of Thought on which are Founded the Mathematical Theories of Logic and Probabilities*, Kessenger, Whitefish, MT. This book was originally published in 1854.

Boolos, G. S., Burgess, J. P. & Jeffrey, R. C. (2003), *Computability and Logic (Fourth Edition)*, Cambridge University Press, Cambridge, UK.

Bringsjord, E. (2001), Computerized-adaptive Versus Paper-and-Pencil Testing Eenvironments: An Experimental Analysis of Examinee Experience, PhD thesis, University at Albany, Albany, NY.

Bringsjord, E. & Bringsjord, S. (2014), Education and ... Big Data Versus Big-But-Buried Data, *in* J. Lane, ed., 'Building a Smarter University', SUNY Press, Albany, NY, pp. 57–89. This url goes to a preprint only.
**URL:** *http://kryten.mm.rpi.edu/SB_EB_BBBD_0201141900NY.pdf*

Bringsjord, E. & Bringsjord, S. (n.d.), Exploring the Relationship between Logic Background and on Performance on the Analytical Section of the GRE. This paper is derived from and supported by the doctoral dissertation of the first author: (Bringsjord, E. 2001).
**URL:** *http://kryten.mm.rpi.edu/bringsjordx2-1.logic.gre1.doc*

Bringsjord, S. (1992), CINEWRITE: An Algorithm-Sketch for Writing Novels Cinematically, and Two Mysteries Therein, *in* M. Sharples, ed., 'Computers and Writing: State of the Art', Kluwer, Dordrecht, The Netherlands.

Bringsjord, S. (1995), Pourquoi hendrik ibsen est-il une menace pour la littérature générée par ordinateur?, *in* A. Vuillemin, ed., 'Littérature et Informatique la Littérature Générée Par Orinateur', Artois Presses Universite, Arras, France, pp. 135–144.

Bringsjord, S. (2015*a*), A 21st-Century Ethical Hierarchy for Humans and Robots: $\mathscr{EH}$, *in* I. Ferreira, J. Sequeira, M. Tokhi, E. Kadar & G. Virk, eds, 'A World With Robots: International Conference on Robot Ethics (ICRE 2015)', Springer, Berlin, Germany, pp. 47–61. This paper was published in the compilation of ICRE 2015 papers, distributed at the location of ICRE 2015, where the paper was presented: Lisbon, Portugal. The URL given here goes to the preprint of the paper, which is shorter than the full Springer version.
**URL:** *http://kryten.mm.rpi.edu/SBringsjord_ethical_hierarchy_0909152200NY.pdf*

Bringsjord, S. (2015*b*), 'A Vindication of Program Verification', *History and Philosophy of Logic* **36**(3), 262–277. This url goes to a preprint.
**URL:** *http://kryten.mm.rpi.edu/SB_progver_selfref_driver_final2_060215.pdf*

Bringsjord, S., Bringsjord, E. & Noel, R. (1998), In Defense of Logical Minds, *in* 'Proceedings of the $20^{th}$ Annual Conference of the Cognitive Science Society', Lawrence Erlbaum, Mahwah, NJ, pp. 173–178.

Bringsjord, S., Clark, M. & Taylor, J. (2010), Honestly Speaking, How Close are We to HAL 9000?, *in* H. Guerra, ed., 'Physics and Computation 2010, 3rd International Workshop', Luxor/Aswan, Egypt, pp. 39–53. Workshop held August 30–September 6, 2010. The draft available via the url given here is subject to change carried out by Bringsjord.
**URL:** *http://kryten.mm.rpi.edu/sb_mc_hal_091510.pdf*

Bringsjord, S., Govindarajulu, N., Banerjee, S. & Hummel, J. (2018), Do Machine-Learning Machines Learn?, *in* V. Müller, ed., 'Philosophy and Theory of Artificial Intelligence 2017', Springer SAPERE, Berlin, Germany, pp. 136–157. This book is Vol. 44 in the book series. The paper answers the question that is its title with a resounding No. A preprint of the paper can be found via the URL given here.
**URL:** *http://kryten.mm.rpi.edu/SB_NSG_SB_JH_DoMachine-LearningMachinesLearn_preprint.pdf*

Bringsjord, S. & Govindarajulu, N. S. (2018), Artificial Intelligence, *in* E. Zalta, ed., 'The Stanford Encyclopedia of Philosophy'.
**URL:** *https://plato.stanford.edu/entries/artificial-intelligence*

Bringsjord, S. & Yang, Y. (2003), 'Problems Used in Psychology of Reasoning are Too Easy, Given What our Economy Demands', *Behavioral and Brain Sciences* **26**(4), 528–530.

Chisholm, R. (1963), 'Contrary-to-Duty Imperatives and Deontic Logic', *Analysis* **24**, 33–36.

Chisholm, R. (1982), Supererogation and Offence: A Conceptual Scheme for Ethics, *in* R. Chisholm, ed., 'Brentano and Meinong Studies', Humanities Press, Atlantic Highlands, NJ, pp. 98–113.

Cohen, P. (1963), 'The Independence of the Continuum Hypothesis I', *Proceedings of the U.S. National Academy of Sciences* **50**, 1143–1148.

Danesi, M. (2017), *The Everything Logic Puzzles Book*, Adams Media, Avon, MA.

Darwin, C. (1997), *The Descent of Man*, Prometheus, Amherst, NY. A reprint edition. The book was first published in 1871.

Ebbinghaus, H. D., Flum, J. & Thomas, W. (1994), *Mathematical Logic (second edition)*, Springer-Verlag, New York, NY.

Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A., Lally, A., Murdock, W., Nyberg, E., Prager, J., Schlaefer, N. & Welty, C. (2010), 'Building Watson: An Overview of the DeepQA Project', *AI Magazine* pp. 59–79.
**URL:** *http://www.stanford.edu/class/cs124/AIMagzine-DeepQA.pdf*

Fitelson, B. (2005), Inductive logic, *in* J. Pfeifer & S. Sarkar, eds, 'Philosophy of Science: An Encyclopedia', Routledge, London, UK, pp. 384–394.

Frege, G. (1967), Letter to Russell, *in* J. van Heijenoort, ed., 'From Frege to Gödel: A Sourcebook in Mathematical Logic', Harvard University Press, Cambridge, MA, pp. 126–127.

Frege, G. (n.d.), *Grundgesetze der Arithmetik, Year = 1893*, Verlag Hermann Pohle, Jena, Germany.

Genesereth, M. & Nilsson, N. (1987), *Logical Foundations of Artificial Intelligence*, Morgan Kaufmann, Los Altos, CA.

Glymour, C. (1992), *Thinking Things Through*, MIT Press, Cambridge, MA.

Goldberg, A. & Suppes, P. (1972), 'A Computer-Assisted Instruction Program for Exercises on Finding Axioms', *Educational Studies in Mathematics* **4**, 429–449.

Govindarajulu, N. & Bringsjord, S. (2016), Crowdsourcing Theorem Proving via Natural Games, *in* U. Furbach & C. Schon, eds, 'Proceedings of the Second Workshop on Bridging the Gap between Human and Automated Reasoning', International Joint Conference on Artificial Intelligence, New York, NY. This paper revolves around the game Catabot Rescue, by Govindarajulu. The URL here goes to the full *Proceedings*, in which the this specific paper appears.
**URL:** *http://ratiolog.uni-koblenz.de/proceedings2016.pdf*

Govindarajulu, N. S. (2013), Uncomputable Games: Games for Crowdsourcing Formal Reasoning, PhD thesis, Rensselaer Polytechnic Institute (RPI), Troy, NY.

Govindarajulu, N. S. & Bringsjord, S. (2015), Ethical Regulation of Robots Must be Embedded in Their Operating Systems, *in* R. Trappl, ed., 'A Construction Manual for Robots' Ethical Systems: Requirements, Methods, Implementations', Springer, Basel, Switzerland, pp. 85–100.
**URL:** *http://kryten.mm.rpi.edu/NSG_SB_Ethical_Reg_at_OS_Level_offprint.pdf*

Inhelder, B. & Piaget, J. (1958), *The Growth of Logical Thinking from Childhood to Adolescence*, Basic Books, New York, NY.

Johnson-Laird, P. (1997), 'Rules and Illusions: A Criticial Study of Rips's *The Psychology of Proof*', *Minds and Machines* **7**(3), 387–407.

Johnson-Laird, P. N. (1983), *Mental Models*, Harvard University Press, Cambridge, MA.

Johnson-Laird, P. & Savary, F. (1995), How to Make the Impossible Seem Probable, *in* M. Gaskell & W. Marslen-Wilson, eds, 'Proceedings of the 17th Annual Conference of the Cognitive Science Society', Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 381–384.

Kahneman, D. & Frederick, S. (2002), Representativeness Revisited: Attribute Substitution in Intuitive Judgment, *in* T. Gilovich, D. Griffin & D. Kahneman, eds, 'Heuristics and Biases: The Psychology of Intuitive Judgment', Cambridge University Press, New York, NY, pp. 49–81.

Khemlani, S., Byrne, R. & Johnson-Laird, P. (2018), 'Facts and Possibilities: A Model-Based Theory of Sentential Reasoning', *Cognitive Science* **42**, 1887–1924.

Kolmogorov, A. (1933), *Grundbegriffe der Wahrscheinlichkeitrechnung*, Ergebnisse Der Mathematik. Translated as *Foundations of Probability*, New York, NY: Chelsea Publishing Company, 1950.

Kolmogorov, A. & Uspenskii, V. (1958), 'On the Definition of an Algorithm', *Uspekhi Matematicheskikh Nauk* **13**(4), 3–28.

Lenzen, W. (2004), Leibniz's Logic, *in* D. Gabbay, J. Woods & A. Kanamori, eds, 'Handbook of the History of Logic', Elsevier, Amsterdam, The Netherlands, pp. 1–83.

Lewis, C. I. (1960), *A Survey of Symbolic Logic: The Classic Algebra of Logic*, Dover, New York, NY. This book is a reprint of the 1918 version, and omits Chapters 5 and 6 from that original version.

Lewis, H. & Papadimitriou, C. (1981), *Elements of the Theory of Computation*, Prentice Hall, Englewood Cliffs, NJ.

Luger, G. (2008), *Artificial Intelligence: Structures and Strategies for Complex Problem Solving (6th Edition)*, Pearson, London, UK.

Marton, N. (2016), PyVivid: A Concrete Framework for Mechanized Diagrammatic Reasoning, PhD thesis, Rensselaer Polytechnic Institute (RPI). This is an MS thesis, not a PhD thesis.

McCullough, D. (2005), *1776*, Simon & Shuster, New York, NY.

McKeon, R., ed. (1941), *The Basic Works of Aristotle*, Random House, New York, NY.

Mukherjee, D. (2009), A Formal and Neuroscientific Investigation of the Distinction Between Normatively Correct and Incorrect Human Reasoning, PhD thesis, Rensselaer Polytechnic Institute; Troy, NY.

Partee, B., Meulen, A. & Wall, R. (1990), *Mathematical Methods in Linguistics*, Kluwer, Dordrecht, The Netherlands.

Penn, D., Holyoak, K. & Povinelli, D. (2008), 'Darwin's Mistake: Explaining the Discontinuity Between Human and Nonhuman Minds', *Behavioral and Brain Sciences* **31**, 109–178.

Post, E. (1943), 'Formal Reductions of the General Combinatorial Decision Problem', *American Journal of Mathematics* **65**(2), 197–215.

Rinella, K., Bringsjord, S. & Yang, Y. (2001), Efficacious Logic Instruction: People are not Irremediably Poor Deductive Reasoners, *in* J. D. Moore & K. Stenning, eds, 'Proceedings of the Twenty-Third Annual Conference of the Cognitive Science Society', Lawrence Erlbaum Associates, Mahwah, NJ, pp. 851–856.

Rips, L. (1989), 'The Psychology of Knights and Knaves', *Cognition* **31**(2), 85–116.

Rips, L. (1994), *The Psychology of Proof*, MIT Press, Cambridge, MA.

Ross, J. (1992), 'Immaterial Aspects of Thought', *The Journal of Philosophy* **89**(3), 136–150.

Russell, B. (1967), Letter to Frege, *in* J. van Heijenoort, ed., 'From Frege to Gödel: A Sourcebook in Mathematical Logic', Harvard University Press, Cambridge, MA, pp. 124–125.

Russell, S. & Norvig, P. (2009), *Artificial Intelligence: A Modern Approach*, Prentice Hall, Upper Saddle River, NJ. Third edition.

Sloman, A. (1971), 'Interactions Between Philosophy and AI: The role of Intuition and Non-logical Reasoning in Intelligence', *Artificial Intelligence* **2**, 209–225.

Smith, P. (2007), *An Introduction to Gödel's Theorems*, Cambridge University Press, Cambridge, UK.

Smith, R. (2017), Aristotle's Logic, *in* E. Zalta, ed., 'The Stanford Encyclopedia of Philosophy'.
**URL:** *https://plato.stanford.edu/entries/aristotle-logic*

Sosa, E. (1999), Are Humans Rational?, *in* K. Korta, E. Sosa & X. Arrazola, eds, 'Cognition, Agency and Rationality', Kluwer, Dordrecht, The Netherlands, pp. 1–8. This book is the Proceedings of the Fifth International Colloquium on Cognitive Science.

Stanovich, K., West, R. & Tolpak, M. (2016), *The Rationality Quotient: Toward a Test of Rational Thinking*, MIT Press, Cambridge, MA.

Tarski, A. (1995), *Introduction to Logic and to the Methodology of Deductive Sciences*, Dover Publications, New York, NY.

Turing, A. M. (1937), 'On Computable Numbers with Applications to the *Entscheidungsproblem*', *Proceedings of the London Mathematical Society* **42**, 230–265.