Decidable fragments of First-Order Logic

Ian Pratt-Hartmann
School of Computer Science
University of Manchester Manchester M13 9PL, UK

Contents

Chapter 1

Introduction

1.1 The *Entscheidungsproblem*

In 1928, D. Hilbert and W. Ackermann published their seminal work *Grundzüge der theoretischen Logik* [38], in which they asked if there exists an algorithm to determine whether any given sentence of first-order logic is *valid*—i.e., true in every structure. Dually: is there an algorithm to determine whether a given sentence of first-order logic is *satisfiable*—i.e. true in some structure? Since a sentence is valid just in case its negation is not satisfiable, any algorithm for determining the validity of a first-order sentence yields one for determining satisfiability, and conversely. Thus we have just one question here; Hilbert and Ackermann called it the *Entscheidungsproblem* ("decision problem").

It was known at the time that the question has a positive answer when restricted to sentences lying in certain restricted subsets of first-order logic. Thus, for example, in 1915, L. Löwenheim had considered the *monadic fragment* of first-order logic, in which only 1-place predicates (and no function-symbols) appear. Löwenheim showed that any such formula, if satisfiable, is satisfiable over a domain of bounded size, a matter which can then be checked by brute-force. A corresponding observation was made by P. Bernays and M. Schönfinkel regarding sentences of the form $\exists x_1 \cdots \exists x_m \forall y_1 \cdots \forall y_n . \psi$, with ψ containing no quantifiers, function-symbols or the equality sign [14]. More impressively, the satisfiability problem for sentences of the form $\exists x_1 \cdots \exists x_m \forall y \exists z_1 \cdots \exists z_n . \psi$ (i.e. one universal quantifier) was shown to be algorithmically solvable by W. Ackermann [1]. And a few years after the first edition of the *Grundzüge*, this result was extended to sentences of the form $\exists x_1 \cdots \exists x_m \forall y_1 \forall y_2 \exists z_1 \cdots \exists z_n . \psi$ (i.e. two adjacent universal quantifiers), independently, by K. Gödel [27], K. Schütte [69] and L. Kalmár [45]. It is worth remarking that each of these fragments embeds the most venerable logical system of all—the classical syllogistic. Hence the problem of determining the validity of arguments couched in the formal language bequeathed to us by Aristotle is also algorithmically solvable.

It is easy to see what an answer to the *Entscheidungsproblem* might look like assuming the task we have been set is possible—one simply produces the sought-after algorithm, together with an argument for its correctness. Less obvious is what to say if that task is in fact impossible. For to show that *no* algorithm exists to determine the satisfiability of a given first-order formula, one first needs a convincing characterization of what, in general, an algorithm *is*—a development which had to wait until the work of A. Church and, more particularly, A. Turing in 1936-7. Church and Turing independently showed that, under their proposed reconstructions of this notion, there is indeed no algorithm to decide the the satisfiability (or validity) of any given first-order sentence [17, 76]. If such an algorithm existed, then the so-called halting problem for Turing machines would itself be solvable by a Turing machine; and that is simply not possible. As one now says: the satisfiability problem for first-order logic is *undecidable*. Furthermore—and as Turing himself realized—this approach can be used to show the undecidability of satisfiability even for very restricted fragments of first-order logic. Thus, Hilbert and Ackermann's *Entscheidungsproblem* becomes a classification problem: of the various fragments of first-order logic one might consider, which are decidable for satisfiability?

In contexts where logical formulas are used to describe finite ensembles of objects, it is more natural to ask whether a given formula is *finitely satisfiable*—i.e. true in a *finite* structure. For first-order logic at least, this question cannot be reduced to that of determining satisfiability, because it is impossible to write a formula chatacterizing precisely the infinite domains. This observation yields a finitary variant of Hilbert and Ackermann's *Entscheidungsproblem*: given a first-order sentence, determine whether it is finitely satisfiable. And that problem too turns out to be undecidable, as shown in 1950 by B. Trakhtenbrot [75]. For many less expressive fragments—including those of Löwenheim, Bernays-Schönfinkel, Ackermann and Gödel-Schütte-Kalmár mentioned above—satisfiability and finite satisfiability coincide: if a formula in one of these fragments is satisfiable at all, then it is finitely satisfiable. (Such a fragment is said to have the *finite model property*). However, it is easy to identify simple fragments which lack the finite model property; indeed, as with the original *Entscheidungsproblem*, so too with its finitary variant, proofs of undecidability for the whole of first-order logic in fact show undecidability for very restricted fragments. Thus, we have second classification problem here: which fragments of first-order logic are decidable for *finite* satisfiability? This book, in a nutshell, is about these two classification problems. We wish to chart, as W. Quine put it in 1968, the *limits of decision* in first-order logic, both for satisfiability and finite satisfiability.

Actually, we shall set ourselves a more ambitious goal. If a problem is algorithmically solvable, the obvious question to ask is: how quickly? The rise of the theory of computational complexity during the 1970s and 80s yielded a conceptual framework for analysing the intrinsic difficulty of problems within the Turing model of computation, leading to the familiar hierarchy of complexity classes, NLogSpace $\subseteq$ PTime $\subseteq$ NPTime $\subseteq$ PSpace $\subseteq$ ExpTime $\subseteq$ NExpTime $\subseteq \cdots$. These considerations can be applied to the *Entscheidungsproblem*. Consider, for example, the problem of determining whether a finite set of formulas of the classical syllogistic is satisfiable (or, dually, deciding whether a putative argument in this language is valid). One might expect this problem to be computationally easy—and indeed it turns out to be complete for the 'small' complexity class NLogSpace. By contrast, the satisfiability problem for Ackermann's fragment (sentences of the form $\exists x_1 \cdots \exists x_m \forall y \exists z_1 \cdots \exists z_n.\psi$, with ψ quantifier- function- and equality-free) is ExpTime-complete, while satisfiability for the Gödel-Schütte-Kalmár fragment (sentences of the form $\exists x_1 \cdots \exists x_m \forall y_1 \forall y_2 \exists z_1 \cdots \exists z_n.\psi$) is NExpTime-complete. Thus, we have a trade-off between expressive power and complexity of satisfiability: roughly, the more expressive a fragment is, the harder it is to reason in it. We would like to establish the exact terms of this trade. That is, for any given fragment of first-order logic, we ask: are its satisfiability and finite satisfiability problems decidable, and if so, what is their computational complexity?

In attempting to answer these questions, we require a scheme for deciding which of the uncountably many fragments of first-order logic we are to consider. An entirely systematic approach appears unachievable; however, we may identify three categories of decidable fragment that, at least historically, have played a salient role in research into the *Entscheidungsproblem*. Conceptually the most straightforward—though not the first to command the attention of the academic community—is to limit the number of variables. More specifically, we consider the fragment of first-order logic (without function-symbols or equality) in which formulas are restricted to use just *two variables*. It was realized in 1962 by D. Scott [70], that any formula of this fragment can be effectively transformed into a formula of the Gödel-Schütte-Kalmár fragment satisfiable over the same domains. It follows that this *two-variable fragment* has the finite model property and that its satisfiability (= finite satisfiability) problem is decidable. (Later researchers showed that the same applies to the slightly larger two-variable fragment *with* equality.) Two is as far as one can go in this respect: the satisfiability and finite satisfiability problems for the corresponding fragment with three variables are both undecidable. Nevertheless, the two variable fragment forms the basis of many decidable fragments of first-order logic, of which the most important examples are covered in this book.

The second principal category of decidable fragments considered here arises from restricting the sorts of formulas to which quantifiers are allowed to apply. In 1989, H. Andréka, I. Németi and J. van Benthem considered the fragment of first-order logic in which all universal quantification to conforms to the pattern $\forall \bar{x}(A \to \psi)$, where A is an atomic formula featuring all the free variables of ψ, and similarly, all existential quantification conforms to the pattern $\exists \bar{x}(A \wedge \psi)$ with the same conditions on A and ψ. They called the atomic formulas A in this context *guards*, and showed the resulting *guarded* fragment to have a decidability satisfiability problem [3]. This development proved timely. At about this time, interest had been growing in the Computer Science community in the use of formal languages—often under the rubric of *description logics*—to reason about structured data ensembles (such as XML documents or the collections of tables encountered in relational databases), with a view to making various reasoning services (consistency-checking, constraint entailment, query simplification) capable of automation. In such applications, guarded quantification appears naturally; indeed, most description logics can be viewed as one or other of the variants of guarded logic treated in this book.

Our third principal category of decidable fragments—and historically the first—is one we have already encountered, namely that of restricting attention to specific quantifier prefixes. Since every first-order formula can be effectively put into *prenex form* (with all the quantifiers moved to the front), it makes sense to undertake a systematic classification of those quantifier prefixes for which the satisfiability problem is decidable. The Bernays-Schönfinkel, Ackermann and Gödel-Schütte-Kalmár fragments mentioned above, are of course characterized in this way. One important issue that arises in this connection is that of the equality predicate. It is straightforward to show that allowing equality in formulas of the Bernays-Schönfinkel fragment compromises neither the finite model property nor the decidability of satisfiability. The same is true of the Ackermann fragment (though here there is a slight increase in computational complexity from ExpTime to NExpTime). Not so with the Gödel-Schütte-Kalmár fragment, however. It was shown by W. Goldfarb in 1984 that adding equality to this fragment results in a logic whose satisfiability and finite satisfiability problems are (different and) undecidable [28]. The whole subject is surveyed in exquisite detail in E. Börger, Y. Gurevich and E. Grädel's monumental work *The Classical Decision Problem* [15]. That book gives the decidability and computational complexity of (almost) all fragments defined by (i) the allowed quantifier prefixes, (ii) the maximal numbers of predicates and function-symbols (of all arities), and (iii) the presence or absence of the equality predicate. We give a (hopefully accessible) one-chapter treatment here.

It is almost, but not entirely, true that there were no significant developments in logic between Aristotle and Frege. One (at least arguable) exception can be found in the attempts of 19th century logicians to extend the expressive range of the syllogistic—most notably in the works of G. Boole and A. De Morgan. Whereas for the original Academicians, logical statements concerned only general categories of being, their Victorian successors were happy to apply them to specific, concrete situations in which it was perfectly natural to count objects. Viewed this way, arguments such as *Some artists are beekeepers; all beekeepers are carpenters; therefore some artists are carpenters* have obvious numerical generalizations: *At least m artists are beekeepers; at most n beekeepers are not carpenters; therefore at leat $(m\text{-}n)$ artists are carpenters.* (To see that this really is a generalization, put $m = 1$ and $n = 0$.) De Morgan laboured in vain to come up with an adequate numerical generalization of the system of syllogism due to Aristotle; but 20th-century logicians, with access to the more generous syntax of first-order logic, enjoyed greater success. In 1999, L. Pacholski, W. Szwast and L. Tendera, and, independently, E.Grädel. M. Otto and E. Rosen showed that the two-variable fragment could be extended with so-called *counting quantifiers*, without losing the decidability of satisfiability [33, 62]. One exciting feature of the resulting *two-variable fragment with counting* is that, unlike the decidable fragments mentioned above, it lacks the finite model property. (The finite satisfiability problem, incidentally, is also decidable.) Furthermore, just as one can restrict the standard quantifiers to guarded patterns, so one can do the same for counting quantifiers. The resulting *guarded two-variable fragment with counting* is of particular interest in the context of description logics, where numerical constraints also naturally arise. Here, logic makes contact with important developments from a quite different area of computer science, namely, the theory of linear programming, which considers algorithmic solution to systems of linear equations and inequalities. Thus, fragments with counting quantifiers will occupy a central position in this book. We remark in this regard that the logician W. Jevons (a contemporary of Boole and De Morgan, principally remembered today for his work on economics) cleary saw this connection [43], but lacked the algorithmic tools to exploit it.

So far, we have discussed decidable fragments of first-order logic in purely syntactic terms: by specifying which logical resources—numbers of variables, allowed patterns of quantification, available quantifiers—the fragment in question has at its disposal. But a quite different means of constructing logical fragments is available, one that, for all its topicality, has its origins further back in the history of modern logic. Ostensibly, *propositional modal logic* augments Boolean logic with the 1-place sentential operators $\Box$ ("It is necessarily the case that ...") and $\Diamond$ ("It is possibly the case that ..."). Research in this area was revolutionized by the work of S. Kripke and others in the early 1960s, however, with the development of a semantics based on the Leibnizian idea of truth at *possible worlds* [51]. Thus, $\Box\varphi$ ("Necessarily, φ") is taken to be true a world if φ is true in all worlds 'accessible' from that world; dually, $\Diamond\varphi$ ("Possibly, φ") true a world if φ is true at some world 'accessible' from it. Under these semantics, formulas of propositional modal logic translate naturally into first-order formulas (with one free variable) in which proposition letters are re-cast as 1-place predicates (holding or not holding at worlds) and quantification over possible worlds is relativized by means of a 2-place predicate denoting the relation of 'accessibility'. Kripke showed that the axiom system of propositional modal logic K (as it is now known) derives exactly those formulas of propositional modal logic whose first-order translations, in this scheme, are valid. Moreover, the problem of determining validity for such formulas is decidable: indeed, Anréka, Németi and van Benthem explicitly conceived of the guarded fragment as the natural generaliation of first-order translations of modal formulas.

More strikingly, perhaps, various additional axioms that had previously been proffered as logical principles governing the modalities of necessity and possibility turned out to correspond, under these semantics, to natural properties of the accessibility relation: the axiom $\Box\varphi \rightarrow \varphi$ to the *reflexivity* of accessibility, the axiom $\Box\varphi \rightarrow \Box\Box\varphi$ to the *transitivity* of accessibility, and so on. The resulting axiom systems can be shown to derive exactly those formulas (of propositional modal logic) whose first-order translations are valid under the corresponding restrictions on the accessibility relation. Thus, we obtain here not one fragment of first-order logic logic, but a whole family—one for each collection of constraints on the accessibility relation. Again, the satisfiability problems for the most salient such fragments turn out to be decidable: their computational complexity—generally lying between NPTIME-complete and PSPACE-complete—was characterized in 1977 by R. Ladner [52]. Insofar as the semantic constraints in question are first-order expressible, the resulting logics are (in a natural sense) fragments of first-order logic; but their characterization nevertheless has a semantic, rather than syntactic flavour.

This is a move that we can make in the context of other decidable fragments of first-order logic; and it is particularly fertile in the context of the two-variable fragment. Take, for example, the property of transitivity. In the *two-variable fragment* of first-order logic, it is impossible to write a formula characterizing precisely those structures in which a given binary predicate is interpreted as a transitive relation. Similarly, one cannot write a formula characterizing precisely those structures in which a given binary predicate is interpreted as an equivalence relation. This fact suggests a family of variants of the satisfiability problem for the two-variable fragment, obtained by stipulating that some number of distinguished binary predicates must be interpreted as transitive relations, equivalence relations, and so on. Again we ask: for which of the resulting logics are the satisfiability and finite satisfiability problems decidable, and with what complexity? When the semantic restrictions in question are first-order expressible—for example, transitivity or the property of being an equivalence relation—we have a fragment of first-order logic. When the semantic restrictions are not-first-order expressible, we technically stray outside first-order logic; however, we do consider some such logics in this book, where they are amenable to techniques we have already encountered. So it is, for example, with logics in which some distinguished binary predicate is required to be interpreted as the graph of a *tree*.

The aim of this book is to provide a survey of the current state of play in respect of decidable fragments of first-order logic, a century after the appearance of the *Grundzüge*. We cannot hope for a such a clinical dissection of the subject as *The Classical Decision Problem* provided for the quantifier prefix fragments: we must deal with a greater body of work and a more scattered topic. Faced with such terrain, we have no option but to confine ourselves to the commanding heights. The book is divided into four parts. Part I begins with the basic decidable fragments that will serve as a base for future exploration. There is only one good place to begin here: the classical syllogistic. We analyse this fragment—together with its relational counterpart that so baffled pre-Fregean logicians—from a modern complexity-theoretic viewpoint, with results that are anything but routine or predictable. We then move on to give a systematic treatment of the three major categories of decidable fragments mentioned above: the two-variable fragment, the guarded fragment, and the quantifier prefix fragments.

Part II considers logics with counting quantifiers, which, as we mentioned, may be used to extend the two-variable fragment, without losing decidability of satisfiability. A principal point of interest here is the translation of satisfiability problems into linear programming and linear integer programming problems. In contradistinction to the fragments considered in Part I, most of these systems lack the finite model property. One important feature of the numerical methods considered here is that they typically yield satisfiability-checking procedures that can be uniformly adapted to the case of finite satisfiability. (Actually, it is rather the other way round: *finite* satisfiability is the more natural problem here; satisfiability is dealt with by a routine extension.) As mentioned above, much of the motivation for studying counting quantifiers originates in the application of logic to modelling structured data. We therefore close this part of the book with some results on this topic.

Part III then considers logics characterized by semantic restrictions, beginning with a brief survey of the salient results on propositional modal logic and *graded modal logic* (essentially: propositoinal modal logic with counting quantifiers) which originally inspired the study of such fragments. Again, the primary focus is on extensions of the two-variable fragment, in which the salient notions of *transitive relation* and *equivalence relation* are not definable. Accordingly, we investigate what happens to the two-variable fragment and its guarded subfragment (possibly in the presence of counting quantifiers) when it is extended with a fixed number of transitive or equivalence relations, presenting an (almost) complete account of the decidability and comptuational complexity of fragments in question. We end with a chapter on logics in which the two-variable fragment with counting is interpreted over a linear orders and trees.

The foregoing results by no means exhaust the study of decidable fragments, however, and Part IV surveys some of the salient remaining fragments. Here, the selection of topics is, perhaps inevitably, more rhapsodic: we consider the *negation-guarded fragment*, the *unary-guarded fragment* and finally, the *fluted fragment*. By way of retrieving at least a scrap of synthetic unity, we mention that the last of these topics finally gives a full answer to a question raised by Quine in his discussion of the 'limits of decision' mentioned above.

1.2 Logic

We assume familiarity with the syntax of first-order logic and the very basics of model theory. This section gives a brief tour of the notation (mostly standard) and foundational results used in the sequel. It is included here as a convenient reference; we expect many readers to skim over it. All the model theory required (and much more) can be found in standard texts such as those by C. Chang and J. Keisler [16] or W. Hodges [40].

As usual, we employ signatures of individual constants, predicates (of various non-negative arities) and function-symbols (of various positive arities). Predicates of arity 0 will be referred to as *proposition letters*. However, we take function symbols to have positive arity by definition; in this book, individual constants do not count as function symbols. Since we do not require signatures of more than countable cardinality in the sequel, we may assume that all individual constants, predicates and function symbols are chosen from some fixed infinite sets. We further assume that the sets of individual constants, function-symbols and predicates to be disjoint, and indeed that the arities of function-symbols and predicates are unambiguous. That said, we shall in practice use whatever letters are convenient to range over any of these sets: thus, c may at times be an individual constant, and times a unary predicate, and so on, depending on the exigencies of the matter at hand. The equality predicate $=$ is taken to be a logical constant, and does not count as part of the signature. A *sentence* is a formula with no free variables. If φ is any formula, we denote by $\|\varphi\|$ the number of symbols in φ, and if Φ is a finite set of formulas, denote by $\|\Phi\| = \sum\{\|\varphi\| : \varphi \in \Phi\}$ the total number of symbols involved.[1]

We assume familiarity with the standard model-theoretic semantics of first-order logic. In this regard, a structure interpreting a signature $(s_1, \ldots, s_n)$ is a tuple $\mathfrak{A} = (A, s_1^{\mathfrak{A}}, \ldots, s_n^{\mathfrak{A}})$, where A is a non-empty set, called the *domain of quantification*, and $s^{\mathfrak{A}}$ denotes the interpretation of symbol s in $\mathfrak{A}$. In particular: if c is an individual constant, $c^{\mathfrak{A}} \in A$; if f is a k-ary function-symbol, $f^{\mathfrak{A}} : A^k \rightarrow A$; and if p is a k-ary predicate, $p^{\mathfrak{A}} \subseteq A^k$. Structures are denoted by (possibly decorated) fraktur letters $\mathfrak{A}$, $\mathfrak{B}$, ..., and their domains by the corresponding Roman letters. Except when explicitly indicated to the contrary (and only very rarely), structures are assumed to have non-empty domains. If $\mathfrak{A}$ and $\mathfrak{A}'$ are structures interpreting signatures Σ and Σ' over some common domain A, with $\Sigma \subseteq \Sigma'$, then we say that $\mathfrak{A}$ is a *reduct* of $\mathfrak{A}'$ (and $\mathfrak{A}'$ an *expansion* of $\mathfrak{A}$) if $\mathfrak{A}$ is obtained from $\mathfrak{A}'$ by simply forgetting the interpretations of all the symbols in $\Sigma' \setminus \Sigma$.

If φ is a formula, with free variables included in the list $\bar{x} = x_1, \ldots, x_n$, $\mathfrak{A}$ is a structure and $\bar{a} = a_1, \ldots, a_n$ a tuple of elements of A, we write $\mathfrak{A} \models \varphi[\bar{a}]$ to indicate that the assignment $\bar{a} \rightarrow \bar{x}$ *satisfies* $\varphi(\bar{x})$ in $\mathfrak{A}$. We say φ is *satisfiable* if, for some $\mathfrak{A}$ and some $\bar{a} \in A^n$, $\mathfrak{A} \models \varphi[\bar{a}]$; if, in addition, A is finite, we say φ is *finitely satisfiable.* A set of formulas Φ with free variables included in the list $\bar{x}$, is (*finitely*) *satisfiable* if for some $\mathfrak{A}$ and some $\bar{a} \in A^n$, $\mathfrak{A} \models \varphi[\bar{a}]$ for every $\varphi(\bar{x}) \in \Phi$. Clearly, finite satisfiability implies satisfiability, but not, in general conversely. A sentence that is satisfiable but not finitely satisfiable is called an *axiom of infinity.* If φ is a sentence, we write $\mathfrak{A} \models \varphi$ to indicate that φ is *true* in $\mathfrak{A}$—i.e. is satisfied by the empty sequence; in that case, we call $\mathfrak{A}$ a *model* of φ. Similarly for sets of sentences. Thus, a sentence—or set of sentences—is (finitely) satisfiable just in case it has a (finite) model. If φ is a sentence such that $\neg\varphi$ is not satisfiable, we say φ is *valid*, and write $\models \varphi$. If $\mathfrak{A}$ is a structure interpreting some structure Σ, then the set of sentences(over Σ) true in $\mathfrak{A}$ is called the *theory of* $\mathfrak{A}$, and denoted $\mathrm{Th}(\mathfrak{A})$. Two structures $\mathfrak{A}$ and $\mathfrak{B}$ interpreting Σ are said to be *elementarily equivalent*, written $\mathfrak{A} \equiv \mathfrak{B}$, if they make the same sentences true, i.e. if $\mathrm{Th}(\mathfrak{A}) = \mathrm{Th}(\mathfrak{B})$. If $\mathfrak{A}$ is a structure interpreting a signature without function symbols, and $B \subseteq A$ is non-empty, we denote by $\mathfrak{A}{\restriction}B$ the strcuture $\mathfrak{B}$ which results by restricting the interpretations in $\mathfrak{A}$ to the elements of B in the obvious way. We say that $\mathfrak{B}$ is a *sub-structure* of $\mathfrak{A}$, and write $\mathfrak{B} \subseteq \mathfrak{A}$. If, in addition, for every formula $\psi(\bar{x})$ in the relevant signature, and every tuple $\bar{b}$ from B, $\mathfrak{B} \models \psi[\bar{b}]$ implies $\mathfrak{A} \models \psi[\bar{b}]$, we say that

[1] A technical complication arises when considering formulas over large signatures, where the writing of different signature elements over a fixed alphabet might require the use of long index strings. Thus, a fairer measure of the size of a formula φ over a signature Σ is surely $\|\varphi\| \cdot \log(|\Sigma|)$. In practice, however, this makes no difference to any of the complexity-theoretic results we obtain, and therefore in the sequel we simply take each symbol to have size 1, no matter how many of them there are.

$\mathfrak{B}$ is an *elementary sub-structure* of $\mathfrak{A}$, and write $\mathfrak{B} \preceq \mathfrak{A}$. It is immediate that $\mathfrak{B} \preceq \mathfrak{A}$ implies $\mathfrak{A} \equiv \mathfrak{B}$, but the converse is not true.

We identify first-order logic with the set of its formulas, and denote it by $\mathcal{FO}$. (Since we assume signatures to be chosen from some fixed infinite set of symbols, it is legitimate to speak of sets, rather than proper classes, of formulas.) A *fragment* of first-order logic is a simply a subset of $\mathcal{FO}$. Thus, for example, we may consider the *monadic fragment* of first-order logic (with equality), in which no function-symbols appear and all predicates in the signature have arity 1. In general, fragments of first-order logic are usually defined by restricting the allowed syntax, such as, for example, those consisting of prenex formulas with specified quantifier prefixes. However, we also consider fragments in which, as explained above, a specified collection of predicates are constrained to have interpretations satisfying certain properties.

If $\mathcal{L}$ is a fragment of first-order logic, the *satisfiability problem* for $\mathcal{L}$ is the problem

Sat($\mathcal{L}$)
Given: a finite set Φ of $\mathcal{L}$-formulas
Return: Yes if Φ is satisfiable
No otherwise.

Likewise, the *finite satisfiability problem* for $\mathcal{L}$ is the problem

FinSat($\mathcal{L}$)
Given: a finite set Φ of $\mathcal{L}$-formulas
Return: Yes if Φ is finitely satisfiable
No otherwise.

Most, but not all, fragments $\mathcal{L}$ considered in this book are closed under conjunction, in the sense that, of $\varphi, \psi \in \mathcal{L}$ then $\varphi \wedge \psi \in \mathcal{L}$. In this case, we may without loss of generality assume that the input to the problems Sat($\mathcal{L}$) and FinSat($\mathcal{L}$) are single formulas, rather than finite sets of formulas. This is the form in which we shall most commonly encounter these problems. As mentioned above, a fragment is said to have the *finite model property* if all satisfiable finite sets of formulas are finitely satisfiable, or, equivalently, if its satisfiability and finite satisfiability problems coincide.

Although this book is concerned in large part with models and their transformations, it employs only the very simplest results from model theory as standardly conceived. We briefly mention those results here; detailed explanations and proofs can be found in any serious introductory logic text. When considering satisfiability of first-order formulas, it typically suffices to confine attention to structures of low cardinality. This is guaranteed by the *Downward Löwenheim-Skolem-Tarski Theorem*, which we require in only a weak form. In this book, *countable* means “finite or countably infinite.”

Proposition 1.1 (Löwenheim-Skolem Theorem). *Let $\mathfrak{A}$ be a structure interpreting a countable signature Σ. Then there exists a countable, elementary substructure $\mathfrak{B} \preceq \mathfrak{A}$.*

In particular, any satisfiable first-order sentence has a countable model. A further result that we will need occasionally is the so-called *compactness theorem* for first-order logic.

Proposition 1.2 (Compactness of first-order logic)**.** *Let Φ be a set of first-order formulas. Then Φ is satisfiable if and only if every finite subset of Φ is satisfiable.*

There are many ways to prove Proposition 1.2. Perhaps the most relevant for our purposes uses the existence of sound and complete axiomatizations. In 1930, K. Gödel showed that the set of valid sentences of first-order logic can be axiomatically characterized in terms of a finite list of axioms schemas and the inference rules of generalization and *modus ponens* [26]. More specifically, for any set of formulas Φ (not necessarily finite), Φ is satisfiable if and only if there is no derivation of the absurd formula $\perp$ in this system from Φ. If a set of formulas is not satisfiable, then, such a derivation exists, and must use only a finite number of formulas from Φ.

1.3 Computability

We assume familiarity with the Turing model of computation. However, we shall at various points carry out detailed constructions involving Turing machines, and this section establishes the relevant notation and machine architecture. All the material here is standard, and we expect most readers to skim over it. In-depth treatments may be found in standard texts, e.g. that by N. Cutland [19].

A *Turing Machine* is a tuple $M = \langle A, S, s_0, T \rangle$, where A is an finite, non-empty set, S a finite set containing the element s_0, and T a finite set of 'transitions' (explained presently). We call A the *alphabet* of M, and write A^* to denote the set of finite strings over A, including the empty string ϵ. By *symbol*, we mean either an element of A or one of the special characters $\sqcup$ (*blank*) and $\triangleleft$ (*start-of-tape*), which are assumed not to be in A. We call the elements of S *states*, and s_0 the *initial state*. We imagine M to operate on a tape comprising an infinite sequence of squares, with each square having exactly one symbol written on it. A *configuration* of M is a triple consisting of a state (i.e. an element of S), a designated tape square (which we shall think of as the position of a read/write head), and an infinite sequence of symbols (which we take to be written on the tape). Finally, a *transition* is a tuple $\tau = \langle a, s, b, t, \delta \rangle$ where a, b are symbols, s, t are states and $\delta \in \{-1, 0, 1\}$. We think of τ as the instruction "If the head is over a square containing a, and the current state is s, write b on that square, make t the current state, and move the head δ squares to the right." (Thus, $\delta = -1$ means move one square left). We say that τ is *enabled* in a configuration if the current state is s and the head is over a square containing a; in that case, τ can be executed in the obvious way, yielding a subsequent configuration. We assume that no transition changes a square containing $\triangleleft$ or attempts to move left when reading this symbol, and no transition writes the symbols $\triangleleft$ or $\sqcup$ to squares not already containing them. A Turing machine M

is *deterministic* if, for any symbol a and state s, there is at most one transition $\langle a, s, b, t, \delta \rangle$ in T. We sometimes speak of a *non-deterministic* Turing machine, in order to stress that it is not necessarily deterministic; strictly, however, the qualifier "non-deterministic" is redundant.

Let $\mathrm{x} \in A^*$. The *initial* configuration *with input* x is that configuration in which the state is s_0, the head is over the initial (left-most) square, and the tape contents are given by $\lhd \, \mathrm{x} \sqcup^*$, where $\sqcup^*$ is an infinite tail of blanks. A *run* of M *with input* x is a sequence of configurations (finite or infinite) starting with the initial configuration, and with each subsequent configuration arising from its predecessor by the execution of some enabled transition $\tau \in T$. If no transition is enabled in some configuration of that run, then that configuration must be the last in the run (so that the run is finite); we insist in addition that, if the run is finite, then in the final configuration, no transition in T is enabled. Note that, if M is deterministic, then it has a unique run with given input x. We call a run *accepting* if it finite and, in the final configuration, the head is over the left-most square. If M has at least one accepting run on input x, we say that M *accepts* the input x. The language *recognized* by M is the set of inputs x (over the alphabet A) it accepts. It is obvious that, at any point in a run, only the initial square containt $\lhd$, and all but finitely many squares will contain $\sqcup$.

In the régime described above, termination of a run occurs just in case there are no enabled transitions, with acceptance (i.e. successful termination) indicated by leaving the input head over the left-most square. Thus, there is no need for M to have a distinguished 'accepting' state. It is convenient to assume that M does not over-write the input x: this makes no difference to the class of languages which can be accepted by Turing machines, and no essential difference to running times.

By a *language* over A, we simply mean a subset of A^*. A language is said to be *recursively enumerable* (or *r.e.*) if it is recognized by some Turing machine. It is straightforward to show that a language is recursively enumerable if and only it is recognized by some deterministic Turing machine. That is, determinism does not affect the class of recognized languages. A language is *co-recursively enumerable* (or *co-r.e.*) if it is the complement of an r.e. language. A language is *recursive* if it is accepted by some Turing machine which halts on all inputs (equivalently, by some deterministic Turing machine which halts on all inputs). It is straightforward to show that a language is recursive if and only if it is both r.e. and co-r.e.

In Computer Science, a *problem* is usually thought of as classification of some possible inputs (encoded as strings over some finite alphabet A) into positive and negative instances. Thus, a problem may be equivalently identified with the set of its positive instances, which is to say, a language. The so-called Church-Turing thesis states that the intuitive notion of *effective* (or *algorithmic*) *computability* is adequately reconstructed by the notion of a recursive language:

Thesis (Church, Turing). *A problem is effectively computable just in case its set of positive instances* (*under some natural encoding*) *is a recursive language.*

In this case, we shall simply say that the problem in question is *decidable*.

Technically speaking, decidability is relative to the encoding chosen; in practice however, it is almost always clear what counts as a reasonable encoding of sensible problems.

Now suppose that the inputs in question are sentences of first-order logic. It is easy to see that the language consisting of those strings encoding well-formed formulas is itself recognizable. Then Sat($\mathcal{FO}$)—the task of distinguishing those formulas of first-order logic that are satisfiable from those that are not—can be regarded as a problem in this technical sense, since it amounts to nothing other than the recognition of the language of strings encoding the satisfiable first-order formulas. According to the Church-Turing thesis, then, Hilbert's *Entscheidungsproblem* askes for a Turing machine, halting on all inputs, that accepts the language Sat($\mathcal{FO}$). Turing and (less directly) Church independently showed that no such Turing machine exists; Traktenbrot did the same for FinSat($\mathcal{FO}$).

We mentioned in Sec. 1.2 Gödel's result that the set of valid $\mathcal{FO}$-formulas can be alternatively characterized as the set of formulas derivable from a finite set of axiom schemas via the rules of generalization and *modus ponens*. It follows that the set of *un*satisfiable first-order formulas (over some fixed signature) is r.e, since it is a straightforward matter to enumerate derivations. Likewise, the set of *finitely* satisfiable formulas is also r.e., since it is a straightforward matter to enumerate finite structures. It follows that every fragment of first-order logic with the finite model property has a decidable satisfiability problem, since the set of satisfiable (= finitely satisfiable) formulas is both co-r.e. and r.e.

The reconstruction of the intuitive notion of algorithmic decidability in terms of Turing-computability is compelling enough as it is. The following fact further cements this identification, however. Observe first that, since Turing machines are finite objects, they can be described by strings over some fixed alphabet. Turing proved the following fundamental result, which we state here very briefly by making some minor concessions to informality.

Proposition 1.3. *There exists a universal Turing machine, U, which when given as input a string describing another Turing machine M together with a possible input string* x *for M, terminates if and only if M terminates on input* x*, and leaving the same output on the tape.*

1.4 Complexity

We assume familiarity with the usual time- and space-complexity classes defined using the Turing model of computation. (Exotic complexity classes hardly appear in this book.) Readers seeking detailed explanations are referred to standard textbooks, such as C. Papadimitriou [63] or D. Kozen [50]; however, we again provide a swift overview in this section for ease of reference.

Let $f : \mathbb{N} \rightarrow \mathbb{N}$ be a function. A Turing machine M operates in *time bounded by f* if, for all inputs x, no run with input x contains more than $f(|\mathrm{x}|)$ steps; M operates in *space bounded by f* if, for all inputs x, every run with input x halts, and none writes on more than $f(|\mathrm{x}|)$ cells of the tape *in addition to* the input x. (Remember: we make the general assumption that the input x is not over-written.) Let F be a set of functions. By $\mathrm{NT{\scriptsize IME}}(F)$, we mean the class of languages recognized by some Turing machine operating in time bounded by some $f \in F$; and by $\mathrm{T{\scriptsize IME}}(F)$, we mean the class of languages recognized by some *deterministic* Turing machine operating in time bounded by some $f \in F$. Similarly for $\mathrm{NS{\scriptsize PACE}}(F)$ and $\mathrm{S{\scriptsize PACE}}(F)$. On this model of computation, sub-linear time bounds are not interesting, since the tape-head could never scan the end of tits input; however, sub-linear space bounds do make sense. As before, for "language" and "recognize", we most often substitute "problem" and "decide".

Salient sets of functions F occupying an important place in complexity theory are $\mathrm{L} = \{n \mapsto \lceil \frac{1}{2}\log(n+1)\rceil\}$, $\mathrm{P} = \{n \mapsto n^k \mid k \in \mathbb{N}\}$, $\mathrm{Exp} = 1\text{-}\mathrm{Exp} = \{n \mapsto 2^{p(n)} \mid p \in \mathrm{P}\}$, and $(k+1)\text{-}\mathrm{Exp} = \{n \mapsto 2^{f(n)} \mid f \in k\text{-}\mathrm{Exp}\}$, for $k \geq 1$. This allows us to define the time complexity-classes

$$\begin{array}{ll} \mathrm{PT{\scriptsize IME}} = \mathrm{T{\scriptsize IME}}(\mathrm{P}) & \mathrm{NPT{\scriptsize IME}} = \mathrm{NT{\scriptsize IME}}(\mathrm{P}) \\ \mathrm{E{\scriptsize XP}T{\scriptsize IME}} = \mathrm{T{\scriptsize IME}}(\mathrm{Exp}) & \mathrm{NE{\scriptsize XP}T{\scriptsize IME}} = \mathrm{NT{\scriptsize IME}}(\mathrm{Exp}) \\ k\text{-}\mathrm{E{\scriptsize XP}T{\scriptsize IME}} = \mathrm{T{\scriptsize IME}}(k\text{-}\mathrm{Exp}) & k\text{-}\mathrm{NE{\scriptsize XP}T{\scriptsize IME}} = \mathrm{NT{\scriptsize IME}}(k\text{-}\mathrm{Exp}) \end{array}$$

as well as the space complexity-classes

$$\begin{array}{ll} \mathrm{PS{\scriptsize PACE}} = \mathrm{S{\scriptsize PACE}}(\mathrm{P}) & \mathrm{NPS{\scriptsize PACE}} = \mathrm{NS{\scriptsize PACE}}(\mathrm{P}) \\ \mathrm{E{\scriptsize XP}S{\scriptsize PACE}} = \mathrm{S{\scriptsize PACE}}(\mathrm{Exp}) & \mathrm{NE{\scriptsize XP}S{\scriptsize PACE}} = \mathrm{NS{\scriptsize PACE}}(\mathrm{Exp}) \\ k\text{-}\mathrm{E{\scriptsize XP}S{\scriptsize PACE}} = \mathrm{S{\scriptsize PACE}}(k\text{-}\mathrm{Exp}) & k\text{-}\mathrm{NE{\scriptsize XP}S{\scriptsize PACE}} = \mathrm{NS{\scriptsize PACE}}(k\text{-}\mathrm{Exp}). \end{array}$$

It is not difficult to show that a non-deterministic Turing machine can always be determinized with at most a quadradic increase in the memory required, a result commonly known as *Savitch's Theorem*:

Proposition 1.4 (Savitch's Theorem)**.** *If* $f(n) \geq \log(n)$, *then* $\mathrm{NS{\scriptsize PACE}}(f) \subseteq \mathrm{S{\scriptsize PACE}}(f^2)$. *Hence* $\mathrm{NPS{\scriptsize PACE}} = \mathrm{PS{\scriptsize PACE}}$ *and* $k\text{-}\mathrm{NE{\scriptsize XP}S{\scriptsize PACE}} = k\text{-}\mathrm{E{\scriptsize XP}S{\scriptsize PACE}}$ *for all* $k \geq 1$.

It is a relatively simple exercise to show that the remaining complexity classes are arranged in a hierarchy

$$\begin{aligned} &\mathrm{L{\scriptsize OG}S{\scriptsize PACE}} \subseteq \mathrm{NL{\scriptsize OG}S{\scriptsize PACE}} \subseteq \mathrm{PT{\scriptsize IME}} \subseteq \mathrm{NPT{\scriptsize IME}} \subseteq \mathrm{PS{\scriptsize PACE}} \subseteq \\ &\qquad \mathrm{E{\scriptsize XP}T{\scriptsize IME}} \subseteq \mathrm{NE{\scriptsize XP}T{\scriptsize IME}} \subseteq \mathrm{E{\scriptsize XP}S{\scriptsize PACE}} \subseteq 2\text{-}\mathrm{E{\scriptsize XP}T{\scriptsize IME}} \cdots \quad . \end{aligned} \tag{1.1}$$

(Note that we have deleted the redundant non-deterministic space-classes $\mathrm{NPS{\scriptsize PACE}}$, $\mathrm{NE{\scriptsize XP}S{\scriptsize PACE}}$ etc. here.) It is also straightforward to show that exponential increases in time- and space-resources yield strict inclusions, thus:

Proposition 1.5. $\mathrm{NL{\scriptsize OG}S{\scriptsize PACE}} \subsetneq \mathrm{NPS{\scriptsize PACE}} = \mathrm{PS{\scriptsize PACE}}$, $\mathrm{PT{\scriptsize IME}} \subsetneq \mathrm{E{\scriptsize XP}T{\scriptsize IME}}$, $\mathrm{NPT{\scriptsize IME}} \subsetneq \mathrm{NE{\scriptsize XP}T{\scriptsize IME}}$, *etc.*

The strictness or otherwise of the remaining inclusions is famously open.

For any class of languages $\mathcal{C}$, we define co-$\mathcal{C}$ to be the class of languages whose complements (over the relevant alphabet) are in $\mathcal{C}$. Any deterministic complexity class $\mathcal{C}$—for example PTime or ExpSpace—is evidently equal to co-$\mathcal{C}$. Indeed, if a deterministic Turing machine M recognizes a language $\mathcal{L}$, and is guaranteed to halt on all inputs, then we can recognize the complement of $\mathcal{L}$ in (essentially) the same time- and space-bounds by reversing the verdict of M upon halting. Since, by Proposition 1.4, the non-deterministic *space*-classes NPSpace and k-NExpSpace ($k \geq 1$) are equal to their deterministic counterparts, it follows trivially that these too are closed under complementation. Of course, Proposition 1.4 does not imply that NLogSpace = NSpace($\log(n)$) $\subseteq$ Space($\log(n)^2$) is equal to LogSpace = Space($\log(n)$); indeed, it is not known whether these two classes are equal. Nevertheless, it transpires that NLogSpace *is* equal to its complement class, a result commonly known as the *Immerman-Szelepscényi Theorem*:

Proposition 1.6 (Immerman-Szelepscényi Theorem)**.** *If $f(n) \geq \log(n)$, then* NSpace(f) = co-Space(f)*. In particular,* NLogSpace = co-NLogSpace*.*

This argumentation does not apply to non-deterministic *time*-classes. For example, it is not known whether NPTime = co-NPTime, or whether NExpTime = co-NExpTime, and so on.

Locating a problem in a given complexity class provides an upper bound on its complexity; corresponding lower bounds are provided by the apparatus of reductions. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be languages over the respective alphabets A_1 and A_2. A *many-one reduction* is a function $g : A_1^* \rightarrow A_2^*$ such that $g^{-1}(\mathcal{L}_2) = \mathcal{L}_1$. We say that g is *log-space* if it is computable by a deterministic Turing machine M in the sense that, if M starts with input x, it eventually terminates, writing the string $g(\mathrm{x})$ on the tape (say, following x), using, in the process, at most $\log |\mathrm{x}|$ additional tape squares as working memory (if $\mathrm{x} \neq \epsilon$). If such a g exists, we say that $\mathcal{L}_1$ is many-one, log-space reducible to $\mathcal{L}_2$, and write $\mathcal{L}_1 \leq^m_{\log} \mathcal{L}_2$. It is a standard result (though not obvious) that the relation of many-one, log-space reducibility is transitive. Intuitively, and thinking of $\mathcal{L}_1$ and $\mathcal{L}_2$ as problems, $\mathcal{L}_1 \leq^m_{\log} \mathcal{L}_2$ says that $\mathcal{L}_2$ is at least as difficult as $\mathcal{L}_1$, since evidently, any method of solving $\mathcal{L}_2$ yields, following a low-cost translation, a method of solving $\mathcal{L}_1$. All of the complexity classes mentioned above are closed under many-one, log-space reductions: if $\mathcal{L}_1 \in \mathcal{C}$, and $L_1 \leq^m_{\log} \mathcal{L}_2$, then $\mathcal{L}_2 \in \mathcal{C}$. Let $\mathcal{C}$ be a complexity class. Say that a language $\mathcal{L}$ is *hard for* $\mathcal{C}$, or $\mathcal{C}$*-hard* if every language in $\mathcal{C}$ is many-one, log-space reducible to $\mathcal{L}$. Say that $\mathcal{L}$ is *complete for* $\mathcal{C}$, or $\mathcal{C}$*-complete* if it is both in $\mathcal{C}$ and $\mathcal{C}$-hard. Intuitively, a $\mathcal{C}$-complete problem is a maximally difficult problem in $\mathcal{C}$, since all other problems in $\mathcal{C}$ can be reduced to it.

There are complete problems for all of the complexity classes mentioned in the hierarchy (1.1). The best-known of these is the problem

SAT
Given: a set Γ of clauses of propositional logic
Return: Yes if (the conjunction of) Γ is satisfiable;
No otherwise.

It is easy to see that SAT is in NPTime, since, if Γ is a positive instance, a satisfying truth assignment can be guessed and checked in time bounded by a polynomial function of $\|\Gamma\|$. The observation that this problem is also NPTime-hard provided the launching point for the contemporary theory of computational complexity, and is commonly known as *Cook's Theorem* or the *Cook-Levin Theorem*:

Proposition 1.7 (Cook-Levin Theorem)**.** SAT *is* NPTime-*complete.*

It is often more convenient to consider a restricted version of SAT in which every input clause has at most three literals.

3-SAT
Given: a set Γ of propositional clauses each containing at most three literals
Return: Yes if (the conjunction of) Γ is satisfiable;
No otherwise.

In fact, the general problem SAT is many-one log-space reducible to is restriction 3-SAT. It immediately follows that

Proposition 1.8. 3-SAT *is* NPTime-*complete.*

Many other natural problems in NPTime can be shown to be NPTime-complete by reduction to 3-SAT. A case in point is that of *graph 3-colourability.* In this book a *graph* is a pair $G = (V, E)$ where V is a finite (possibly empty) set (the *vertices*) and E is a set of 2-element subsets of V (the *edges*). Thus, graphs, in our sense, have no 'loops' or 'multiple edges'. A 3-*colouring* of G is a function t mapping the vertices of G to the set $\{0, 1, 2\}$ such that no edge of G joins two vertices mapped to the same value. We say that G is 3-*colourable* if a 3-colouring of G exists. The problem

3-COLOURABILITY
Given: a graph G
Return: Yes if there exists a 3-colouring of G;
No otherwise.

is easily seen to be in NPTime, since a colouring can be guessed and straightforwardly checked. Not so obvious is the fact that 3-SAT can be reduced to this problem too, whence we have:

Proposition 1.9. 3-COLOURABILITY *is* NPTime-*complete.*

Another relatively straightforward (and fundamental) hardness result concerns the problem of determining whether one vertex is reachable from another in a directed graph.

REACHABILITY
Given: a directed graph $G = (V, E)$ and $u, v \in V$
Return: Yes if there is a path in G from u to v;
No otherwise.

It is easy to see that REACHABILITY is in NLogSpace, since, for a given directed graph G and vertices u and v, we may simply *guess* a sequence of vertices, starting at u and checking that each guessed vertex is connected to the previous one by an edge of G: if we ever reach v we have found a path from u to v; if we ever guess a vertex not connected to the previous one, or if we have made $|V|$ guesses without success, then we fail. This (non-deterministic) procedure has a successfully terminating run if and only v is reachable from u in G, and evidently requires only logarithmic space, since we need remember only the *index* of last guessed vertex and the *number* of vertices guessed so far. The observation that this problem is also NLogSpace-hard does not seem to have a name:

Proposition 1.10. *REACHABILITY is* NLogSpace*-complete.*

In the course of this book, we will consider the satisfiability and finite satisfiability problems for a great variety of fragments of first-order logic, and show that they are complete for various of the complexity-classes mentioned above.

It is sometimes easier to obtain complexity bounds (and particularly lower bounds) using a slightly generalized model of computation. An *alternating Turing machine* is a Turing machine M whose set of states Q is partitioned into $Q_\exists$ and $Q_\forall$—the *existential* and *universal* states, respectively. The *run* of $M = \langle A, S^*, s_0, T \rangle$ is a (possibly infinite) finitely branching tree whose vertices are labelled by configurations of M. The root is the initial configuration as before, and the daughters of any vertex v are simply the configurations resulting from the execution of those transitions of T which are enabled in v. A vertex is *existential* if, in the configuration in question, M is in a state $q \in Q_\exists$; otherwise, the vertex is *universal.* The set of *accepting* vertices of the run is the smallest set v of vertices Y satisfying the following two conditions: (i) if v is existential, there exists a daughter w of v such that $w \in Y$; (ii) if v is universal, every daughter w of v is in Y. The run as a whole is *accepting* if the root is accepting. Note that a vertex with no daughters (no enabled transitions) will be accepting just in case it is universal. The time and space requirements of the run of an alternating Turing Machine are defined in the expected way. In particular, if M is time- or space-bounded by some function f, we take its run to be finite. This means in particular that there are no enabled transitions at the leaves.

If F is a set of functions $f : \mathbb{N} \to \mathbb{N}$, then the complexity class ATime(F) is the class of languages recognized by an alternating Turing machine running in time f, for some $f \in F$, and similarly for ASpace(F). Accordingly, we define

$$\begin{array}{ll} \text{APTime} = \text{ATime(P)} & \text{APSpace} = \text{ASpace(P)} \\ \text{AExpTime} = \text{ATime(E)} & \text{AExpSpace} = \text{ASpace(E)} \end{array}$$

and so on, to obtain the standard alternating complexity classes. These turn out to merge with their non-alternating cousins.

Proposition 1.11. *We have* APTime = PSpace *and* APSpace = ExpTime*. Moreover, for all* $k \geq 1$*,* Ak-ExpTime = k-ExpSpace *and* Ak-ExpSpace = $(k+1)$-ExpTime*.*

Thus, the hierarchy of deterministic time- and space-classes in (1.1) is 'shifted right' by alternation. This fact sometimes facilitates the task of showing that a particular problem is hard for certain complexity classes, in particular when instances of the problem in question can be naturally used to encode runs of alternating Turing machines. We remark that, when considering alternating Turing machines, one can without loss of generality suppose that the alternation is *strict* (i.e. the transitions in T lead from existential states only to universal states and vice versa), and in fact *binary branching* (i.e. in each configuration either no transitions are enabled or exactly two are). This does not affect the set of accepted languages or essentially increase the time- or space-requirements.

Part I

Basics

Chapter 2

Origins

Consider the set English sentences having any of the four forms

$$\begin{array}{ll} \text{Every } p \text{ is a } q & \text{No } p \text{ is a } q \\ \text{Some } p \text{ is a } q & \text{Some } p \text{ is not a } q, \end{array} \tag{2.1}$$

where p and q are common (count) nouns, for example: “Every artist is a beekeeper”, “No dentist is an electrician”, and so on. It is easy to construct logically valid inferences involving such sentences. Thus, in the following example, we see that, if the premises (above the horizontal line) are true then so must be the conclusion (below the horizontal line):

$$\begin{array}{l} \text{Some artist is a beekeeper} \\ \text{Every beekeeper is a carpenter} \\ \text{Every carpenter is a dentist} \\ \hline \text{Some artist is a dentist.} \end{array} \tag{2.2}$$

Aristotle’s *Prior Analytics* is devoted, in large part, to the problem of determining the validity of arguments involving (modulo translation) precisely the sentence forms in (2.1). What Aristotle bequeathed to us, more precisely, is a complete list of the valid two-premise argument forms in this language. Such rules are commonly called *syllogisms*. Here are two of them, known by their mediæval mnemonics *Barbara* and *Darii*, respectively:

$$\frac{\text{Every } p \text{ is a } q \quad \text{Every } o \text{ is a } p}{\text{Every } o \text{ is a } q} \qquad \frac{\text{Every } p \text{ is a } q \quad \text{Some } o \text{ is a } p}{\text{Some } o \text{ is a } q} \,. \tag{2.3}$$

Syllogisms may be chained together to form derivations establishing the validity of arguments with more than two premises. Here is a derivation showing the validity of argument in (2.2) using the syllogisms in (2.3):

$$\frac{\text{Some artist is a beekeeper} \quad \dfrac{\text{Every beekeeper is a carpenter} \quad \text{Every carpenter is a dentist}}{\text{Every beekeeper is a dentist}}}{\text{Some artist is a dentist.}}$$

The sentence forms in (2.1) may be rendered in first-order logic by formulas of the forms

$$\begin{array}{ll} \forall x(p(x) \rightarrow q(x)) & \forall x(p(x) \rightarrow \neg q(x)) \\ \exists x(p(x) \wedge q(x)) & \exists x(p(x) \wedge \neg q(x)), \end{array} \qquad (2.4)$$

where p and q are unary (i.e. 1-place) predicates. These renditions are justified, of course, by the fact the intuitive notion of logical validity corresponds precisely to validity of the corresponding sequent under the standard semantics of first-order logic. Let us call the set of first-order formulas of the forms (2.4) the *classical syllogistic*, and denote it by $\mathcal{S}$. Thus, the classical syllogistic is a fragment of first-order logic. Since Aristotle had no conception of an independent semantics in the modern sense, he could not properly formulate the question of whether his syllogisms are adequate to account for all valid arguments in this way. As we shall see, with a little good will in the form of additional rules to handle 'corner cases', this is however the case.

A moment's thought shows that the fragment $\mathcal{S}$ is, effectively, closed under negation. For example, the negation of the $\mathcal{S}$-formula $\forall x(p(x) \rightarrow q(x))$ is logically equivalent to the $\mathcal{S}$-formula $\exists x(p(x) \wedge \neg q(x))$. Since an argument with premises Φ and conclusion ψ is valid if and only if the set of formulas $\Phi \cup \{\neg\psi\}$ is unsatisfiable, we may direct our attention to the satisfiability problem

Sat($\mathcal{S}$):
Given: a finite set Φ of $\mathcal{S}$-formulas
Return: Yes if Φ is satisfiable;
No otherwise.

Clearly, any algorithm for solving the satisfiability problem Sat($\mathcal{S}$) immediately yields a method for determining the validity of putative arguments in the classical syllogistic. Of course, when devising such algorithms, we may employ any techniques we choose, and are certainly not limited to those based on inference rules such as the Aristotelian syllogisms. On the other hand, syllogism-like rules are sometimes surprisingly useful in analysing the complexity of inference, as we shall see.

Perusal of the formulas in (2.4) reveals some 'missing' possibilities. Thus, the classical syllogistic does not contain formulas corresponding to the sentence-forms

Every non-p is a q Some non-p is not a q. (2.5)

This is not exactly English-as-she-is-spoken, of course, but it is close enough; and at any rate such sentence-forms—or rather their pseudo-Greek equivalents—are considered in Aristotle's *De Interpretatione*. They are naturally rendered in first-order logic as, respectively:

$$\forall x(\neg p(x) \rightarrow q(x)) \qquad \exists x(\neg p(x) \wedge \neg q(x)). \qquad (2.6)$$

Accordingly, we call the set of first-order formulas having any of the forms in (2.4) or (2.6) the *extended classical syllogistic*, denoted $\mathcal{S}^+$, and we define the satisfiability problem, $\mathrm{Sat}(\mathcal{S}^+)$, analogously to that for $\mathcal{S}$. Since the formulas $\exists x(\neg p(x) \wedge q(x))$ and $\forall x(\neg p(x) \rightarrow \neg q(x))$ are logically equivalent to the respective $\mathcal{S}^+$-formulas $\exists x(q(x) \wedge \neg p(x))$ and $\forall x(q(x) \rightarrow p(x))$, we see that $\mathcal{S}^+$ is, up to logical equivalence, the set of all formulas of the forms $\forall x(\pm q(x) \rightarrow \pm p(x))$ and $\exists x(\pm q(x) \wedge \neg \pm p(x))$, and thus forms a syntactically rather natural fragment of first-order logic. In particular, $\mathcal{S}^+$ is effectively closed under negation in the same sense as $\mathcal{S}$; hence, any algorithm for solving the satisfiability problem $\mathrm{Sat}(\mathcal{S}^+)$ immediately yields a method for determining the validity of putative arguments in the extended classical syllogistic.

The sentence forms in (2.1) and (2.5) feature no transitive verbs, and therefore have no access to the structure of relational facts. It is natural to ask what would happen if we extended the language with such a facility, incorporating sentences such as, for example,

Every artist admires some beekeeper
No dentist hates every electrician. (2.7)

Here, we have a subject-quantifier, a transitive verb (possibly negated) and an object-quantifier. And these resources allow us to formulate arguments whose validity is less apparent:

$$\frac{\begin{array}{l}\textsf{Some artist hates no beekeeper}\\ \textsf{Every beekeeper hates some artist}\end{array}}{\textsf{Some artist is not a beekeeper.}} \tag{2.8}$$

Indeed, consider an artist, a, who hates no beekeeper. If a is not a beekeeper, the conclusion is true; otherwise, there is an artist, b, whom a hates. Since a hates no beekeeper, but does hate b, the latter is not a beekeper, and the conclusion is again true. One might wonder whether this extended language has a set of inference rules analogous to those for the classical syllogistic, able to account for all valid arguments. We shall see that this is in fact not the case.

The sentences (2.7) are naturally rendered in first-order logic as

$$\begin{array}{l}\forall x(\mathrm{admires}(x) \rightarrow \exists y(\mathrm{artist}(y) \wedge \mathrm{admires}(x,y)))\\ \forall x(\mathrm{dentist}(x) \rightarrow \exists y(\mathrm{electrician}(y) \wedge \neg\mathrm{hate}(x,y))).\end{array} \tag{2.9}$$

We refer to the classical syllogistic together with formulas such as (2.9) as the *relational syllogistic*, and denote it by $\mathcal{R}$. (A precise definition will be given below.) Again, perusal of these formulas reveals missing possibilities analogous to those of $\mathcal{S}$, for example, as expressed by the pseudo-English sentence:

Every *non*-artist admires some *non*-beekeeper

$$\forall x(\neg\mathrm{admires}(x) \rightarrow \exists y(\neg\mathrm{artist}(y) \wedge \mathrm{admires}(x,y))).$$

We call the resulting set of first-order formulas, together with those of the extended classical syllogistic, the *extended relational syllogistic*, and denote it by $\mathcal{R}^+$. (Again, a precise definition is given below.) Both $\mathcal{R}$ and $\mathcal{R}^+$ turn out to be closed under negation; hence, any algorithm for solving the satisfiability problems Sat($\mathcal{R}$), and Sat($\mathcal{R}^+$) again yields a method for determining the validity of putative arguments in the relational syllogistic and extended relational syllogistic.

In Section 2.1, we show that the problems Sat($\mathcal{S}$) and Sat($\mathcal{S}^+$) are decidable, and establish their computational complexity. In Section 2.2, we show that the Aristotelian syllogisms (with some minor additions) are adequate for establishing the validity of arguments in $\mathcal{S}^+$. In Sec. 2.3, we show that there is a finite collection of inference rules that can be used to establish the validity of arguments in $\mathcal{R}$, *though only indirectly*, by supposing the negation of the conclusion, and inferring an absurdity. We conclude that the problem Sat($\mathcal{R}$) is decidable, and establish its computational complexity. In Sec. 2.4, we show, using very different techniques, that the problem Sat($\mathcal{R}^+$) is decidable, but with higher computational complexity. We deduce, as a corollary, that that there is *no* finite collection of syllogism-like inference rules adequate for establishing the satisfiability of finite sets of sentences in $\mathcal{R}^+$, even indirectly.

Thus, we begin this book, as logic itself began, with the syllogism. However, we do so not out of any particular fascination with classical antiquity, but merely as a convenient point of departure for the concepts, themes and techniques we shall encounter in the sequel. For the classical syllogistic and its cousins are perhaps the simplest and most perspicuous examples of naturally delineated fragments of first-order logic whose model-theoretic, proof-theoretic and computational properties are worthy of study. The topic itself may be ancient, but our analysis will be thoroughly modern.

2.1 The classical syllogistic

Let us establish some notation. In the fragments $\mathcal{S}$ and $\mathcal{S}^+$, the Boolean connective ($\wedge$ or $\rightarrow$) is actually determined by the preceding quantifier ($\exists$ or $\forall$); moreover, the variable x conveys no particular information (a topic to which we return in this book's final chapter). Both elements may therefore be deleted. Accordingly, we use the term *unary atom* as a synonym for a *unary predicate*, and *unary literal* for an expression of either of the forms p or $\bar{p}$, where p is a unary atom; and with these conventions, we may regard $\mathcal{S}$ as the set of expressions

$$\exists(p, \ell), \qquad \exists(\ell, p), \qquad \forall(p, \ell), \qquad \forall(\ell, \bar{p}),$$

where p is a unary atom and ℓ a unary literal. Likewise, we may regard $\mathcal{S}^+$ as the set of expressions

$$\exists(\ell, m), \qquad \forall(\ell, m),$$

where ℓ and m are unary literals. We shall employ this more compact notation in the remainder of this chapter. Formulas beginning with $\forall$ we shall call *universal*; those beginning with $\exists$, *existential.*

A unary literal is called *positive* if it is a unary atom; otherwise, *negative.* We use the (possibly decorated) variables o, p, q to range over unary atoms, and

l, m (and occasionally n) to range over unary literals. The languages $\mathcal{S}$ and $\mathcal{S}^+$ of course inherit their semantics semantics from first-order logic. Alternatively, we may give them directly: if $\mathfrak{A}$ is a structure and p a unary predicate, write $p^{\mathfrak{A}}$ for the interpretation of p in $\mathfrak{A}$, and extend this notation to all literals by setting $\bar{p}^{\mathfrak{A}} = A \setminus p^{\mathfrak{A}}$. We then simply define:

$$\mathfrak{A} \models \exists(\ell, m) \text{ iff } \ell^{\mathfrak{A}} \cap m^{\mathfrak{A}} \neq \emptyset$$
$$\mathfrak{A} \models \forall(\ell, m) \text{ iff } \ell^{\mathfrak{A}} \subseteq m^{\mathfrak{A}},$$

and everything is as expected. In particular, valid sequents then correspond to intuitively valid arguments in the obvious sense. For example, the sequent

$$\{\exists(\text{artist}, \text{beekeeper}),\ \forall(\text{beekeeper}, \text{carpenter}),$$
$$\forall(\text{carpenter}, \text{dentist})\} \models \exists(\text{carpenter}, \text{dentist}) :$$

corresponds to the argument (2.2).

If $\ell = \bar{p}$ is a negative literal, define $\bar{\ell}$ to be the positive literal p. If $\varphi = \forall(p, \ell)$ is a universal formula, define $\bar{\varphi}$ to be the existential formula $\exists(p, \bar{\ell})$; similarly, if $\varphi = \exists(p, \ell)$, define $\bar{\varphi} = \forall(p, \bar{\ell})$. Likewise for $\mathcal{S}^+$-formulas. It is obvious that $\bar{\bar{\varphi}} = \varphi$, and that $\mathfrak{A} \models \varphi$ if and only if $\mathfrak{A} \not\models \bar{\varphi}$. We refer to $\bar{\varphi}$ as the *negation* of φ, and think of is as the 'formula' $\neg\varphi$. Thus, as remarked above, both the classical syllogistic and the extended classical syllogistic are closed under negation. The $\mathcal{S}^+$-formulas $\exists(\ell, m)$ and $\exists(m, \ell)$ are clearly logically equivalent, as are $\forall(\ell, m)$ and $\forall(\bar{m}, \bar{\ell})$. In the sequel, therefore, we shall simply identify them, changing silently from one form to the other. Indeed, by agreeing to treat $\exists(\bar{p}, q)$ and $\forall(\bar{p}, \bar{q})$, by courtesy, as $\mathcal{S}$-formulas (which we may harmlessly do), we can perform such alternations in the langauge $\mathcal{S}$ as well.

A set of literals V is said to be *consistent* if there exists no literal ℓ such that both ℓ and $\bar{\ell}$ are in V. Let Φ be a set of universal $\mathcal{S}^+$-formulas and ℓ, m unary literals. We write $\ell \Rightarrow_\Phi m$ if there exists a sequence of unary literals $\ell_0, \ldots, \ell_k$ ($k \geq 0$) such that $\ell = \ell_0$, $\ell_k = m$, and $\forall(\ell_i, \ell_{i+1}) \in \Phi$ for all i ($0 \leq i < k$). Let V be a set of literals and Φ a set of universal $\mathcal{S}^+$-formulas. Define

$$V^\Phi = \{m \mid \ell \Rightarrow_\Phi m \text{ for some } \ell \in V \text{ or } \bar{m} \Rightarrow_\Phi m\}.$$

We say that V is Φ-*closed* (*under* Φ) if $V = V^\Phi$. The function $V \mapsto V^\Phi$ is not a closure operator in the usual sense, since $\emptyset^\Phi$ need not be $\emptyset$. Nevertheless, we do have $V \subseteq V^\Phi$ and $(V^\Phi)^\Phi = V^\Phi$; moreover, the union of any collection of Φ-closed sets of literals is Φ-closed.

Lemma 2.1. *Let Φ be a set of universal $\mathcal{S}^+$-formulas. Any non-empty, Φ-closed, consistent set of literals has a Φ-closed, consistent, complete extension.*

Proof. Let V_0 be a non-empty, Φ-closed, consistent set of literals. Enumerate the set of all literals as $\ell_1, \ell_2, \ldots$. For all $i \geq 0$, define

$$V_{i+1} = \begin{cases} V_i \text{ if } \ell_{i+1} \in V_i \\ \left(V_i \cup \{\bar{\ell}_{i+1}\}\right)^\Phi \text{ otherwise.} \end{cases}$$

and define $V = \bigcup_i V_i$. Thus, V is Φ-closed, complete and includes V_0. We remark that, since $V_0 \subseteq V_i$, V_i is non-empty for all $i \geq 0$. To show that V is consistent, suppose otherwise. Let i be the least natural number such that V_{i+1} is inconsistent. Hence, $V_{i+1} = \left(V_i \cup \{\bar{\ell}_{i+1}\}\right)^\Phi$, where $\ell_{i+1} \notin V_i$. Since V_{i+1} is supposedly inconsistent, let m be such that $m, \bar{m} \in V_{i+1}$. Since we certainly do not have $m, \bar{m} \in V_i$, suppose, without loss of generality that $m \in V_{i+1} \setminus V_i$. Hence $\bar{\ell}_{i+1} \Rightarrow_\Phi m$, i.e. $\bar{m} \Rightarrow_\Phi \ell_{i+1}$, whence $\bar{m} \notin V_i$, since otherwise $\ell_{i+1} \in V$, by the fact that V is Φ-closed, contrary to assumption. But since $\bar{m} \in V_{i+1} = \left(V_i \cup \{\bar{\ell}_{i+1}\}\right)^\Phi$, we have $\bar{\ell}_{i+1} \Rightarrow_\Phi \bar{m}$, and thus $\bar{l}_{i+1} \Rightarrow_\Phi \ell_{i+1}$, and so $\ell_{i+1} \in V_i$, again a contradiction. □

The next lemma gives us a method for solving a special case of Sat($\mathcal{S}^+$).

Lemma 2.2. *Let Φ be a set of universal $\mathcal{S}^+$-formulas and $\exists(\ell, m)$ an existential $\mathcal{S}^+$-formula. Then $\Phi \cup \{\exists(\ell, m)\}$ is satisfiable if and only if $\{\ell, m\}^\Phi$ is consistent.*

Proof. Suppose $\mathfrak{A} \models \Phi \cup \{\exists(\ell, m)\}$, and pick $a \in A$ such that $a \in \ell^{\mathfrak{A}} \cap m^{\mathfrak{A}}$. A simple induction on the lengths of $\Rightarrow_\Phi$-chains shows that, if $m' \in \{\ell, m\}^\Phi$, then $a \in (m')^{\mathfrak{A}}$. Hence, $\{\ell, m\}^\Phi$ is consistent. Conversely, suppose $\{\ell, m\}^\Phi$ is consistent, and let V^* be a consistent, complete extension, by Lemma 2.2. Define a structure $\mathfrak{A}$ over the singleton domain $\{a\}$ by setting $a \in p^{\mathfrak{A}}$ if and only if $p \in V^*$ for any unary atom p. Therefore $a \in \ell^{\mathfrak{A}}$ if and only if $\ell \in V^*$, and a simple check verifies that $\mathfrak{A} \models \Phi \cup \{\exists(\ell, m)\}$. □

Unravelling the definition of $\{\ell, m\}^\Phi$, Lemma 2.2 tells us that $\Phi \cup \{\exists(\ell, m)\}$ is *un*satisfiable just in case, for some unary atom $o \in \Sigma_0$, both (i) and (ii) hold:

(i) $\ell \Rightarrow_\Phi o$ or $m \Rightarrow_\Phi o$ or $\bar{o} \Rightarrow_\Phi o$; and

(ii) $\ell \Rightarrow_\Phi \bar{o}$ or $m \Rightarrow_\Phi \bar{o}$ or $o \Rightarrow_\Phi \bar{o}$.

The next lemma tells us that Lemma 2.2 is all we need for the general case.

Lemma 2.3. *Let Φ be a set of universal $\mathcal{S}^+$-formulas and Ψ a non-empty set of existential $\mathcal{S}^+$-formulas. Then $\Phi \cup \Psi$ is satisfiable if and only if, for each $\psi \in \Psi$, the set of formulas $\Phi \cup \{\psi\}$ is satisfiable.*

Proof. For the non-trivial direction, for each $\psi \in \Psi$, let $\mathfrak{A}_\psi \models \Phi \cup \{\psi\}$, and let $\mathfrak{A}$ be the disjoint union of the various $\mathfrak{A}_\psi$. Obviously, $\mathfrak{A} \models \Phi \cup \Psi$. □

This yields a quadratic-time algorithm for determining satisfiability in $\mathcal{S}^+$. To test whether $\Phi \cup \Psi$ is satisfiable, where Φ is a set of universal formulas and Ψ a set of existential formulas, check whether the set $\Phi \cup \{\psi\}$ is satisfiable for every $\psi \in \Psi$:

```
begin SatS($\Phi$,$\Psi$)
    for all $\exists(\ell, m) \in \Psi$
        for all unary atoms $o$ mentioned in $\Phi$
```

```
4.         if ($\ell \Rightarrow_\Phi o$ or $m \Rightarrow_\Phi o$ or $\bar{o} \Rightarrow_\Phi o$) & ($\ell \Rightarrow_\Phi o$ or $m \Rightarrow_\Phi o$ or $\bar{o} \Rightarrow_\Phi o$)
5.             return No
6.     return Yes
7. end
```

Determining the conditions of the form $\ell \Rightarrow_\Phi m$ in line 4 is of course easy. Let $G = (V, E)$ be the directed graph whose vertices are all the literals p, $\bar{p}$, where p occurs in Φ, and whose edges are the pairs $(\ell, m) \in E$ such that $\forall(\ell, m) \in \Phi$. Remember: we identify the formulas $\forall(\ell, m)$ and $\forall(\bar{m}, \bar{\ell}) \in \Phi$; hence, (ℓ, m) is an edge if and only if $(\bar{m}, \bar{\ell})$ is. Then $\ell \Rightarrow_\Phi m$ just says that m is reachable from ℓ in G. That is, this condition is an instance of the problem REACHABILITY defined in Sec. 1.4. As we remarked, this problem can be solved in linear time, for example by the standard depth-first search algorithm.

More significantly for our purposes here, the problem REACHABILITY is in NLogSpace, as observed in Prop. 1.10. Together with the Immerman-Szelepcśenyi theorem (Prop. 1.6), this yields an alternative complexity bound for Sat($\mathcal{S}^+$).

Lemma 2.4. Sat($\mathcal{S}^+$) *is in* NLogSpace.

Proof. Consider the following non-deterministic algorithm, where Φ is a set of universal $\mathcal{S}^+$-formulas and Ψ a set of existential $\mathcal{S}^+$-formulas:

```
1. begin NDunSatS($\Phi$,$\Psi$)
2.     guess unary literals $\ell$, $m$ and unary atom $o$
3.     if  $\exists(\ell, m) \notin \Psi$
4.         return No
5.     if  ($\ell \Rightarrow_\Phi o$ or $m \Rightarrow_\Phi o$ or $\bar{o} \Rightarrow_\Phi o$) & ($\ell \Rightarrow_\Phi o$ or $m \Rightarrow_\Phi o$ or $\bar{o} \Rightarrow_\Phi o$)
7.         return Yes
7.     return No
8. end
```

Lemmas 2.2 and 2.3 ensure that this algorithm has a run asnwering Yes if and only if $\Phi \cup \Psi$ is *un*satisfiable. Moreover, by Proposition 1.10, conditions of the form $\ell \Rightarrow_\Phi m$ can be tested non-deterministically in logarithmic space, whence NDunSatS(Φ,Ψ), with the tests in line 5 implemented using the usual non-deterministic procedure for REACHABILITY, requires only logarithmic space. Thus, the complement of Sat($\mathcal{S}^+$) is in NLogSpace. The result then follows by Proposition 1.6. □

A matching lower bound is also easy to obtain.

Lemma 2.5. *The problem* Sat($\mathcal{S}$) *is* NLogSpace-*hard.*

Proof. We proceed by reduction from un-REACHABILITY (the complement of REACHABILITY), which, by Propositions 1.6 and 1.10, is NLogSpace-complete. Let $G = (V, E)$ be a directed graph, and $u, v \in V$: we construct a set of $\mathcal{S}$-formulas $\Phi_{G,u,v}$ such that $\Phi_{G,u,v}$ is satisfiable if and only if v is not reachable from u in G. Let the vertices V be regarded as unary atoms, and set

$$\Phi_{G,u,v} = \{\exists(u, \bar{v})\} \cup \{\forall(u, v) \mid (u, v) \in E\}.$$

It is routine to check that $\Phi_{G,u,v}$ can be computed (from G, u and v) using space logarithmic in the size of G. We must show that the mapping $(G, u, v) \mapsto \Phi_{G,u,v}$ is indeed a reduction of un-REACHABILITY to Sat($\mathcal{S}$). That is, we must show that $\Phi_{G,u,v}$ is satisfiable if and only if v is not reachable from u in G.

Suppose that v is not reachable from u. Define a structure $\mathfrak{A}$ with singleton domain $A = \{a\}$ by setting $a \in p^{\mathfrak{A}}$ just in case p is reachable from u. Then $\mathfrak{A} \models \Phi$ by construction, and $\mathfrak{A} \models \exists(u, \bar{v})$ by the fact that u is reachable from u but v is not. Suppose conversely that v is reachable from u. In any model of $\mathfrak{A} \models \Phi$, if $a \in u^{\mathfrak{A}}$, then, by a simple induction on the lengths of paths, if any vertex w is reachable from u, we have $a \in w^{\mathfrak{A}}$. Thus, $\mathfrak{A} \not\models \exists(u, \bar{v})$. □

Lemmas 2.4 and 2.5 yield

Theorem 2.6. *The problems* Sat($\mathcal{S}$) *and* Sat($\mathcal{S}^+$) *are* NLogSpace-*complete.*

The above argument shows us a little more. If Φ is satisfiable, it is natural to ask how the size of a smallest model of Φ depends on Φ. Inspection of the proofs of Lemmas 2.2 and 2.3 shows:

Corollary 2.7. *Let Φ be a set of universal $\mathcal{S}^+$-formulas and Ψ a non-empty set of existential $\mathcal{S}^+$-formulas. If $\Phi \cup \Psi$ is satisfiable then it is satisfiable over a domain of at most $|\Psi|$ elements.*

Recall from Sec. 1.2 that a logic $\mathcal{L}$ is said to have the *finite model property* if every finite set Φ of $\mathcal{L}$-formulas which is satisfiable is finitely satisfiable—i.e., if the problems Sat($\mathcal{L}$) and FinSat($\mathcal{L}$) coincide. Corollary 2.7 ensures that $\mathcal{S}^+$ has the finite model property; hence, we do not have to ask separately about the complexity of FinSat($\mathcal{S}^+$) or FinSat($\mathcal{S}$). In fact, Corollary 2.7 gives us a computable (actually, linear) bound on the minimum size of any model of Φ, as a function of $\|\Phi\|$. We sometimes refer to such results as establishing a *small model property* for the logic in question. In the sequel, we shall obtain a great many such small-model properties.

2.2 Syllogisms

Aristotle did not provide a model-theoretic semantics for the classical syllogistic, and one can only speculate on what he would have made of the analysis of Sec. 2.1. The main subject of the *Prior Analytics*, in fact, is the syllog*ism.* And it seems only fitting that we should devote some space to this topic here, though, again, our approach will be thoroughly contemporary. In fact, the machinery developed in this section will be useful in determining the complexity of the satisfiability problem for the relational syllogistic in Sec. 2.3. We shall be working here with the extended classical syllogistic $\mathcal{S}^+$, as it makes for a neater presentation; analgous results can be obtained for $\mathcal{S}$.

To streamline the presentation, let us agree henceforth to use the term *syllogism* to denote any inference rule (in the language at hand) consisting of a finite set Φ of formulas, called the *antecedents*, and an additional formula, ψ, called the *consequent*. This syllogism is *valid* if $\Phi \models \psi$. We shall generally display syllogisms with the antecedents separated from the conclusion by a horizontal line. Aristotle listed all the valid two-premise syllogisms in $\mathcal{S}$. having exactly two premises. We have already encountered the rules of *Barbara* and *Darii* in (2.3), which, in the more compact syntax of $\mathcal{S}$, become, respectively,

$$\frac{\forall(o,p) \qquad \forall(p,q)}{\forall(o,q)} \qquad\qquad \frac{\exists(o,p) \qquad \forall(p,q)}{\exists(o,q)}\;.$$

Aristotle's list is actually longer than the one we would give, because he worked under the assumption that predicates have non-empty extensions. In keeping with modern practice, we shall make no such assumption here, though we shall always insist that structures have non-empty domains. Derivations may be constructed from syllogisms by uniform substitution (of atoms for atoms), and chaining into tree-like structures in the familiar way. We have already illustrated this for the argument (2.2). In our new syntax, the derivation is:

$$\cfrac{\exists(\text{artist}, \text{beekeeper}) \qquad \cfrac{\forall(\text{beekeeper}, \text{carpenter}) \quad \forall(\text{carpenter}, \text{dentist})}{\forall(\text{beekeeper}, \text{dentist})}\ {\scriptstyle\text{Barbara}}}{\exists(\text{artist}, \text{dentist})}\ {\scriptstyle\text{Darii}}.$$

Given any (finite) set of rules X, we write $\Phi \vdash_{\mathsf{X}} \psi$ if ψ can be derived from the set of premises Φ via the rules in X. We seek a set of rules X such that $\vdash_{\mathsf{X}}$ is *sound* ($\Phi \vdash_{\mathsf{X}} \psi$ entails $\Phi \models \psi$) and *complete* ($\Phi \models \psi$ entails $\Phi \vdash_{\mathsf{X}} \psi$).

When presenting syllogisms for $\mathcal{S}$ and $\mathcal{S}^+$, we make use of the convention that the letters ℓ and m range over binary literals to compress the presentation. Thus, we could subsume the two valid Aristotelian syllogisms (*Darii* and *Ferio*)

$$\frac{\exists(o,p) \qquad \forall(p,q)}{\exists(o,q)} \qquad \frac{\exists(o,p) \qquad \forall(p,\bar{q})}{\exists(o,\bar{q})}$$

under the single presentation

$$\frac{\exists(o,p) \qquad \forall(p,\ell)}{\exists(o,\ell)}\;.$$

Remember also that we silently identify the formulas $\exists(\ell, m)$ and $\exists(m, \ell)$, and similarly for $\forall(\ell, m)$ and $\forall(\bar{m}, \bar{\ell})$.

Let S^+ be the following set of rules, where ℓ, m and n range over unary literals, and and φ, ψ over $\mathcal{S}$-formulas:

$$\frac{\exists(\ell,n) \qquad \forall(\ell,m)}{\exists(m,n)}\ (\mathrm{D}) \qquad \frac{\forall(\ell,m) \qquad \forall(m,n)}{\forall(\ell,n)}\ (\mathrm{B}) \qquad \frac{\psi \quad \bar{\psi}}{\varphi}\ (\mathrm{X})$$

$$\frac{\forall(\ell,\bar{\ell})}{\forall(\ell,m)}\ (\mathrm{A}) \qquad \frac{}{\forall(\ell,\ell)}\ (\mathrm{T}) \quad \frac{\exists(\ell,m)}{\exists(\ell,\ell)}\ (\mathrm{I}) \qquad \frac{\forall(\bar{\ell},\ell)}{\exists(\ell,\ell)}\ (\mathrm{N})$$

Rules (D) and (B) are generalizations of Aristotelian syllogisms. Rule (X) is the mediæval rule of *ex falso quodlibet*: from a contradiction, anything follows. Rules (A), (T) (I) and (N) have no classical or mediæval counterparts. Rule (A) stems from the fact that if every ℓ is a non-ℓ, then there are no ℓ whatsoever; vacuously, then, every ℓ is an m. To see that (T) is needed, note that without it there would be no way to derive $\forall(p,p)$ from the empty set of premises. Rule (I) is self-explanatory. Rule (N) is needed because of our assumption that domains are non-empty: the premise states that everything is an ℓ; therefore something is an ℓ

It is obvious that $\vdash_{\mathsf{S}+}$ is sound. We now prove completeness. In the remainder of this section, then, a *formula* is an $\mathcal{S}^\dagger$-formula unless otherwise stated. If $\mathfrak{A}$ and $\mathfrak{B}$ are structures with disjoint domains A and B, respectively, denote by $\mathfrak{A} \cup \mathfrak{B}$ the structure with domain $A \cup B$ and interpretations $p^{\mathfrak{A}\cup\mathfrak{B}} = p^{\mathfrak{A}} \cup p^{\mathfrak{B}}$ for any unary atom p. (Note that $\mathcal{S}^\dagger$ features only unary atoms.)

Lemma 2.8. *Suppose $\Phi \cup \Psi \models \theta$, where Φ is a set of universal formulas, Ψ a set of existential formulas, and θ a formula.*

1. *If $\Phi \cup \Psi$ is satisfiable and θ is universal, then $\Phi \models \theta$.*

2. *If $\Psi \neq \emptyset$ and θ is existential, then there exists $\psi \in \Psi$ such that $\Phi \cup \{\psi\} \models \theta$.*

3. *If $\Psi = \emptyset$ and $\theta = \exists(\ell, m)$ is existential, then $\Phi \models \forall(\bar{\ell}, \ell)$ and $\Phi \models \forall(\bar{m}, m)$.*

Proof. In each case, assume the contrary.

1. There exist structures $\mathfrak{A} \models \Phi \cup \Psi$ and $\mathfrak{B} \models \Phi \cup \bar{\theta}$. We may assume that $A \cap B = \emptyset$. But then $\mathfrak{A} \cup \mathfrak{B} \models \Phi \cup \Psi \cup \{\bar{\theta}\}$, a contradiction.

2. For every $\psi \in \Psi$, there exists a structure $\mathfrak{A}_\psi$ such that $\mathfrak{A}_\psi \models \Phi \cup \{\psi, \bar{\theta}\}$. Again, we assume that the domains are pairwise disjoint. But then $\bigcup_{\psi\in\Psi} \mathfrak{A}_\psi \models \Phi \cup \Psi \cup \{\bar{\theta}\}$, a contradiction.

3. If $\Phi \not\models \forall(\bar{\ell}, \ell)$, there exists a structure $\mathfrak{A}$ such that $\mathfrak{A} \models \Phi \cup \{\exists(\bar{\ell}, \bar{\ell})\}$. Choose $a \in \bar{\ell}^{\mathfrak{A}}$, and let $\mathfrak{B}$ be the structure obtained by restricting $\mathfrak{A}$ to the singleton domain $\{a\}$. Then $\mathfrak{B} \models \Phi \cup \{\bar{\theta}\}$, a contradiction. A similar argument applies if $\Phi \not\models \forall(\bar{m}, m)$.

□

We write $\Phi \vdash_{\mathrm{BTA}} \varphi$ if there is a derivation of φ from Φ employing only the rules (B), (T) and (A). Given any set of universal formulas Φ, the relation $\ell \Rightarrow_{\Phi} m$ certainly implies $\Phi \vdash_{\mathrm{BTA}} \forall(l, m)$. For suppose there exists a sequence of unary literals $\ell = \ell_0, \ldots, \ell_k = m$ such that $\forall(\ell_i, \ell_{i+1}) \in \Phi$ for all i $(0 \leq i < k)$. If $k = 0$, then $\ell = m$, and we have the derivation

$$\frac{}{\forall(m,m)}\ (\mathrm{T});$$

and if $k > 0$, then, supposing the result to hold for smaller k, we have the derivation

$$\frac{\begin{array}{c}\vdots\\ \forall(\ell,\ell_{k-1})\end{array} \qquad \forall(\ell_{k-1},m)}{\forall(\ell,m)}\ (\mathrm{B}).$$

In fact:

Lemma 2.9. *Let Φ be a set of universal formulas and V be a non-empty set of literals. Then $V^{\Phi} = \{m : \Phi \vdash_{\mathrm{BTA}} \forall(\ell, m)$ for some $\ell \in V\}$.*

Proof. Suppose $m \in V^{\Phi}$; we show that $\Phi \vdash_{\mathrm{BTA}} \forall(\ell, m)$ for some $\ell \in V$. By definition, either $\ell \Rightarrow_{\Phi} m$ for some $\ell \in V$, or $\bar{m} \Rightarrow_{\Phi} m$. In the former case, we have shown above that $\Phi \vdash_{\mathrm{BTA}} \forall(\ell, m)$. In the latter case, we have shown that $\Phi \vdash_{\mathrm{BTA}} \forall(\bar{m}, m)$, and, picking *any* $\ell \in V$, we have the derivation

$$\frac{\begin{array}{c}\vdots\\ \forall(\bar{m},m)\end{array}}{\forall(\ell,m)}\ (\mathrm{A}).$$

The argument in the other direction is likewise a simple induction on the lengths of proofs. □

Lemma 2.10. *Let Φ be a set of universal formulas and V a set of literals. If V^{Φ} is inconsistent, then there exist literals $\ell, \ell' \in V$ such that $\Phi \vdash_{\mathrm{BTA}} \forall(\ell, \bar{\ell}')$.*

Proof. If $m, \bar{m} \in V^{\Phi}$, pick $\ell, \ell' \in V$ with $\Phi \vdash_{\mathrm{BTA}} \forall(\ell, m)$ and $\Phi \vdash_{\mathrm{BTA}} \forall(\ell', \bar{m})$. Re-writing $\forall(\ell', \bar{m})$ as $\forall(m, \bar{\ell}')$, we have a derivation from Φ

$$\frac{\begin{array}{c}\vdots\\ \forall(\ell,m)\end{array} \qquad \begin{array}{c}\vdots\\ \forall(m,\bar{\ell}')\end{array}}{\forall(\ell,\bar{\ell}')}\ (\mathrm{B}),$$

as claimed. □

Theorem 2.11. *The derivation relation $\vdash_{\mathsf{S}^+}$ is sound and complete for $\mathcal{S}^+$.*

Proof. Soundness is routine. To prove completeness, let Θ be a set of $\mathcal{S}^+$-formulas and θ an $\mathcal{S}^+$-formula, and suppose $\Theta \models \theta$. By the compactness theorem for first-order logic, we may safely assume that Θ is finite. Suppose for the moment that Θ is satisfiable, and write $\Theta = \Phi \cup \Psi$, where Φ is a set of universal formulas and Ψ a set of existential formulas. We consider three cases: (1) θ is universal; (2) θ is existential and Ψ is non-empty; and (3) θ is existential and Ψ is empty.

In the remainder of this proof, we simplify our notation to write $\vdash$ for $\vdash_{\mathsf{S}}$.

Case (1): Write $\theta = \forall(\ell_0, m_0)$. By Lemma 2.8, Part 1, $\Phi \models \theta$. Let

$$V_0 = \{\ell_0, \bar{m}_0\}^{\Phi}.$$

Thus, V_0 is non-empty and Φ-closed. We claim that V_0 is inconsistent. For suppose otherwise. By Lemma 2.1, let V be a consistent complete extension of V_0, and define $\mathfrak{A}$ to be the structure with singleton domain $\{a\}$ given by

$$p^{\mathfrak{A}} = \begin{cases} \{a\} \text{ if } p \in V \\ \emptyset \text{ otherwise,} \end{cases}$$

for every atom p. It is easily seen that $\mathfrak{A} \models \Phi \cup \bar{\theta}$, a contradiction. So by Lemma 2.10, there exist literals $\ell, \ell' \in \{\ell_0, \bar{m}_0\}$ such that

$$\Phi \vdash_{\text{BTA}} \forall(\ell, \bar{\ell}'). \tag{2.10}$$

By exchanging ℓ and ℓ' if necessary, we have two sub-cases: (i) $\ell = \ell_0$ and $\ell' = \bar{m}_0$; (ii) $\ell = \ell' \in \{\ell_0, \bar{m}_0\}$. In sub-case (i), (2.10) simply asserts that $\Phi \vdash \theta$. In sub-case (ii), we have one of the derivations from Φ

$$\cfrac{\begin{matrix}\vdots\\ \forall(\ell_0, \bar{\ell}_0)\end{matrix}}{\forall(\bar{m}_0, \bar{\ell}_0)}\ (\text{A}) \qquad\qquad \cfrac{\begin{matrix}\vdots\\ \forall(\bar{m}_0, m_0)\end{matrix}}{\forall(\ell_0, m_0)}\ (\text{A}),$$

and so $\Phi \vdash \theta$.

Case (2): Write $\theta = \exists(\ell, m)$. By Lemma 2.8, Part 2, there exists $\psi = \exists(\ell_0, m_0) \in \Psi$ such that $\Phi \cup \{\psi\} \models \theta$. Set

$$V_0 = \{\ell_0, m_0, \bar{\ell}\}^{\Phi}.$$

The set V_0 must be inconsistent. For otherwise, we can easily construct, using a parallel argument to that employed in Case (1), a structure $\mathfrak{A}$ such that $\mathfrak{A} \models \Phi \cup \{\psi, \bar{\theta}\}$, contradicting the fact that $\Phi \cup \{\psi\} \models \theta$. Hence, there exist literals $\ell_1, \ell_2 \in \{\ell_0, m_0, \bar{\ell}\}$ such that $\Phi \vdash_{\text{BTA}} \forall(\ell_1, \bar{\ell}_2)$. If ℓ_1 and ℓ_2 are both in $\{\ell_0, m_0\}$, then Θ is unsatisfiable, contrary to hypothesis. So assume, without loss of generality, that $\ell_2 = \bar{\ell}$. Thus, $\Phi \vdash_{\text{BTA}} \forall(\ell_1, \ell)$, and we have the following possibilities: (i) $\ell_1 = \ell_0$; (ii) $\ell_1 = m_0$; (iii) $\ell_1 = \bar{\ell}$. Possibility (i) yields $\Phi \vdash \forall(\ell_0, \ell)$. Possibility (ii) yields $\Phi \vdash \forall(m_0, \ell)$. Possibility (iii) also yields $\Phi \vdash \forall(\ell_0, \ell)$, via the derivation

$$\cfrac{\begin{matrix}\vdots\\ \forall(\bar{\ell}, \ell)\end{matrix}}{\forall(\ell_0, \ell)}\ (\text{A}).$$

In other words, we have proved:

$$\text{either } \Phi \vdash \forall(\ell_0, \ell) \text{ or } \Phi \vdash \forall(m_0, \ell). \tag{2.11}$$

Replacing ℓ by m in the above argument yields, in exactly the same way:

$$\text{either } \Phi \vdash \forall(\ell_0, m) \text{ or } \Phi \vdash \forall(m_0, m). \tag{2.12}$$

Considering (2.11), we may assume, by transposing ℓ_0 and m_0 if necessary, that $\Phi \vdash \forall(\ell_0, \ell)$. This leaves us with the two possibilities in (2.12). If $\Phi \vdash \forall(m_0, m)$, we have

$$\dfrac{\dfrac{\exists(\ell_0, m_0) \qquad \overset{\vdots}{\forall(\ell_0, \ell)}}{\exists(\ell, m_0)}\,(\mathrm{D}) \qquad \overset{\vdots}{\forall(m_0, m)}}{\exists(\ell, m)}\,(\mathrm{D});$$

if, on the other hand, $\Phi \vdash \forall(\ell_0, m)$, we have

$$\dfrac{\dfrac{\dfrac{\exists(\ell_0, m_0)}{\exists(\ell_0, \ell_0)}\,(\mathrm{I}) \qquad \overset{\vdots}{\forall(\ell_0, \ell)}}{\exists(\ell, \ell_0)}\,(\mathrm{D}) \qquad \overset{\vdots}{\forall(\ell_0, m)}}{\exists(\ell, m)}\,(\mathrm{D}).$$

Either way, $\Theta \vdash \theta$, as required.

Case (3): Write $\theta = \exists(\ell, m)$. Since $\Theta = \Phi \models \theta$, by Lemma 2.8, Part 3, $\Phi \models \forall(\bar{\ell}, \ell)$ and $\Phi \models \forall(\bar{m}, m)$. By Case (1), $\Phi \vdash \forall(\bar{\ell}, \ell)$ and $\Phi \vdash \forall(\bar{m}, m)$. Therefore, we have the derivation from Φ

$$\dfrac{\dfrac{\overset{\vdots}{\forall(\bar{\ell}, \ell)}}{\exists(\ell, \ell)}\,(\mathrm{N}) \qquad \dfrac{\overset{\vdots}{\forall(\bar{m}, m)}}{\forall(\ell, m)}\,(\mathrm{A})}{\exists(\ell, m)}\,(\mathrm{D}),$$

and $\Theta \vdash \theta$.

We have now shown that, for Θ satisfiable, $\Theta \models \theta$ implies $\Theta \vdash \theta$. It remains only to consider the case where Θ is unsatisfiable. If so, let $\Theta' \cup \{\theta'\}$ be a minimal unsatisfiable subset of Θ. (Remember, we are allowed to assume that Θ is finite.) Hence, Θ' is satisfiable, with $\Theta' \models \bar{\theta}'$. By the previous argument, $\Theta' \vdash \bar{\theta}'$. Thus, we have the derivation from $\Theta' \cup \{\theta'\}$

$$\dfrac{\overset{\vdots}{\bar{\theta}'} \qquad \theta'}{\theta}\,(\mathrm{X}),$$

and $\Theta \vdash \theta$. □

Theorem 2.6 shows that, with a little good will in the form of rules (A), (I), (N), (T) and (X)—which may arguably be said to treat ‘corner cases’—the Aristotelian syllogisms here subsumed under the rules (B) and (D) form a sound and complete inference system for the extended classical syllogistic.

2.3 The relational syllogistic

We mentioned in the introduction to this chapter that the classical syllogistic cannot capture inferences that are intrinsically relational in character, and we proposed the *relational syllogistic* to fill this lacuna. In this section we provide a formal definition of this language, and use the techniques established above to analyse its satisfiability problem.

By the *relational syllogistic*, we understand the set $\mathcal{R}$ of first-order formulas either belonging to the classical syllogistic or having any of the forms

$$\begin{array}{ll}
\forall x(\alpha \rightarrow \exists y(\beta \wedge r(x,y))) & \forall x(\alpha \rightarrow \exists y(\beta \wedge \neg r(x,y))) \\
\exists x(\alpha \wedge \exists y(\beta \wedge r(x,y))) & \exists x(\alpha \wedge \exists y(\beta \wedge \neg r(x,y))) \\
\forall x(\alpha \rightarrow \forall y(\beta \rightarrow r(x,y))) & \forall x(\alpha \rightarrow \forall y(\beta \rightarrow \neg r(x,y))) \\
\exists x(\alpha \wedge \forall y(\beta \rightarrow r(x,y))) & \exists x(\alpha \wedge \forall y(\beta \rightarrow \neg r(x,y)))
\end{array} \tag{2.13}$$

where α is an atomic formula $p(x)$ and β is an atomic formula $q(y)$. The *extended relational syllogistic* is the set $\mathcal{R}^+$ of first-order formulas either belonging to the *extended* classical syllogistic or having any of the forms (2.13), where α is a *literal* $p(x)$ or $\neg p(x)$, and β a *literal* $q(y)$ or $\neg q(y)$. Thus, the extended relational syllogistic is the relational syllogistic with added ‘noun-level negation’.

Since the variables and Boolean connectives here convey no additional information, we may again drop them. Let us use the term *binary atom* as a synonym for a *binary predicate*, and *binary literal* for an expression of either of the forms r or $\bar{r}$, where r is a binary atom. Define a *c-term* to be an expression of any of the forms

$$\ell, \qquad \exists(p,t), \qquad \forall(p,t).$$

where ℓ is a unary literal, p a unary predicate and t a binary literal. With these conventions, we may regard $\mathcal{R}$ as the set of expressions

$$\exists(p,c), \qquad \exists(c,p), \qquad \forall(p,c), \qquad \forall(c,\bar{p}),$$

where p is a unary atom and c a c-term.

We gloss (non-literal) c-terms using complex noun phrases, as follows:

$\exists(q,r)$	thing which r’s some q	$\forall(q,r)$	thing which r’s every q
$\exists(q,\bar{r})$	thing which does not r every q	$\forall(q,\bar{r})$	thing which r’s no q.

And we gloss $\mathcal{R}$-formulas involving such c-terms accordingly, thus:

$\forall(p,\exists(q,r))$	Every p r’s some q	$\forall(p,\exists(q,\bar{r}))$	No p r’s every q
$\exists(p,\exists(q,r))$	Some p r’s some q	$\exists(p,\exists(q,\bar{r}))$	Some p does not r every q
$\forall(p,\forall(q,r))$	Every p r’s every q	$\forall(p,\forall(q,\bar{r}))$	No p r’s any q
$\exists(p,\forall(q,r))$	Some p r’s every q	$\exists(p,\forall(q,\bar{r}))$	Some p r’s no q.

In these glosses, quantifiers in subjects are assumed to have wide scope; those in objects, narrow scope. Also, the "not" in "Some p does not r every q" is assumed to scope over the direct object. (Equivalently: there is some p such that there is some q which it doesn't r.) Again, we shall employ the variable-free notation for $\mathcal{R}$ in the sequel, because it is more compact.

The main task of this section is to show that the satisfiability problem for $\mathcal{R}$ remains in NLogSpace, and hence is NLogSpace-complete (because $\mathcal{R}$ contains $\mathcal{S}$ as a sub-fragment). However, the complexity bound for $\mathcal{R}$ is much less obvious than that for $\mathcal{S}^+$ obtained in Corollary 2.4. Our strategy employs a system of syllogisms for determining satisfiability in $\mathcal{R}$; by analysing the structure of deductions in this system, we obtain the desired complexity bound. Unfortunately—and somewhat surprisingly—the exact analogue of Theorem 2.11 fails for $\mathcal{R}$: there is *no* sound and complete set of syllogisms for $\mathcal{R}$, as there is for $\mathcal{S}$. Indeed, the proof of this fact is relatively easy.

Theorem 2.12. *There exists no finite set* X *of syllogisms in* $\mathcal{R}$ *such that* $\vdash_{\mathsf{X}}$ *is both sound and complete.*

Proof. Let X be any finite set of syllogisms for $\mathcal{R}$, and suppose $\vdash_{\mathsf{X}}$ is sound. We show that it is not complete. Since X is finite, fix $n \in \mathbb{N}$ greater than the number of antecedents in any rule in X.

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

$$\forall(p_i, \exists(p_{i+1}, r)) \qquad (1 \leq i < n) \qquad (2.14)$$

$$\forall(p_1, \forall(p_n, r)) \qquad (2.15)$$

$$\forall(p, p) \qquad (p \text{ a unary atom}) \qquad (2.16)$$

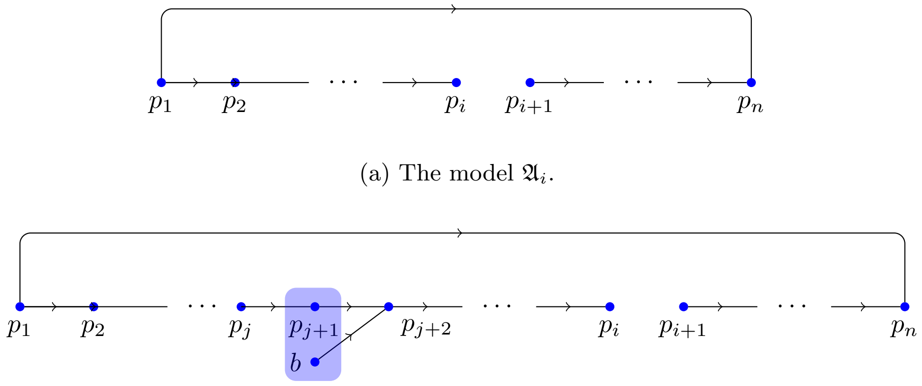
$$\forall(p_i, \bar{p}_j) \qquad (1 \leq i < j \leq n) \qquad (2.17)$$

and let γ be the $\mathcal{R}$-formula $\forall(p_1, \exists(p_n, r))$. Observe that, in (2.16), p ranges over *all* unary atoms whatever, so Γ is in fact infinite. Evidently, $\Gamma \models \gamma$. For let $\mathfrak{A} \models \Gamma$. If $p_1^{\mathfrak{A}} = \emptyset$, then trivially $\mathfrak{A} \models \gamma$; on the other hand, if $p_1^{\mathfrak{A}} \neq \emptyset$, a simple induction using formulas (2.14) shows that $p_i^{\mathfrak{A}} \neq \emptyset$ for all i $(1 \leq i \leq n)$, whence $\mathfrak{A} \models \gamma$ by (2.15). For $1 \leq i < n$, let $\Delta_i = \Gamma \setminus \{\forall(p_i, \exists(p_{i+1}, r))\}$.

Claim 2.13. *If* $\varphi \in \mathcal{R}$ *and* $\Delta_i \models \varphi$, *then* $\varphi \in \Gamma$.

It follows from this claim that $\Gamma \not\vdash_{\mathsf{X}} \gamma$. For, since no rule of X has more than $n-1$ antecedents, any instance of those antecedents contained in Γ must be contained in Δ_i for some i. Let δ be the corresponding instance of the consequent of that rule. Since $\vdash_{\mathsf{X}}$ is sound, $\Delta_i \models \delta$. By Claim 2.13, $\delta \in \Gamma$. By induction on the number of steps in derivations, we see that no derivation from Γ leads to a formula not in Γ. But $\gamma \notin \Gamma$.

Proof of Claim 2.13. Certainly, Δ_i has a model, for instance the model $\mathfrak{A}_i$ given in Fig. 2.1a). Here, $A = \{p_1, \ldots, p_n\}$, $p_j^{\mathfrak{A}_i} = \{p_j\}$ for all j $(1 \leq j \leq n)$, and $r^{\mathfrak{A}_i}$ is indicated by the arrows. All other atoms (unary or binary) are assumed



(a) The model $\mathfrak{A}_i$.

(b) The model $\mathfrak{B}_{i,j}$, case $(j+2<i)$.

Figure 2.1: Two models of Δ_i (proof of Claim 2.13): arrows indicate the relation r.

to have empty extensions. Note that there is no arrow from p_i to p_{i+1}. We consider the various possibilities for φ in turn and check that either $\varphi \in \Gamma$ or there is a model of Δ_i in which φ is false.

(i) φ is of the form $\forall(p,p)$. Then $\varphi \in \Gamma$, by (2.16).

(ii) φ is not of the form $\forall(p,p)$, and involves at least one unary or binary atom other than $p_1,\ldots,p_n,r$. In this case, it is straightforward to modify $\mathfrak{A}_i$ so as to obtain a model $\mathfrak{A}'_i$ of Δ_i such that $\mathfrak{A}'_i \not\models \varphi$. Henceforth, then, we may assume that φ involves no atoms other than $p_1,\ldots,p_n,r$.

(iii) φ is of the form $\forall(p_j,p_k)$. If $j=k$, then $\varphi \in \Gamma$, by (2.16). If $j \neq k$, then $\mathfrak{A}_i \not\models \varphi$, since $p_j^{\mathfrak{A}_i} \not\subseteq p_k^{\mathfrak{A}_i}$.

(iv) φ is of the form $\forall(p_j,\bar{p}_k)$. If $j=k$, then $\mathfrak{A}_i \not\models \varphi$, since $p_j^{\mathfrak{A}_i} \neq \emptyset$. If $j \neq k$, then $\varphi \in \Gamma$, by (2.17) and the identification $\forall(p_j,\bar{p}_k) = \forall(p_k,\bar{p}_j)$.

(v) φ is of the form $\forall(p_j,\forall(p_k,r))$. If $j=1$ and $k=n$, then $\varphi \in \Gamma$, by (2.15). So we may assume that either $j>1$ or $k<n$, in which case, $k \neq j+1$ implies $\mathfrak{A}_i \not\models \varphi$, by inspection of $\mathfrak{A}_i$ in Fig. 2.1(a). Hence, we may assume that $\varphi = \forall(p_j,\forall(p_{j+1},r))$, with $j<n$. Let $\mathfrak{B}_{i,j}$ be the structure obtained from $\mathfrak{A}_i$ by adding a second point b to the interpretation of p_{j+1}, and to which p_j is not related by r. The case where $j+2<i$ is shown in Fig. 2.1(b); similar pictures are possible in all other cases. By inspection, $\mathfrak{B}_{i,j} \models \Delta_i$, but $\mathfrak{B}_{i,j} \not\models \varphi$.

(vi) φ is of the form $\forall(p_j,\exists(p_k,r))$. If $k=j+1$, then $\varphi \in \Gamma$, by (2.14). Moreover, if $k \neq j+1$, then, unless $j=1$ and $k=n$, $\mathfrak{A}_i \not\models \varphi$, by inspection of Fig. 2.1(a). Hence we may assume $\varphi = \forall(p_1,\exists(p_n,r))$. Let $\mathfrak{C}_i = \mathfrak{A}_i \restriction \{p_1,\ldots,p_i\}$ be the structure obtained by restricting $\mathfrak{A}_i$ to the elements $p_1,\ldots,p_i$. In particular, $p_j^{\mathfrak{C}_i} = \emptyset$ for all j $(i<j\leq n)$. Then $\mathfrak{C}_i \models \Delta_i$, but $\mathfrak{C}_i \not\models \varphi$.

(vii) φ is of either of the forms $\forall(p_j, \forall(p_k, \bar{r}))$, $\forall(p_j, \exists(p_k, \bar{r}))$. Define $\mathfrak{A}''_i$ to be like $\mathfrak{A}_i$ except that $r^{\mathfrak{A}''_i}$ additionally contains the pair of points $\langle p_j, p_k \rangle$. It is then obvious that $\mathfrak{A}''_i \models \Delta_i$, but $\mathfrak{A}''_i \not\models \varphi$.

(viii) φ is of the form $\exists(p, c)$. Let $\mathfrak{A}_0$ be a structure over any domain in which every atom has empty extension. Then $\mathfrak{A}_0 \models \Delta_i$, but $\mathfrak{A}_0 \not\models \varphi$. □

This also completes the proof of Theorem 2.12. □

This negative result notwithstanding, we can rescue the situation by relaxing the notion of completeness. Observe that any formula of the form $\exists(p, \bar{p})$ is unsatisfiable. We refer to such a formula as an *absurdity*, and we write $\perp$ to stand (ambiguously) for any absurdity (i.e. different occurrences of $\perp$ may denote different absurdities). Say that a proof relation $\vdash_{\mathsf{X}}$ is *refutation complete* if, whenever Φ is unsatisfiable, $\Phi \vdash_{\mathsf{X}} \perp$, for some absurdity $\perp$.

We remind the reader that p and q range over unary atoms, c over c-terms, and t over binary literals. Let R be the following set of syllogisms:

$$\frac{\exists(p,q) \qquad \forall(q,c)}{\exists(p,c)}\ (\text{D1}) \qquad\qquad \frac{\forall(p,q) \qquad \forall(q,c)}{\forall(p,c)}\ (\text{B})$$

$$\frac{\forall(p,q) \qquad \exists(p,c)}{\exists(q,c)}\ (\text{D2}) \qquad\qquad \frac{}{\forall(p,p)}\ (\text{T}) \qquad \frac{\exists(p,c)}{\exists(p,p)}\ (\text{I})$$

$$\frac{\forall(q,\bar{c}) \qquad \exists(p,c)}{\exists(p,\bar{q})}\ (\text{D3}) \qquad\qquad \frac{\forall(p,\bar{p})}{\forall(p,c)}\ (\text{A}) \qquad \frac{\exists(p,\exists(q,t))}{\exists(q,q)}\ (\text{II})$$

$$\frac{\forall(p,\forall(q',t)) \qquad \exists(q,q')}{\forall(p,\exists(q,t))}\ (\forall\forall) \qquad\qquad \frac{\exists(p,\exists(q,t)) \qquad \forall(q,q')}{\exists(p,\exists(q',t))}\ (\exists\exists)$$

$$\frac{\forall(p,\exists(q,t)) \qquad \forall(q,q')}{\forall(p,\exists(q',t))}\ (\forall\exists).$$

Rules (D1), (D2), (D3), (B), (A), (T) and (I) are natural variants of their namesakes in the system $\mathsf{S}^\dagger$. By contrast, $(\forall\forall)$, $(\exists\exists)$, $(\forall\exists)$ and (II) express genuinely relational logical principles. Because we seek only refutation-completeness for $\vdash_{\mathsf{R}}$, we do not need a version of the rule (X).

To illustrate these rules, let n be any integer greater than 1, let

$$\Gamma^* = \{\forall(p_i, \exists(p_{i+1}, r)) \mid 1 \leq i < n\} \cup \{\forall(p_1, \forall(p_n, r))\}$$

and let $\gamma = \forall(p_1, \exists(p_n, r))$. Noting that $\bar{\gamma} = \exists(p_1, \forall(p_n, \bar{r}))$, we have the derivation (shown here for $n > 3$)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\exists(p_1,\forall(p_n,\bar{r}))}{\exists(p_1,p_1)}\;(\text{I}) \qquad \forall(p_1,\exists(p_2,r))}{\exists(p_1,\exists(p_2,r))}\;(\text{D1})}{\exists(p_2,p_2)}\;(\text{II}) \qquad \forall(p_2,\exists(p_3,r))}{\exists(p_2,\exists(p_3,r))}\;(\text{D1})}{\exists(p_3,p_3)}\;(\text{II})$$

$$\ddots$$

$$\dfrac{\dfrac{\forall(p_1,\forall(p_n,r)) \qquad \exists(p_n,p_n)}{\forall(p_1,\exists(p_n,r))}\;(\forall\forall) \qquad \exists(p_1,\forall(p_n,\bar{r}))}{\exists(p_1,\bar{p}_1)}\;(\text{D3}),$$

showing that $\Gamma^* \cup \{\bar{\gamma}\} \vdash_{\mathsf{R}} \bot$. By contrast, since $\Gamma^* \subseteq \Gamma$, where Γ is the set of formulas used in the proof of Theorem 2.12, we know that, for *any* finite set X of syllogisms, n can be made sufficiently large that $\Gamma^* \not\vdash_{\mathsf{X}} \gamma$.

Our next task is to show that the rules R are sound and refutation complete. In doing so, our proof will yield, as a by-product, an upper complexity bound for the problem $\text{Sat}(\mathcal{R})$. For the remainder of this section, fix a finite, non-empty set Γ of $\mathcal{R}$-formulas. As usual, we take the (possibly decorated) variables p, q to range over unary atoms, r over binary atoms, t over binary literals, and c, d over c-terms. We write $c \Rightarrow d$ if $c = d$ or there exists a sequence of unary atoms $p_0, \ldots, p_k$ such that $c = p_0$, $\forall(p_k, d) \in \Gamma$, and $\forall(p_i, p_{i+1}) \in \Gamma$ for all i $(0 \leq i < k)$. If V is a set of c-terms, write $V \Rightarrow d$ if $c \Rightarrow d$ for some $c \in V$.

Lemma 2.14. *Let V be a set of c-terms.*

1. *If $V \Rightarrow c$, then either $c \in V$ or there exists $p \in V$ such that $\Gamma \vdash_{\mathsf{R}} \forall(p, c)$;*
2. *if $V \Rightarrow p$, then there exists $p_0 \in V$ such that $\Gamma \vdash_{\mathsf{R}} \forall(p_0, p)$;*
3. *if $p \Rightarrow c$, then $\Gamma \vdash_{\mathsf{R}} \forall(p, c)$.*

Proof. Immediate, noting that R contains the rules (B) and (T). □

In the ensuing lemmas, we show that, if Γ is consistent (with respect to $\vdash_{\mathsf{R}}$), then Γ is satisfiable. As a first step, we create plenty of objects from which to construct a potential model of Γ. If $0 \leq i \leq 2$ and V is a set of c-terms with $1 \leq |V| \leq 2$, let $b_{V,i}$ denote some object or other, and assume that the various $b_{V,i}$ are distinct. Now set

$$B_0 = \{b_{\{p,c\},0} \mid \exists(p, c) \in \Gamma\}.$$

Lemma 2.15. *Let $b_{V,0} \in B_0$, and let $p, c \in V$. Then $\Gamma \vdash_{\mathsf{R}} \exists(p, c)$.*

Proof. If $p \neq c$, then $V = \{p, c\}$ and $\exists(p, c) \in \Gamma$ by construction. If $p = c$, then $V = \{p, d\}$ for some d, and we have the derivation

$$\frac{\exists(p, d)}{\exists(p, p)}\ \text{(I)}.$$

□

We now define sets B_1, B_2, ... inductively as follows. Suppose B_k has been defined. Let

$$\begin{aligned} B_{k+1} = B_k \cup \{b_{\{p\},i} \mid 1 \leq i \leq 2 \text{ and}, \\ \text{for some } b_{V,j} \in B_k \text{ and some } t,\ V \Rightarrow \exists(p, t)\}. \end{aligned}$$

Let $B = \bigcup_{0 \leq k} B_k$. Evidently, B is finite. (Indeed, $|B|$ is bounded by a linear function of $|\bar{\Gamma}|$.) It is immediate from the construction of B that, if $b_{V,i} \in B_0$, then $1 \leq |V| \leq 2$, V contains at least one unary atom p, and $i = 0$. On the other hand, if $b_{V,i} \in B_k$ for $k > 0$, then $V = \{p\}$ for some unary atom p, and i is either 1 or 2. The intuition here is that the elements of B_0 are witnesses for the existential formulas of Γ, while the elements of B_{k+1} are the witnesses for existential c-terms satisfied by elements of B_k.

Lemma 2.16. *If $b_{V,i} \in B$, $V \Rightarrow p$ and $V \Rightarrow c$, then $\Gamma \vdash_{\mathsf{R}} \exists(p, c)$.*

Proof. Let k be the smallest number such that $b_{V,i} \in B_k$. We proceed by induction on k.

For the case $k = 0$, we have $b_{V,i} = b_{V,0} \in B_0$. By Lemma 2.14 Part 2, there exists $q_1 \in V$ such that $\Gamma \vdash_{\mathsf{R}} \forall(q_1, p)$. By Lemma 2.14 Part 1, either $c \in V$ or there exists $q_2 \in V$ such that $\Gamma \vdash_{\mathsf{R}} \forall(q_2, c)$. In the former case, Lemma 2.15 yields $\Gamma \vdash_{\mathsf{R}} \exists(q_1, c)$, so that we have the derivation

$$\frac{\begin{matrix}\vdots \\ \exists(q_1, c)\end{matrix} \qquad \begin{matrix}\vdots \\ \forall(q_1, p)\end{matrix}}{\exists(p, c)}\ \text{(D2)}.$$

In the latter case, Lemma 2.15 yields $\Gamma \vdash_{\mathsf{R}} \exists(q_1, q_2)$, so that we have the derivation

$$\frac{\dfrac{\begin{matrix}\vdots \\ \exists(q_1, q_2)\end{matrix} \qquad \begin{matrix}\vdots \\ \forall(q_2, c)\end{matrix}}{\exists(q_1, c)}\ \text{(D1)} \qquad \begin{matrix}\vdots \\ \forall(q_1, p)\end{matrix}}{\exists(p, c)}\ \text{(D2)}.$$

For the case $k > 0$, $b_{V,i} \in B_k$ implies $V = \{p_k\}$ for some p_k, and $1 \leq i \leq 2$. By construction of B_k, there exist $b_{W,j} \in B_{k-1}$, $p_{k-1} \in W$ and binary literal t, such that $W \Rightarrow \exists(p_k, t)$. By inductive hypothesis, $\Gamma \vdash_{\mathsf{R}} \exists(p_{k-1}, \exists(p_k, t))$, and by Lemma 2.14 Part 3, $\Gamma \vdash_{\mathsf{R}} \forall(p_k, p)$, and $\Gamma \vdash_{\mathsf{R}} \forall(p_k, c)$. Therefore, we have the derivation

$$\dfrac{\dfrac{\dfrac{\overset{\vdots}{\exists(p_{k-1}, \exists(p_k, t))}}{\exists(p_k, p_k)}\,(\mathrm{II}) \qquad \overset{\vdots}{\forall(p_k, p)}}{\exists(p_k, p)}\,(\mathrm{D1}) \qquad \overset{\vdots}{\forall(p_k, c)}}{\exists(p, c)}\,(\mathrm{D1}).$$

□

Lemma 2.17. *If $b_{V,i} \in B$, $V \Rightarrow c$, $V \Rightarrow d$, and $c \neq d$, then there exists a unary atom p such that either:* (i) $\Gamma \vdash_{\mathsf{R}} \exists(p, c)$ *and* $\Gamma \vdash_{\mathsf{R}} \forall(p, d)$*; or* (ii) $\Gamma \vdash_{\mathsf{R}} \exists(p, d)$ *and* $\Gamma \vdash_{\mathsf{R}} \forall(p, c)$.

Proof. Suppose first that $c = q$ for some q. By Lemma 2.16, $\Gamma \vdash_{\mathsf{R}} \exists(q, d)$, and by rule (T), $\Gamma \vdash_{\mathsf{R}} \forall(q, q)$. Putting $p = q$ then satisfies Condition (ii). On the other hand, if $d = q$ for some q, then Condition (i) is satisfied, by a similar argument. Hence we may assume that neither c nor d is a unary atom. Since $c \neq d$, we have either $c \notin V$ or $d \notin V$, by construction of B. If the latter, then, by Lemma 2.14 Part 1, there exists $p \in V$ such that $\Gamma \vdash_{\mathsf{R}} \forall(p, d)$. But now we have $V \Rightarrow p$ and $V \Rightarrow c$, so that, by Lemma 2.16, $\Gamma \vdash_{\mathsf{R}} \exists(p, c)$, and Condition (i) is satisfied. If, on the other hand, $c \notin V$, Condition (ii) is satisfied, by a similar argument. □

The set B will form the domain of a structure $\mathfrak{B}$, defined as follows. If p is a unary atom, set

$$p^{\mathfrak{B}} = \{b_{V,i} \in B \mid V \Rightarrow p\};$$

and if r is a binary atom, set

$$\begin{aligned} r^{\mathfrak{B}} = \{\langle b_{V,i}, b_{\{p\},1}\rangle \in B^2 \mid V \Rightarrow \exists(p, r)\} \cup \\ \{\langle b_{V,i}, b_{W,j}\rangle \in B^2 \mid \text{ for some } q,\ V \Rightarrow \forall(q, r) \text{ and } W \Rightarrow q\}. \end{aligned}$$

The intuition is that the elements $b_{\{p\},1}$ are witnesses for the existential quantifiers in c-terms of the form $\exists(p, r)$, while the elements $b_{\{p\},2}$ are witnesses for the existential quantifiers in c-terms of the form $\exists(p, \bar{r})$.

Lemma 2.18. *If Γ is unsatisfiable, then there exist an element $b_{V,i}$ of B, a unary atom p and a c-term c, such that $V \Rightarrow p$, $V \Rightarrow c$, $b_{V,i} \in p^{\mathfrak{B}}$ and $b_{V,i} \notin c^{\mathfrak{B}}$.*

Proof. Since Γ is unsatisfiable, let $\varphi \in \Gamma$ be such that $\mathfrak{B} \not\models \varphi$. If $\varphi = \exists(p, c)$, let $V = \{p, c\}$. Trivially, $V \Rightarrow p$ and $V \Rightarrow c$. By construction of B, $b_{V,0} \in B$, and by construction of $\mathfrak{B}$, $b_{V,0} \in p^{\mathfrak{B}}$, whence (since $\mathfrak{B} \not\models \varphi$) $b_{V,0} \notin c^{\mathfrak{B}}$. If, on the other hand, $\varphi = \forall(p, c)$, there exists $b_{V,i} \in B$ such that $b_{V,i} \in p^{\mathfrak{B}}$ and $b_{V,i} \notin c^{\mathfrak{B}}$. By construction of $\mathfrak{B}$, $V \Rightarrow p$; and since $\forall(p, c) \in \Gamma$, we have $V \Rightarrow c$, as required. □

We now prove the main Lemma, from which both the complexity and the refutation-completeness results follow.

Lemma 2.19. *If Γ is unsatisfiable, then condition* ($\mathbf{C}$) *holds, given as follows.*

$$(\mathbf{C}) \qquad \left\{ \begin{array}{l} \textit{There exist elements } b_{V,i}, b_{W,j} \textit{ of } B, \textit{ unary atoms } q,\ o, \textit{ and} \\ \textit{a binary atom } r, \textit{ such that one of the following is true:} \\ \quad \textit{1. } V \Rightarrow q \textit{ and } V \Rightarrow \bar{q}; \\ \quad \textit{2. } V \Rightarrow \exists(q,\bar{r}),\ V \Rightarrow \forall(o,r), \textit{ and } q \Rightarrow o; \\ \quad \textit{3. } V \Rightarrow \forall(q,\bar{r}),\ V \Rightarrow \exists(o,r), \textit{ and } o \Rightarrow q; \\ \quad \textit{4. } V \Rightarrow \forall(q,\bar{r}),\ V \Rightarrow \forall(o,r),\ W \Rightarrow q, \textit{ and } W \Rightarrow o. \end{array} \right.$$

Proof. Let V, i, p and c be as in Lemma 2.18. We claim first that c cannot be of the form q, $\exists(q,r)$ or $\forall(q,r)$. For consider each possibility in turn.

(i) If $c = q$, then, by the construction of $\mathfrak{B}$, $V \Rightarrow c$ implies $b_{V,i} \in q^{\mathfrak{B}}$, contradicting $b_{V,i} \notin c^{\mathfrak{B}}$.

(ii) If $c = \exists(q,r)$, then, by the construction of $\mathfrak{B}$, $V \Rightarrow c$ implies $b_{\{q\},1} \in B$, $b_{\{q\},1} \in q^{\mathfrak{B}}$, and $\langle b_{V,i}, b_{\{q\},1} \rangle \in r^{\mathfrak{B}}$, whence $b_{V,i} \in \exists(q,r)^{\mathfrak{B}}$, contradicting $b \notin c^{\mathfrak{B}}$.

(iii) If $c = \forall(q,r)$, then, since $V \Rightarrow c$, the construction of $\mathfrak{B}$ ensures that, for any $b_{W,j} \in q^{\mathfrak{B}}$, we have $W \Rightarrow q$ and hence $\langle b_{V,i}, b_{W,j} \rangle \in r^{\mathfrak{B}}$. That is, $b_{V,i} \in \forall(q,r)^{\mathfrak{B}}$, contradicting $b \notin c^{\mathfrak{B}}$.

Therefore, c is of one of the forms $\bar{q}$, $\exists(q,\bar{r})$, or $\forall(q,\bar{r})$. We consider each possibility in turn, and show that one of the four cases of Condition ($\mathbf{C}$) holds.

(i) If $c = \bar{q}$, then $b_{V,i} \notin c^{\mathfrak{B}}$ means that $b_{V,i} \in q^{\mathfrak{B}}$, so that, by construction of $\mathfrak{B}$, $V \Rightarrow q$. But by assumption, $V \Rightarrow c$, and we have Case 1 of Condition ($\mathbf{C}$).

(ii) If $c = \exists(q,\bar{r})$, then, since $V \Rightarrow c$, we have, by the construction of $\mathfrak{B}$, $b_{\{q\},2} \in B$, and in fact $b_{\{q\},2} \in q^{\mathfrak{B}}$. Since $b_{V,i} \notin c^{\mathfrak{B}}$, we have $\langle b_{V,i}, b_{\{q\},2} \rangle \in r^{\mathfrak{B}}$. The construction of $\mathfrak{B}$ then guarantees that for some unary atom o, $V \Rightarrow \forall(o,r)$ and $q \Rightarrow o$; thus we have Case 2 of Condition ($\mathbf{C}$).

(iii) If $c = \forall(q,\bar{r})$, then $b_{V,i} \notin c^{\mathfrak{B}}$ implies that there exists $b_{W,j} \in B$ such that $b_{W,j} \in q^{\mathfrak{B}}$ and $\langle b_{V,i}, b_{W,j} \rangle \in r^{\mathfrak{B}}$. By construction of $\mathfrak{B}$, $W \Rightarrow q$, and, for some unary atom o, either

   (a) $V \Rightarrow \exists(o,r)$, $W = \{o\}$ and $j = 1$, or

   (b) $V \Rightarrow \forall(o,r)$ and $W \Rightarrow o$.

   But these yield Cases 3 and 4 of Condition ($\mathbf{C}$), respectively.

□

Lemma 2.20. *The following are equivalent:*

1. $\Gamma \vdash_{\mathsf{R}} \bot$*;*

2. Γ *is unsatisfiable;*

3. *Condition* (**C**) *of Lemma 2.19 holds.*

Proof. For the implication $1 \Rightarrow 2$, we observe that $\vdash_{\mathsf{R}}$ is obviously sound. The implication $2 \Rightarrow 3$ is Lemma 2.19. For the implication $3 \Rightarrow 1$, suppose Condition (**C**) of Lemma 2.19 holds. This condition has four cases: we consider each in turn, showing that $\Gamma \vdash_{\mathsf{R}} \exists(p, \bar{p})$ for some unary atom p.

(i) $V \Rightarrow q$ and $V \Rightarrow \bar{q}$. By Lemma 2.16 we immediately have $\Gamma \vdash_{\mathsf{R}} \exists(q, \bar{q})$.

(ii) $V \Rightarrow \exists(q, \bar{r})$, $V \Rightarrow \forall(o, r)$, $q \Rightarrow o$. By Lemma 2.14 Part 2 (or Part 3), $\Gamma \vdash_{\mathsf{R}} \forall(q, o)$, and by Lemma 2.17, there exists p such that either:

(a) $\Gamma \vdash_{\mathsf{R}} \exists(p, \exists(q, \bar{r}))$ and $\Gamma \vdash_{\mathsf{R}} \forall(p, \forall(o, r))$; or

(b) $\Gamma \vdash_{\mathsf{R}} \exists(p, \forall(o, r))$ and $\Gamma \vdash_{\mathsf{R}} \forall(p, \exists(q, \bar{r}))$.

In Case (a), we then have

$$\dfrac{\dfrac{\overset{\vdots}{\exists(p, \exists(q, \bar{r}))} \qquad \overset{\vdots}{\forall(q, o)}}{\exists(p, \exists(o, \bar{r}))}\,(\exists\exists) \qquad \overset{\vdots}{\forall(p, \forall(o, r))}}{\exists(p, \bar{p})}\,(\text{D3}),$$

while in Case (b), we have

$$\dfrac{\overset{\vdots}{\exists(p, \forall(o, r))} \qquad \dfrac{\overset{\vdots}{\forall(p, \exists(q, \bar{r}))} \qquad \overset{\vdots}{\forall(q, o)}}{\forall(p, \exists(o, \bar{r}))}\,(\forall\exists)}{\exists(p, \bar{p})}\,(\text{D3}).$$

(iii) $V \Rightarrow \forall(q, \bar{r})$, $V \Rightarrow \exists(o, r)$, $o \Rightarrow q$. By Lemma 2.14 Part 2 (or Part 3), $\Gamma \vdash_{\mathsf{R}} \forall(o, q)$, and by Lemma 2.17, there exists p such that either: (a) $\Gamma \vdash_{\mathsf{R}} \exists(p, \exists(o, r))$ and $\Gamma \vdash_{\mathsf{R}} \forall(p, \forall(q, \bar{r}))$; or (b) $\Gamma \vdash_{\mathsf{R}} \exists(p, \forall(q, \bar{r}))$ and $\Gamma \vdash_{\mathsf{R}} \forall(p, \exists(o, r))$. But then we can employ exactly the same derivation patterns as for Cases (ii)(a) and (ii)(b), respectively.

(iv) $V \Rightarrow \forall(q, \bar{r})$, $V \Rightarrow \forall(o, r)$, $W \Rightarrow q$, $W \Rightarrow o$. By Lemma 2.17, there exists p such that $\Gamma \vdash_{\mathsf{R}} \exists(p, \forall(p_1, u))$ and $\Gamma \vdash_{\mathsf{R}} \forall(p, \forall(p_2, \bar{u}))$, where u is either r or

$\bar{r}$, and p_1 and p_2 are q and o in some order. By Lemma 2.16, $\Gamma \vdash_{\mathsf{R}} \exists(o, q)$, i.e. $\Gamma \vdash_{\mathsf{R}} \exists(p_1, p_2)$. Thus we have the derivation

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ \forall(p, \forall(p_2, \bar{u}))\end{matrix} \qquad \begin{matrix}\vdots\\ \exists(p_1, p_2)\end{matrix}}{\forall(p, \exists(p_1, \bar{u}))}\,(\forall\forall) \qquad\qquad \begin{matrix}\vdots\\ \exists(p, \forall(p_1, u))\end{matrix}}{\exists(p, \bar{p})}\,(\mathrm{D3}).$$

□

Theorem 2.21. *The derivation relation $\vdash_{\mathsf{R}}$ is sound and refutation-complete for $\mathcal{R}$.*

Proof. Soundness is obvious. Refutation-completeness is the implication from 2 to 1 in Lemma 2.20. □

Lemma 2.20 is more than just an auxiliary result for proving Theorem 2.21: it yields both a small model property and an upper complexity bound for the relational syllogistic.

Corollary 2.22. *If Γ is a finite, satisfiable set of $\mathcal{R}$-formulas, then Γ is satisfiable over a domain of size $O(|\Gamma|)$*

Proof. The set B has cardinality $O(|\Gamma|)$. □

Theorem 2.23. *The problem* Sat($\mathcal{R}$) *is* NLogSpace*-complete.*

Proof. The lower bound is immediate from Lemma 2.5. For the upper bound, by Proposition 1.6, it suffices to show that the problem of determining the *un*satisfiability of a given finite set of $\mathcal{R}$-formulas is in NLogSpace. Let Γ be a finite set of $\mathcal{R}$-formulas. Let B and $\mathfrak{B}$ be as defined for Lemmas 2.15–2.20. Lemma 2.20 guarantees that Γ is unsatisfiable if and only if there exist $b_{S,i}, b_{T,j} \in B$, unary atoms q, o, and a binary atom r satisfying one of the four cases in Condition (**C**) of Lemma 2.19. Nondeterministically guess these V, W, i, j, q, o and r. This requires only logarithmic space, because only the indices of the relevant atoms need to be encoded, and the size of B is linear in the number of formulas in Γ. To check that $b_{V,i}, b_{W,j} \in B$ is essentially a graph-reachability problem, as are all the requirements in the four cases of Condition (**C**). Since REACHABILITY is in NLogSpace (Proposition 1.10), this proves the theorem. □

2.4 The extended relational syllogistic

We now turn to the extended relational syllogistic, $\mathcal{R}^{+}$, which adds 'noun-level negation' to the relational syllogistic, just as the extended classical syllogistic did for the classical syllogistic. Formally, we define an *e-term* to be an expression of any of the forms

$$\ell, \qquad \exists(\ell, t), \qquad \forall(\ell, t).$$

where ℓ is a unary literal and t a binary literal; and we define $\mathcal{R}^+$ to be the set of expressions

$$\exists(\ell, e), \qquad \exists(e, \ell), \qquad \forall(\ell, e), \qquad \forall(e, \ell), \tag{2.18}$$

where p is a unary atom and e a c-term. Formulas of $\mathcal{R}^+$ are given the expected semantics, and receive the obvious glosses, for example:

$$\exists(\bar{p}, \forall(\bar{q}, r)) \qquad \text{Some non-}p\ r\text{s every non-}q.$$

Of course, the syntax in (2.18) can be regarded as a shorthand notation for the obvious collection of first-order formulas. Thus, $\mathcal{R}^+$ too is a fragment of first-order logic.

We saw above that the satisfiability problems for $\mathcal{S}$, $\mathcal{S}^+$ and $\mathcal{R}$ are all NLogSpace-complete. Not so for $\mathcal{R}^+$. Our task in this section is to show that $\text{Sat}(\mathcal{R}^+)$ is in fact complete for ExpTime, the (much larger) class of languages accepted by deterministic Turing machines in exponential time. We begin with the upper bound.

Lemma 2.24. *The problem* $\text{Sat}(\mathcal{R}^+)$ *is in* ExpTime.

Proof. Let a finite set of $\mathcal{R}^+$-formulas Φ be given. For brevity, we use the phrase *in exponential time* to mean "in time bounded by an exponential function of $\|\Phi\|$", where $\|\Phi\|$ denotes the number of symbols in Φ. Denote by E the set of all the e-terms e mentioned in Φ, as well as their opposites, $\bar{e}$. Denote by P the set of pairs of (not necessarily distinct) unary literals $\{\ell, m\}$ from from E such that $\ell \neq \bar{m}$. Denote by V the set of all consistent complete subsets of E. As before, a set v of e-terms is *consistent* if $e \in v$ implies $\bar{e} \notin v$, and *complete* if, for all $e \in E$, at least one of e or $\bar{e}$ is in v. In the ensuing proof, p and q range, as usual, over unary atoms, ℓ and m over unary literals, r over binary atoms, t over binary literals, and e over e-terms. For any $v \in V$ and any binary literal t, define

$$W_{v,t} = \{\bar{m} \mid \forall(m, t) \in v\}.$$

Carry out the following procedure. Consider every subset $Q \subseteq P$ in turn, and, for each such Q, form a set V_Q as follows. Starting with V, remove any element v if any of the following apply for some ℓ, m and t:

1. $\{\ell, m\} \subseteq v$, and $\{\ell, m\} \notin Q$;
2. $\{\forall(\ell, t), \forall(m, \bar{t})\} \subseteq v$, and $\{\ell, m\} \in Q$;
3. $\{\ell, e\} \subseteq v$, and $\forall(\ell, \bar{e}) \in \Phi$.

Then remove remove any element v if, for any ℓ and t:

4. $\exists(\ell, t) \in v$ and there exists no $w \in V$ such that $\{\ell\} \cup W_{v,\bar{t}} \subseteq w$,

repeating this last step until no further elements can be removed. Let V_Q be the set of those v which remain. We refer to the above conditions 1–4 as 'removal rules'. Once V_Q has been computed, check that, for all $\exists(\ell, e) \in \Phi$, there exists $v \in V_Q$ such that $\{\ell, e\} \subseteq v$; if so return the value Y (for "Yes") and stop. Otherwise, continue with the next value of Q. If all subsets $Q \subseteq P$ have been tried, return the value N ("No") and stop.

As a guide to intuition, think of Q as a guess as to which pairs of unary literals are realized by the same element in some putative model of Φ. The set V_Q will then form a collection of types of element consistent with this guess. Evidently, $|E| \leq \|\Phi\|$, whence $|V| \leq 2^{\|\Phi\|}$; moreover, the number of possibilities for Q is evidently bounded by $2^{\|\Phi\|^2}$. It is easy to see that each of the removal rules can be applied in exponential time. Moreover, the last removal rule can be applied at most $|V|$ times in total. Thus, the computation of V_Q, and indeed the whole procedure, takes only exponential time. We must show that Y is returned if and only is Φ is satisfiable.

Suppose Φ is satisfiable, and let $\mathfrak{A} \models \Phi$. If $a \in A$, denote by $\mathrm{etp}^{\mathfrak{A}}[a]$, the *e-term-type* of a in $\mathfrak{A}$, defined by

$$\mathrm{etp}^{\mathfrak{A}}[a] = \{e \in E \mid a \in e^{\mathfrak{A}}\}.$$

Now let $Q = \{\{\ell, m\} \mid \ell^{\mathfrak{A}} \cap m^{\mathfrak{A}} \neq \emptyset\}$. We show that, if the procedure has not returned Y beforehand, it will do so when Q is considered. We first prove:

Claim 2.25. *For all $a \in A$, $\mathrm{etp}^{\mathfrak{A}}[a] \in V_Q$.*

Proof of claim. We consider each of the removal rules in turn.

Rule 1: By the choice of Q, $\mathrm{etp}^{\mathfrak{A}}[a]$ certainly cannot be removed by this rule.

Rule 2: Suppose $\{\forall(\ell, t), \forall(m, \bar{t})\} \subseteq \mathrm{etp}^{\mathfrak{A}}[a]$. By the semantics for $\mathcal{R}^+$, there exists no $b \in A$ such that $\{\ell, m\} \subseteq \mathrm{etp}^{\mathfrak{A}}[b]$, whence $\{\ell, m\} \notin Q$.

Rule 3: If $\mathfrak{A} \models \forall(\ell, \bar{e})$, there exists no a such that $\{\ell, e\} \subseteq \mathrm{etp}^{\mathfrak{A}}[a]$.

Rule 4: Suppose for contradiction that some application of this rule removes some element $v = \mathrm{etp}^{\mathfrak{A}}[a]$, and consider the *first* such a. By the semantics for $\mathcal{R}^+$, there exists $b \in A$ such that $\{\ell\} \cup W_{v,\bar{t}} \subseteq \mathrm{etp}^{\mathfrak{A}}[b]$, and, by assumption, $w = \mathrm{etp}^{\mathfrak{A}}[b]$ will not yet have been removed. But then the rule does not apply. □

Since $\mathfrak{A} \models \Phi$, there exists, for all $\exists(\ell, e) \in \Phi$, some $a \in A$ such that $a \in \ell^{\mathfrak{A}} \cap e^{\mathfrak{A}}$, and thus, by Claim 2.25, some $v = \mathrm{etp}^{\mathfrak{A}}[a] \in V_Q$ such that $\{\ell, e\} \subseteq v$. Hence, the procedure returns Y.

Conversely, suppose the procedure returns Y, and let Q be the subset of P such that Y was returned on examination of V_Q. Let $A = V_Q \times \{0, 1\}$. We define a structure $\mathfrak{A}$ on A by setting

$$p^{\mathfrak{A}} = \{(v, i) \mid p \in v,\ i \in \{0, 1\}\}$$

for any unary atom p occurring in Φ, and

$$\begin{aligned} t^{\mathfrak{A}} = &\{\langle (v,i),(w,0)\rangle \mid \exists(\ell,r) \in v,\ i \in \{0,1\} \text{ and } \{\ell\} \cup W_{v,\bar{r}} \subseteq w\} \cup \\ &\{\langle (v,i),(w,j)\rangle \mid \forall(\ell,r) \in v,\ \{i,j\} \subseteq \{0,1\} \text{ and } \ell \in w\} \end{aligned}$$

for any binary atom r occurring in Φ. The intuition here is that elements $(w,0)$ will serve as witnesses for realized e-terms of the form $\exists(\ell,r)$, and elements $(w,1)$ will serve as witnesses for realized e-terms of the form $\exists(\ell,\bar{r})$.

Claim 2.26. *If $a = (v,i) \in A$ and $e \in v$, then $a \in e^{\mathfrak{A}}$.*

Proof of claim. If e is a unary literal, this is immediate from the consistency of v and the construction of $\mathfrak{A}$. There are four remaining forms of e to consider. (i) If $e = \exists(\ell,r)$, then removal rule 4 ensures that there exists $w \in V_Q$ such that $\{\ell\} \cup W_{v,\bar{r}} \subseteq w$. Hence, setting $b = (w,0)$, the construction of $\mathfrak{A}$ ensures that $\langle a,b\rangle \in r^{\mathfrak{A}}$, whence $a \in e^{\mathfrak{A}}$ by the semantics of $\mathcal{R}^+$. (ii) If $e = \forall(\ell,r)$, then, for all $b = (w,j) \in A$, if $b \in \ell^{\mathfrak{A}}$ (so that $\ell \in w$), then $\langle a,b\rangle \in r^{\mathfrak{A}}$ by the construction of $\mathfrak{A}$, whence $a \in e^{\mathfrak{A}}$ by the semantics of $\mathcal{R}^+$. (iii) If $e = \exists(\ell,\bar{r})$, then removal rule 4 ensures that there exists $w \in V_Q$ such that $\{\ell\} \cup W_{v,r} \subseteq w$. Setting $b = (w,1)$, and noting that $W_{v,r} \subseteq w$, we see that, for any binary literal m, $\forall(m,r) \in w$ implies $\bar{m} \in w$, whence $m \notin w$, whence $\langle a,b\rangle \notin r^{\mathfrak{A}}$ by the construction of $\mathfrak{A}$. Hence, $a \in e^{\mathfrak{A}}$ by the semantics of $\mathcal{R}^+$. (iv) If $e = \forall(\ell,\bar{r})$, suppose there exists $b = (w,j) \in A$ with $b \in \ell^{\mathfrak{A}}$, and hence $\ell \in w$. We need to check that $\langle a,b\rangle \notin r^{\mathfrak{A}}$. First, since w is consistent, $\bar{\ell} \notin w$, and hence $W_{v,\bar{r}} \not\subseteq w$. Second, if, for some unary literal $m \in w$, then removal rule 1 ensures that $\{\ell,m\} \in Q$, whence removal rule 2 ensures that $\forall(m,r) \notin v$. Hence, by the construction of $\mathfrak{A}$, we have $\langle a,b\rangle \notin r^{\mathfrak{A}}$ as required. Therefore, $a \in e^{\mathfrak{A}}$ by the semantics of $\mathcal{R}^+$. □

It is now easy to see that $\mathfrak{A} \models \varphi$ for any $\varphi \in \Phi$. Indeed, if $\varphi = \exists(\ell,e)$, since the procedure returns Y, we have $\{\ell,e\} \subseteq v \in V$, whence, by two applications of Claim 2.26, $(v,0) \in \ell^{\mathfrak{A}} \cap e^{\mathfrak{A}}$. On the other hand, if $\varphi = \forall(\ell,e)$ and $(v,i) \in \ell^{\mathfrak{A}}$, then, as noted above, $\ell \in v$. Removal rule 3 then ensures that $e \in v$, and hence, by Claim 2.26, that $(v,i) \in e^{\mathfrak{A}}$. In either case, the semantics of $\mathcal{R}^+$ guarantee that $\mathfrak{A} \models \varphi$. □

We remark that the upper complexity bound obtained for $\text{Sat}(\mathcal{R}^+)$ has a different character to those obtained earlier for $\text{Sat}(\mathcal{S}^+)$ and $\text{Sat}(\mathcal{R})$. For $\mathcal{S}^+$ and $\mathcal{R}$, we (in effect) used syllogisms to build small—i.e. polynomial sized—models witness by witness; for $\mathcal{R}^+$, by contrast, we began with a large—i.e. exponential sized—structure and whittled it away defect by defect. One might wonder whether the bottom-up approach could be used in the case of $\mathcal{R}^+$ to get a more efficient decision procedure. We now show that this is not possible, by obtaining a matching lower complexity bound.

We begin with a lemma that will also prove useful in Ch. 3. It provides a good example of a typical lower complexity bound proof encountered in this area. We again make use of one of the standard results in complexity theory from Sec. 1.4: Proposition 1.11 states that ExpTime is exactly APSpace, the class of languages accepted by *alternating* Turing machines using polynomial space.

Lemma 2.27. *Let $\mathcal{L}_{\text{Alt}}$ be the set of first-order formulas of the forms*

$$\begin{array}{ll} \exists x.\alpha(x) & \forall x(\alpha(x) \rightarrow \exists y(r(x,y) \wedge \beta(y))) \\ \forall x.\gamma(x) & \forall x(\alpha(x) \rightarrow \forall y(r(x,y) \rightarrow \gamma(y))), \end{array}$$

where $\alpha(x)$ and $\beta(x)$ are conjunctions of unary literals, and $\gamma(x)$ is a disjunction of unary literals. The problem $\text{Sat}(\mathcal{L}_{\text{Alt}})$ *is* ExpTime*-hard.*

Proof. Let M be an alternating Turing machine, with alphabet A and set of states S, and running in space bound f, for some polynomial f. As usual, we take a *symbol* of M to be any element of A together with $\sqcup$ (blank) and $\triangleleft$ (start-of-tape). Recall that the run of M (on a particular input) is a tree of machine configurations, with the initial configuration as the root. Our strategy is to encode the run of M on some input string $\text{x} \in A^*$ by means of a set of $\mathcal{L}_{\text{Alt}}$-formulas, $\Phi_{M,\text{x}}$. We shall ensure that: (i) if $\Phi_{M,\text{x}}$ has a model, then M accepts x; (ii) if M accepts x, then $\Phi_{M,\text{x}}$ has a model. We ensure that, for fixed M, the construction requires only space $O(\log|\text{x}|)$. Thus there is a many-one log-space reduction from the language recognized by M to $\text{Sat}(\mathcal{L}_{\text{Alt}})$. This proves that $\text{Sat}(\mathcal{L}_{\text{Alt}})$ is APSpace-hard; the theorem follows by Proposition 1.11. We may assume without loss of generality that that the states encountered in runs of M alternate strictly between universal and existential, starting with a universal state, and that, for any state s and any symbol a, M has at most two transitions available. Let these transitions be arbitrarily assigned to the sets T_L (left) and T_R (right), so that for any state s and symbol a, T_L contains at most one transition $\langle s, a, \dots \rangle$, and similarly for T_R. Recall that a universal configuration is accepting just in case all its successors are, and an existential configuration is accepting just in case some successor is. Thus, a terminating configuration (with no enabled transitions) is accepting just in case it is universal.

The idea of the encoding is that accepting configurations will be represented by elements in models of $\Phi_{M,\text{x}}$. Write $n = |\text{x}|$ and $m = f(n)$, assuming, without loss of generality, that $m \geq n$. For every symbol $a \in A \cup \{\sqcup, \triangleleft\}$, every state $s \in S$ and every i $(0 \leq i < m)$, let a_i, h_i, and s be a unary predicates, and let r be a binary predicate. As a guide to intuition, read $a_i(x)$ as "Tape square i contains a in configuration x", $h_i(x)$ as "The head is over tape square i in the configuration x", and $s(x)$ as "The program state in configuration x is s". The set $\Phi_{M,\text{x}}$ will contain formulas stating that these predicates form partitions in the expected way. To state that, in any configuration x, the ith square contains exactly one symbol from A, we write, for all i $(0 \leq i < m)$,

$$\forall x \left(\bigvee_{a \in A \cup \{\sqcup, \triangleleft\}} a_i(x) \right) \wedge \bigwedge_{a,a' \in A \cup \{\sqcup, \triangleleft\}}^{a \neq a'} \forall x(\neg a_i(x) \vee \neg a'_i(x)). \tag{2.19}$$

Similarly, we can state that, in any configuration x, the head is over exactly one tape square and M is in exactly one state. To state that the initial configuration of M with $\mathrm{x} = \mathrm{x}^1 \cdots \mathrm{x}^n$ as input is an accepting configuration, we take $\Phi_{M,\mathrm{x}}$ to contain the formula

$$\begin{aligned}&\exists x(s_0(x) \wedge h_0(x) \wedge \\ &\qquad \lhd_0(x) \wedge \mathrm{x}^1_1(x) \wedge \cdots \wedge \mathrm{x}^n_n(x) \wedge \sqcup_{n+1}(x) \wedge \cdots \wedge \sqcup_m(x)),\end{aligned} \tag{2.20}$$

where s_0 is the initial state of M. (Remember that the zeroth tape-square always contains $\lhd$, and the tape always has an infinite tail of $\sqcup$s.)

Let L and R be unary predicates. We should read $L(x)$ as "x is a configuration in which any enabled transitions in T_L will be executed", and similarly for $R(x)$. Then $\Phi_{M,\mathrm{x}}$ will contain, for all existential states $s \in S$,

$$\forall x(s(x) \to (L(x) \vee R(x))) \tag{2.21}$$

all for all universal states $s \in S$,

$$\forall x(s(x) \to (L(x) \wedge R(x))). \tag{2.22}$$

Consider now any transition $\tau = \langle a, s, b, t, \delta\rangle \in T_L \cup T_R$, with meaning

> If the head is scanning a in state s, write b, displace the head by δ, and go to state t.

If $\tau \in T_L$, then, for each tape-square i ($0 \leq i < m$), $\Phi_{M,\mathrm{x}}$ will contain

$$\begin{aligned}&\forall x((a_i(x) \wedge h_i(x) \wedge s(x) \wedge L(x)) \to \\ &\qquad \exists y(b_i(y) \wedge h_{i+\delta}(y) \wedge t(y) \wedge r(x, y))),\end{aligned} \tag{2.23}$$

and if $\tau \in T_R$, $\Phi_{M,\mathrm{x}}$ will contain

$$\begin{aligned}&\forall x((a_i(x) \wedge h_i(x) \wedge s(x) \wedge R(x)) \to \\ &\qquad \exists y(b_i(y) \wedge h_{i+\delta}(y) \wedge t(y) \wedge r(x, y))).\end{aligned} \tag{2.24}$$

In addition, for each symbol a and each tape-square i ($0 \leq i < m$), $\Phi_{M,\mathrm{x}}$ will contain

$$\forall x((a_i(x) \wedge \neg h_i(x)) \to \forall y(r(x, y) \to a_i(y)) \tag{2.25}$$

to state that the contents of tape squares not being visited by the head remain unchanged. Finally, we add formulas ensuring that, for existential states, M is not left without an available transition. For each symbol a, each tape-square i ($0 \leq i < m$), and each *existential* state $s \in S$, $\Phi_{M,\mathrm{x}}$ will contain

$$\forall x(s(x) \wedge h_i(x) \wedge L(x) \to \bigvee\{a_i(x) \mid \langle s, a, t, b, \delta\rangle \in T_L\}) \tag{2.26}$$

$$\forall x(s(x) \wedge h_i(x) \wedge R(x) \to \bigvee\{a_i(x) \mid \langle s, a, t, b, \delta\rangle \in T_R\}). \tag{2.27}$$

This completes the definition of $\Phi_{M,\mathrm{x}}$. It is routine to check that the construction of this formula (for fixed M) requires working memory $O(\log |\mathrm{x}|)$.

Suppose $\mathfrak{A} \models \Phi_{M,\mathrm{x}}$, and let a_0 be a witness for (2.20). Evidently, a encodes the initial configuration of M with input x. We claim that, for all $d \geq 0$, $\mathfrak{A}$ contains a subset B_d such that $\mathfrak{A}\restriction B_d$ encodes an initial segment of a run of M of depth d, with the sets B_d nested by inclusion. Since, by assumption, all runs of M terminate, it follows that $\mathfrak{A}$ contains a finite subset B of elements such that $\mathfrak{A}\restriction B$ encodes the entire run of M on input x. We proceed by induction on d. The case $d = 0$ has been dealt with. Suppose now $a \in B_d$ encodes a configuration at level d. If that configuration is existential, then (2.21) ensures that either $\mathfrak{A} \models L[a]$ or $\mathfrak{A} \models R[a]$. By (2.23), (2.24) and (2.25), then, *either* for every (at most one) enabled transition $\tau \in T_L$, *or* for every (at most one) enabled transition $\tau \in T_R$, there exists a $b \in A$ with $\mathfrak{A} \models r[a, b]$, such that b encodes the result of executing τ. If the configuration is universal, then (2.22) ensures that both $\mathfrak{A} \models L[a]$ and $\mathfrak{A} \models R[a]$, so that, for every (at most two) enabled transition $\tau \in T_L \cup T_R$, there exists a b with $\mathfrak{A} \models r[a, b]$, such that b encodes the result of executing τ. Now let B_{d+1} be B_d together with all new elements b chosen in this way. This completes the induction. We need to show that the run of M in question is accepting. But this requires only that each existential state has at least one enabled transition, and that is ensured by (2.26) and (2.27).

Conversely, if M has an accepting run on input x, consider the set of configurations encountered in this run, and define a structure $\mathfrak{A}$ on this set by declaring $\mathfrak{A} \models r[a, b]$ if b is a daughter of a in the run, interpreting the unary predicates occurring in $\Phi_{M,\mathrm{x}}$ as indicated above. It is then a simple matter to check that $\mathfrak{A} \models \Phi_{M,\mathrm{x}}$. □

While on the subject of $\mathcal{L}_{\mathrm{Alt}}$, we take the opportunity to derive a simple corollary Lemma 2.27 that will prove useful in Ch. 5.

Corollary 2.28. *Let $\mathcal{L}_{\mathrm{AltVar}}$ be the set of first-order formulas of the forms $\exists x.\zeta$ and $\forall x \exists y.\eta$, where ζ is a quantifier- and equality-free formula with free variables $\{x\}$ and η a quantifier- and equality-free formula with free variables $\{x, y\}$, both over a signature of unary and binary predicates. The problem* Sat($\mathcal{L}_{\mathrm{AltVar}}$) *is* ExpTime*-hard. Indeed, the problem remains* ExpTime*-hard even under the restriction that the input contains at most* one *formula of the form $\exists x.\zeta$ and at most* two *formulas of the form $\forall x \exists y.\eta$.*

Proof. In the set of formulas encountered in the proof of Lemma 2.27 at most one is of the form $\exists x.\zeta$. Moreover, the remaining formulas are either of the form $\forall x.\zeta$ (purely universal in one variable), or are the formulas (2.23)–(2.25), recording the 'active' and 'passive' effects of some transition τ. Now, we can fold these latter into a single formula stating all effects—active and passive—of

τ. Thus, for each $\tau \in T_L$, and for each tape-square i ($0 \leq i < m$), we write

$$\forall x \exists y((a_i(x) \wedge h_i(x) \wedge s(x) \wedge L(x)) \rightarrow \\ [b_i(y) \wedge h_{i+\delta}(y) \wedge t(y) \wedge r(x,y) \wedge \bigwedge_{1 \leq j \leq m}^{j \neq i} (a_j(x) \rightarrow a_j(y))]. \quad (2.28)$$

This clearly has the same effect. If $\tau \in T_R$, we proceed similarly. Now all formulas are (modulo trivial logical manipulation) in the fragment $\mathcal{L}_{\mathrm{AltVar}}$.

The last statement follows by repeating this 'folding' trick for all the formulas in T_L and, separately, all the formulas in T_R. Write the formula for τ given in (2.28) as $\forall x(\zeta_{\tau,i} \rightarrow \eta_{\tau,i})$, where $\zeta_{\tau,i}$ gives the pre-conditions of the transition, and $\eta_{\tau,i}$, its effects (i.e. the part of (2.28) enclosed in square brackets). Since, in the intended interpretation, the pre-conditions $\zeta_{\tau,i}$ will never be simultaneously satisfied for two different values of $\tau \in T_L$, we can simply replace the conjunction of all such formulas for $\tau \in T_L$ by the single formula

$$\forall x \exists y \left(\bigwedge_{\tau \in T_L} \bigwedge_{0 \leq i < m} (\zeta_{\tau,i} \rightarrow \eta_{\tau,i}) \right).$$

Proceeding similarly for T_R, we obtain two formulas of the form $\forall x \exists y.\eta$. Any remaining purely universal formulas can of course be merged into either of these. □

Returning to the topic of the present section, we work our way from the language $\mathcal{L}_{\mathrm{Alt}}$ to the language $\mathcal{R}^+$, in two steps. The following notation (not standard in the literature) will be used throughout this book. If $\varphi(\bar{x})$ and $\psi(\bar{x})$ are formulas with no free variables other than $\bar{x}$, we write $\varphi \triangleleft_n \psi$ if: (i) ψ entails φ over domains of cardinality at least n; and (ii) every structure $\mathfrak{A}$ interpreting the signature of φ over a domain of cardinality at least n can be expanded to a structure $\mathfrak{A}'$ interpreting the signature of ψ such that, for any tuple of elements $\bar{a}$, $\mathfrak{A} \models \varphi[\bar{a}] \Rightarrow \mathfrak{A}' \models \psi[\bar{a}]$. Observe that, if $\varphi \triangleleft \psi$, then φ is (finitely) satisfiable if and only if ψ is.

Lemma 2.29. *Let $\mathcal{L}_{\mathcal{R}^+}$ be the set of first-order formulas of the forms*

$$\begin{array}{rl}
\forall x(p(x) \rightarrow \pm q(x)) & \forall x(p(x) \rightarrow \forall y(q(y) \rightarrow \pm r(x,y))) \\
\forall x(\neg p(x) \rightarrow \pm q(x)) & \forall x(p(x) \rightarrow \exists y(q(y) \wedge r(x,y))) \\
\exists x.p(x) & \forall x.p(x) \\
 & \forall x(p(x) \wedge q(x) \rightarrow o(x)),
\end{array}$$

where o, p and q are unary predicates and r is a binary predicate. The problem $\mathrm{Sat}(\mathcal{L}_{\mathcal{R}^+})$ *is* ExpTime*-hard.*

Proof. We proceed by reduction from $\mathrm{Sat}(\mathcal{L}_{\mathrm{Alt}})$ to $\mathrm{Sat}(\mathcal{L}_{\mathcal{R}^+})$, where $\mathcal{L}_{\mathrm{Alt}}$ is the language of Lemma 2.27. Let a collection of $\mathcal{L}_{\mathrm{Alt}}$-formulas Φ be given. It suffices to compute, in logarithmic space, a collection of $\mathcal{L}_{\mathcal{R}^+}$-formulas Ψ such that $(\bigwedge \Phi) \lhd (\bigwedge \Psi)$. Let $\varphi \in \Phi$. If φ contains a subformula $\theta = \neg p(x)$, let o_θ be a new predicate letter. Evidently, $\varphi \lhd \varphi[\theta/o_\theta(x)] \wedge \psi$, where ψ is

$$\forall x(o_\theta(x) \rightarrow \neg p(x)) \wedge \forall x(\neg o_\theta(x) \rightarrow p(x)).$$

Similarly, if φ contains some subformula $\theta = p(x) \wedge q(x)$, let o_θ be a new predicate letter. Again, $\varphi \lhd \varphi[\theta/o_\theta(x)] \wedge \psi$, where ψ is

$$\forall x(o_\theta(x) \rightarrow p(x)) \wedge \forall x(o_\theta(x) \rightarrow q(x)) \wedge \forall x(p(x) \wedge q(x) \rightarrow o_\theta(x)).$$

Similar remarks apply if φ contains some subformula $\theta = p(x) \vee q(x)$. Observe that ψ in all cases is a conjucntion of $\mathcal{L}_{\mathcal{R}^+}$-formulas. By means of a sequence of such transformations, we can replace the subformulas α, β and γ in the forms of Lemma 2.27 with unary atomic formulas, which yields a collection formulas in $\mathcal{L}_{\mathcal{R}^+}$. Thus, we eventually obtain a collection Ψ of $\mathcal{L}_{\mathcal{R}^+}$-formulas such that $(\bigwedge \Phi) \lhd (\bigwedge \Psi)$ as required. It is routine to check that this procedure requires only logarithmic working memory. □

Lemma 2.30. *The problem* Sat($\mathcal{R}^+$) *is* ExpTime*-hard.*

Proof. We proceed by reduction from Sat($\mathcal{L}_{\mathcal{R}^+}$) to Sat($\mathcal{R}^+$), where $\mathcal{L}_{\mathcal{R}^+}$ is the language of Lemma 2.29. Let a collection of $\mathcal{L}_{\mathcal{R}^+}$-formulas Φ be given. If $\varphi \in \Phi$, then either φ is directly expressible in $\mathcal{R}^+$, or is of the form $\varphi = \forall x(p(x) \wedge q(x) \rightarrow o(x))$. Let o^* be a new unary predicate and r^* a new binary predicate, and define ψ to be the conjunction of the formulas

$$\forall x(\neg o(x) \rightarrow \exists z(o^*(z) \wedge r^*(x, z))) \tag{2.29}$$
$$\forall x(p(x) \rightarrow \forall z(\neg p(z) \rightarrow \neg r^*(x, z))) \tag{2.30}$$
$$\forall x(q(x) \rightarrow \forall z(p(z) \rightarrow \neg r^*(x, z))) \tag{2.31}$$

We claim that $\varphi \lhd \psi$. Indeed, it is easy to check that $\models \psi \rightarrow \varphi$. For suppose (for contradiction) that $\mathfrak{A} \models \psi$ and a satisfies p and q but not o in $\mathfrak{A}$. By (2.29), there exists b such that $\mathfrak{A} \models r^*[a, b]$. If $\mathfrak{A} \not\models p[b]$, then (2.30) is false in $\mathfrak{A}$; on the other hand, if $\mathfrak{A} \models p[b]$, then (2.31) is false in $\mathfrak{A}$. Thus, $R_\theta \models \varphi$ as claimed. Conversely, if $\mathfrak{A} \models \varphi$, expand $\mathfrak{A}$ to a structure $\mathfrak{A}'$ by interpreting o^* and r^* as:

$$\begin{aligned} (o^*)^{\mathfrak{A}'} &= A \\ (r^*)^{\mathfrak{A}'} &= \{\langle a, a\rangle \mid \mathfrak{A} \not\models o[a]\}. \end{aligned}$$

We check that $\mathfrak{A}' \models R_\theta$. Formula (2.29) is true, because $\mathfrak{A}' \not\models o[a]$ implies $\mathfrak{A}' \models r^*[a, a]$. Formula (2.30) is true, because $\mathfrak{A}' \models r^*[a, b]$ implies $a = b$. To see that Formula (2.31) is true, suppose $\mathfrak{A}' \models q[a]$ and $\mathfrak{A}' \models p[b]$. If $a = b$, then $\mathfrak{A} \models o[a]$ (since $\mathfrak{A}' \models \varphi$); that is, either $a \neq b$ or $\mathfrak{A} \models o[a]$. By construction, then, $\mathfrak{A}' \not\models r^*[a, b]$.

All of the formulas (2.29)–(2.31) are directly expressible in $\mathcal{R}^+$. This proves the theorem. □

Theorem 2.31. *The problem* $\mathrm{Sat}(\mathcal{R}^+)$ *is* ExpTime*-complete. Moreover,* $\mathcal{R}^+$ *has the finite model property.*

Proof. The first statement is immediate Lemmas 2.24 and 2.30. For the finite model property, let Φ be a satisfiable finite set of formulas. Then the procedure described in the proof of Lemma 2.24 returns with success. Now let $\mathfrak{A}$ be the structure featured in Claim 2.26, over the (finite) domain A. We showed that $\mathfrak{A} \models \Phi$. □

We end this chapter with a remarkable corollary of Lemma 2.30. We saw above that the relational syllogistic, $\mathcal{R}$, has no finite set of syllogisms that is sound and complete; however, it does have one that is sound and refutation-complete. For the extended relational syllogistic, $\mathcal{R}^+$, not even this is true. To see this, we use (yet) another standard complexity-theoretic results from Sec. 1.4, namely, Proposition 1.5, which states that PTime is properly contained in ExpTime.

Corollary 2.32. *There exists no finite set of syllogisms* X *in* $\mathcal{R}^+$ *such that* $\vdash_{\mathsf{X}}$ *is sound and refutation-complete.*

Proof. Suppose such a set X exists. Note that there are only finitely many sentence forms in $\mathcal{R}$, and that each rule in X has finitely many antecedents. Let a set Φ of $\mathcal{R}^+$-formulas be given. We may assume without loss of generality that Φ contains at least one unary atom and at least one binary atom. Since syllogisms are by definition closed under uniform substitutions of (unary and binary) atoms, if there is a derivation of an absurdity from Φ, then there is one featuring no unary or binary atoms occurring outside Φ, and the number N of $\mathcal{R}^+$-formulas over this signature is thus bounded by a polynomial function of $\|\Phi\|$. But then it is a simple matter to search the space of valid derivations of length at most N in time bounded by a polynomial function of $\|\Phi\|$. The corollary then follows from Lemma 2.30 and Proposition 1.5. □

Concluding remarks

We began this chapter with an examination of one of the oldest and most intensively studied of all fragments of first-order logic, the classical syllogistic, $\mathcal{S}$, together with its extension by noun-level negation, $\mathcal{S}^+$. We showed that the satisfiability problems for both fragments are NLogSpace-complete, and provided a finite system of syllogisms for $\mathcal{S}^+$ that is sound and complete. We then examined the relational extensions of these fragments, namely $\mathcal{R}$ and $\mathcal{R}^+$. We showed that there is no finite system of syllogisms for $\mathcal{R}$ that is sound and complete; however, by providing one that is sound and *refutation*-complete, we were able to prove that the satisfiability problem for $\mathcal{R}$ remains NLogSpace-complete. We then went on to show that the corresponding problem for $\mathcal{R}^+$ is ExpTime-complete, and that $\mathcal{R}^+$ (and hence each of its sub-fragments) has the finite model property. We ended with the surprising corollary that there is no finite system of syllogisms for $\mathcal{R}^+$ that is sound and refutation-complete.

The languages $\mathcal{S}$, $\mathcal{S}^+$, $\mathcal{R}$ and $\mathcal{R}^+$ owe their salience to the syntax not of first-order logic, but rather of European languages such as Greek or English. For most decidable fragments of first-order logic, the opposite is the case. In the next chapter, we consider a fragment whose formulation in natural languages is typically quite awkward, but whose mathematical and computational properties make it every bit as worthy an object of study as the topic of the present chapter.

Exercises

1. To be done.

Bibliographic notes

The *locus classicus* (quite literally) for the language that we have called the classical syllogistic is Aristotle's *Prior Analytics*, book A, of which many readable translations are available (e.g. that by R. Smith [4]). The sentence-forms of the extended classical syllogistic are considered in *De Interpretatione*, Ch. 10 (ostensibly by Aristotle), though these are not integrated into any system of inference-rules. So much has been written on the classical syllogistic that it would be absurd to attempt a summary here: the reader is referred to a standard text such as W. and M. Kneale [48].

Relational sentences were commonly studied in mediæval logic. A good starting point is the engaging account by P. Spade [73].| Arguments involving relational notions in fact make sporadic appearances thereafter, including, for example, the *Port Royal Logic* [5, Ch. III]. However, it was perhaps De Morgan who most clearly recognized the inability of the classical syllogistic to cope with relational reasoning [57, p. 114]. De Morgan struggled mightily, but in vain, to understand doubly-quantified sentences in particular and relational propositions in general [58]. We cannot resist mentioning in this context the delicious argument between A. De Morgan and Sir William Hamilton (Bart.) on 'quantification of the predicate' [34, pp. 682–683] and [35, p. 277]; for latter-day analyses see R. Fogelin [23] and I. Pratt-Hartmann [65].

Reference needs revising.

The first mathematically useful analysis of the syllogistic, including a completeness theorem, was provided by J. Łukasiewicz and his student, J. Słupecki, in 1939, though the original manuscript was destroyed in a bombing raid, and did not appear in an easily accessible publication until 1956 [54]. Łukasiewicz' system (see also D. Westerståhl [79]) embeds the syllogistic in propositional logic, and uses a very unusual proof method. More standard completeness results can be found in the work of J. Sheperdson [71], T. Smiley [72], J. Corcoran [18] and J. Martin [55]. The particular treatment given above in Secs. 2.1 and 2.2 is taken from I. Pratt-Hartmann and L. Moss [66].

The first presentation of a complete proof system for a fragment close to the relational syllogistic seems to be that by N. Nishihara, K. Morita, and S. Iwata [60]. This logic is in effect a relational version of Łukasiewicz'. The results on the relational syllogistic presented here, in particular, Theorems 2.12 and 2.21, are taken from I. Pratt-Hartmann and L. Moss [66]. That paper gives a more detailed analysis account of the unavailability of syllogistic inference procedures, including those incorporating the rule of *reductio ad absurdum* (not considered in this chapter).

Lemma 2.30 and Theorem 2.31 may be regarded as folklore.

Chapter 3

Variables

In Ch. 2, we considered the relational syllogistic—a logic capturing English sentences such as "Every artist admires some beekeeper" or "No beekeeper despises every carpenter." The relational syllogistic cannot, however, accommodate sentences featuring relative clauses, for example,

Every artist whom some beekeeper admires despises some dentist
Every artist admires some beekeeper who despises some dentist
 who hates every electrician.

And it is natural to ask whether the satisfiability problem for the corresponding fragment of first-order logic remains decidable. In this chapter, we shall show that that it does, though with increased complexity. Consider the natural first-order translations of the above sentences, namely:

$$\forall x(\mathrm{artst}(x) \wedge \exists y(\mathrm{bkpr}(y) \wedge \mathrm{adm}(y,x)) \rightarrow \exists y(\mathrm{dntst}(y) \wedge \mathrm{desp}(x,y)))$$
$$\forall x(\mathrm{artst}(x) \rightarrow \exists y(\mathrm{bkpr}(y) \wedge \forall x(\mathrm{elctr}(x) \rightarrow \mathrm{hate}(y,x)) \wedge \mathrm{adm}(x,y))).$$

These formulas feature only two variables. Observe, however, the 're-use' of the variables x and y: there is nothing to prevent us from binding a variable by a fresh quantifier where its value in the surrounding context is not needed. It transpires that the fragment of first-order logic consisting of the function-free formulas in which at most two variables appear has the finite model property, and that its satisfiability problem is NExpTime-complete (thus, harder than for the relational syllogistic). Showing these facts will be the principal topic of this chapter.

The study of the two-variable fragment is significant firstly because many other naturally defined logics—in particular, standard propositional modal logic, as well as most description logics arising in the area of data modelling—are embedded in it, and secondly because it forms the starting point for more expressive extensions with decidable satisfiability problems—in particular, the two-variable fragment with counting quantifiers. The observation that restricting attention to two variables renders the satisfiability problem decidable prompts us to ask whether we can do any better. Here the answer is negative: the satisfiability and finite satisfiability problems for the three-variable fragment are already undecidable; and this we show as well. In Sec. 3.6 we show that the presence of predicates of arity 2 or more is essential for this latter result. We conclude the chapter with a characterization of the expressive power of the fragments of first-order logic obtained by limiting the number of variables.

3.1 Preliminaries

Denote by $\mathcal{FO}^k$ the fragment of first-order logic (including equality) over a signature with no function-symbols, in which just k variables, $x_1, \ldots, x_k$, occur. When $k \leq 3$, we shall write x, y, z rather than (the slightly clinical) x_1, x_2, x_3. We show in the sequel that the satisfiability (= finite satisfiability) problem for for $\mathcal{FO}^2$ is NExpTime-complete, but that the corresponding problems for $\mathcal{FO}^3$—and hence for all $\mathcal{FO}^k$ with $k > 2$—are undecidable. Since $\mathcal{FO}^k$ is closed under conjunction, we may take the inputs to Sat($\mathcal{FO}^k$) and FinSat($\mathcal{FO}^k$) to be single formulas, rather than finite sets of formulas.

We employ a technique that will be encountered frequently in the sequel when analysing logical fragments: the identification of *normal forms*. It will be convenient to generalize the notation $\varphi \lhd \psi$ first introduced in Sec. 2.4. If $\varphi(\bar{x})$ and $\psi(\bar{x})$ are formulas with no free variables other than $\bar{x}$, we write $\varphi \lhd_n \psi$ if: (i) ψ entails φ over domains of cardinality at least n; and (ii) every structure $\mathfrak{A}$ interpreting the signature of φ over a domain of cardinality at least n can be expanded to a structure $\mathfrak{A}'$ interpreting the signature of ψ such that, for any tuple of elements $\bar{a}$, $\mathfrak{A} \models \varphi[\bar{a}] \Rightarrow \mathfrak{A}' \models \psi[\bar{a}]$. In particular, if $\varphi \lhd_n \psi$, then φ and ψ are satisfiable over the same domains having at least n elements. Obviously, $\lhd_n$ is a transitive relation, and $\varphi \lhd_n \psi$ implies $\varphi \lhd_{n'} \psi$ whenever $n < n'$. We write $\varphi \lhd \psi$ to mean $\varphi \lhd_1 \psi$.

Lemma 3.1. *Let φ be an $\mathcal{FO}^k$-formula. We can construct, in time bounded by a polynomial function of $\|\varphi\|$, an $\mathcal{FO}^k$-formula*

$$\psi := \forall \bar{x}.\alpha \wedge \bigwedge_{1 \leq h \leq m} \forall \bar{x}_h \exists y_h.\beta_h, \tag{3.1}$$

where $0 \leq m \leq \|\varphi\|$, $\bar{x}, \bar{x}_1, \ldots, \bar{x}_m$, are (possibly empty) tuples of variables, $y_1, \ldots, y_m$ are variables, and $\alpha, \beta_1, \ldots, \beta_m$ are quantifier-free $\mathcal{FO}^k$-formulas, such that $\varphi \lhd \psi$.

Proof. Write $\varphi_0 = \varphi$. If φ_0 is quantifier-free, we may put $\alpha = \varphi_0$, $\bar{x} = \epsilon$ (the empty sequence), $m = 0$, and we are done.

Suppose, then φ_0 has a sub-formula $\theta(\bar{u}) = \exists v.\chi$, with χ quantifier-free and vars(χ) $= \bar{u}v$. Let p be a fresh predicate of arity $|\bar{u}|$, let φ_1 be $\varphi[\theta/p(\bar{u})]$, i.e. the result of substituting $p(\bar{u})$ for θ in φ, and let

$$\psi_1 := \forall \bar{u} \exists v (p(\bar{u}) \rightarrow \chi) \wedge \forall \bar{u} \forall v (\chi \rightarrow p(\bar{u})).$$

We claim that $\varphi_0 \lhd (\varphi_1 \wedge \psi_1)$. Observing that $\psi_1 \equiv \forall \bar{u}(p(\bar{u}) \leftrightarrow \theta)$, it is immediate that $\varphi_1 \wedge \psi_1$ entails φ_0; moreover, any model $\mathfrak{A}$ of φ_0 may be expanded to a model of $\varphi_1 \wedge \psi_1$ by interpreting p to be satisfied by a tuple $\bar{a} \in A$ if and only if $\mathfrak{A} \models \theta[\bar{a}]$. Similarly, if φ_0 has a proper sub-formula $\theta = \forall v.\chi$, with χ quantifier-free, define φ_1 as before, but set

$$\psi_1 := \forall \bar{u} \forall v(p(\bar{u}) \rightarrow \chi) \wedge \forall \bar{u} \exists v(\chi \rightarrow p(\bar{u})).$$

Again, then, $\varphi_0 \lhd (\varphi_1 \wedge \psi_1)$. Now process φ_1 in the same way to obtain φ_2 and ψ_2, and continue until some quantifier-free formula φ_n is reached. Thus, setting

$$\psi' := \varphi_n \wedge \psi_n \wedge \psi_{n-1} \wedge \cdots \wedge \psi_1,$$

we have $\varphi \lhd \psi'$. By renaming the variables if necessary, we can ensure that φ_n has no variables in common with any of the tuples $\bar{x}_h$ $(1 \leq h \leq n)$. By re-arranging the conjuncts of ψ', we obtain the desired formula ψ. □

We mention at this point an immediate corollary that will be useful in subsequent chapters. A formula is in *prenex form* if none of its Boolean connectives out-scopes any of its quantifiers. Thus, for example the sentence

$$\forall x(\text{grad-student}(x) \rightarrow (\text{student}(x) \wedge \exists y.\text{supervises}(y, x)))$$

is not in prenex form, while the (logically equivalent) sentence

$$\forall x \exists y(\text{grad-student}(x) \rightarrow (\text{student}(x) \wedge \text{supervises}(y, x))),$$

in which the existential quantifier has been fronted, is.

Lemma 3.2. *Let φ be a formula of first-order logic. We can construct, in time bounded by a polynomial function of $\|\varphi\|$, a first-order formula ψ in prenex form, such that $\varphi \lhd \psi$.*

Proof. Let ψ be the formula guaranteed by Lemma 3.1. By renaming variables if necessary, we may assume that the existentially quantified variables y_h do not occur outside the respective subformulas β_h $(1 \leq h \leq m)$. Then ψ is logically equivalent to

$$\psi := \forall \bar{x} \bar{x}_1 \cdots \bar{x}_m \exists y_1 \cdots y_m \left(\alpha \wedge \bigwedge_{1 \leq h \leq m} \beta_h \right).$$

□

We remark that this is not the only method of conversion to prenex form; indeed, it has the special property that all the universal quantifiers outscope all the existential quantifiers. (See Exercise 1 for an alternative.) But it will do.

Returning to the subject of this chapter, we obtain a more specialized normal form in the case of $\mathcal{FO}^2$, often referred to as *Scott normal form.*

Lemma 3.3. *Let φ be a formula in $\mathcal{FO}^2$. We can construct, in time bounded by a polynomial function of $\|\varphi\|$, an $\mathcal{FO}^2$-formula*

$$\psi := \forall x \forall y(\alpha \vee x = y) \wedge \bigwedge_{1 \leq h \leq m} \forall x \exists y(\beta_h(x, y) \wedge x \neq y), \tag{3.2}$$

where $1 \leq m \leq \|\varphi\|$, and $\alpha, \beta_1, \ldots, \beta_m$ are quantifier-free, equality-free $\mathcal{FO}^2$-formulas, such that $\varphi \lhd_2 \psi$.

Proof. Let ψ' be the formula guaranteed by Lemma 3.1. Since no sub-formula encountered in processing of φ involves more than two variables, we may write

$$\psi' := \forall x \forall y.\alpha'(x, y) \wedge \bigwedge_{1 \leq h \leq m} \forall x \exists y.\beta'_h(x, y).$$

It remains only to reform the occurrences of $=$ in $\alpha'(x, y)$ and the $\beta'_h(x, y)$. Restricting attention to domains containing at least 2 elements, we have the following logical equivalences:

$$\begin{aligned} \forall x \forall y.\alpha'(x, y) &\equiv \forall x \forall y((\alpha'(x, x) \wedge \alpha'(x, y)) \vee x = y) \\ \forall x \exists y.\beta'_h(x, y) &\equiv \forall x \exists y((\beta'_h(x, x) \vee \beta'_h(x, y)) \wedge x \neq y). \end{aligned}$$

Now, if θ is any $\mathcal{FO}^2$-formula, denote by θ^* the result of replacing all atoms $x = x$ and $y = y$ in θ by $\top$, and all atoms $x = y$ and $y = x$ by $\bot$. Thus, we have the logical equivalences

$$\theta \vee x = y \equiv \theta^* \vee x = y \qquad \theta \wedge x \neq y \equiv \theta^* \wedge x \neq y.$$

Setting

$$\begin{aligned} \alpha &:= \big(\alpha'(x, y) \wedge \alpha'(x, x)\big)^* \\ \beta_h &:= \big(\beta'_h(x, y) \vee \beta'_h(x, x)\big)^*, \end{aligned}$$

for all h ($1 \leq h \leq m$), and then

$$\psi := \forall x \forall y(\alpha(x, y) \vee x = y) \wedge \bigwedge_{1 \leq h \leq m} \forall x \exists y(\beta_h(x, y) \wedge x \neq y),$$

we see that ψ and ψ' are logically equivalent over domains containing at least 2 elements, whence $\psi' \lhd_2 \psi$. Thus, ψ has the properties required for the lemma. □

Note that individual constants and predicates of arity 3 or higher are permitted in $\mathcal{FO}^2$, as for example in the formula $\forall x \forall y(p(c, x, y, e) \rightarrow p(d, y, y, x, c))$, and yet it seems that, with only two variables at our disposal, these resources must somehow be superfluous. The proof of this fact is surprisingly tricky, if perhaps slightly dry. We present it, for completeness' sake, in the next lemma, which the reader may wish to skip on first reading.

Lemma 3.4. *Let φ be a formula of $\mathcal{FO}^2$. Then we may compute, in time bounded by a polynomial function of $\|\varphi\|$, a $\mathcal{FO}^2$-formula ψ over a signature of nullary, unary and binary predicates only, such that φ and ψ are satisfiable over the same domains.*

Proof. Let C be the set of individual constants appearing in φ, and $\mathbf{A}$ the set of atoms appearing in φ and featuring *either* a predicate of arity greater than 2 *or* an individual constant. By a *substitution*, we here mean a function $\tau : \{x, y\} \to \{x, y\}$, where x and y are regarded simply as symbols. We extend any substitution to expressions containing x and y homomorphically. Let $\mathbf{A}^+$ be the set of atoms defined by $\mathbf{A}^+ = \{\tau(\alpha) \mid \alpha \in \mathbf{A},\, \tau \text{ a substitution}\}$. Thus, we certainly have $|\mathbf{A}^+| \leq 4 \cdot \|\varphi\|$. For each $c \in C$, let p_c be a fresh unary predicate, and for each $\alpha(x, y) \in \mathbf{A}^+$, let p_α be a fresh binary predicate. Now let $\hat{\varphi}$ be the result of replacing α by $p_\alpha(x, y)$ in φ, for every $\alpha \in \mathbf{A}$. Thus, $\hat{\varphi}$ features only predicates of arity 0 1 and 2, and is constant-free. The idea is that $\hat{\varphi}$ functions as a surrogate for φ in respect of satisfiability.

To make this idea viable, we must enforce various constraints on the interpretations of the predicates p_α. These constraints will in fact feature not only the predicates p_α for α in $\mathbf{A}$, but the predicates p_α for α in the whole of $\mathbf{A}^+$ and the predicates p_c for $c \in C$. Think of of $p_\alpha[b, b']$ as meaning "$\alpha(x, y)$ is satisfied by the ordered pair b, b'" and $p_c[b]$ as meaning "c is interpreted as the element b". We conjoin $\hat{\varphi}$ with a sentence $\bigwedge \Psi$, where the elements of Ψ are sentences in the vocabulary of $\hat{\varphi}$, constructed as follows. First, Ψ contains sentences ensuring that each predicate p_c is uniquely satisfied:

$$\exists x . p_c(x) \wedge \forall x \forall y (p_c(x) \wedge p_c(y) \to x = y). \tag{3.3}$$

Second, Ψ contains sentences ensuring that any two atoms of $\mathbf{A}^+$ differing only by a substitution τ correspond to binary predicates with commensurate extensions. Thus, for all $\alpha \in \mathbf{A}^+$ and all substitutions τ, we have the constraint

$$\forall x \forall y (p_{\tau(\alpha)}(x, y) \leftrightarrow p_\alpha(\tau(x), \tau(y))). \tag{3.4}$$

Third, Ψ contains sentences ensuring that atoms of $\mathbf{A}^+$ involving individual constants correspond to binary predicates with appropriate extensions. Here we require some notation. Any $\alpha \in \mathbf{A}^+$ can be written $\alpha = r(v)$, where v is a word in $(C \cup \{x, y\})^*$ of length k (the arity of the predicate r). Write $v = v[1] \cdots v[k]$. Now, for any distinct elements s, t of the alphabet $C \cup \{x, y\}$, define the formula $\eta_{s,t}(x, y)$ as follows (the order of indices, s and t, is not significant):

$$\begin{aligned}
&\eta_{c,d} := \exists x (p_c(x) \wedge p_d(x)) && c, d \in C \\
&\eta_{c,x} := p_c(x) && c \in C \\
&\eta_{x,x} := \eta_{y,y} := \top && \\
&\eta_{x,y} := x = y &&
\end{aligned}$$

As an aide to intuition, read $\eta_{s,t}[b,b']$ as stating that s and t must denote the same object under the assignment $x, y \mapsto b, b'$. We take Ψ to contain the sentences

$$\forall x \forall y \left(\left(\bigwedge_{i=1}^{|w_1|} \eta_{w_1[i],w_2[i]}(x,y) \right) \to (p_{r(w_1)}(x,y) \to p_{r(w_2)}(x,y)) \right), \tag{3.5}$$

for all pairs of atoms $r(w_1)$, $r(w_2)$ in $\mathbf{A}^+$ (of course with $|w_1| = |w_2|$). Thus, these constraints state that, if w_1 and w_2 denote the same tuple of objects under the assignment $x, y \mapsto b', b'$, then the predicates $p_{r(w_1)}$ and $p_{r(w_2)}$ are either both satisfied by that assignment, or both not satisfied by it. (Remember that w_1 and w_2 may contain—possibly distinct but co-denoting—individual constants.) Finally, define $\psi := \hat{\varphi} \wedge \bigwedge \Psi$.

It remains to show that φ is satisfiable over a given domain if and only if ψ is. Suppose first that $\mathfrak{A} \models \varphi$. Let $\hat{\mathfrak{A}}$ be the expansion of $\mathfrak{A}$ obtained by setting $p_c^{\hat{\mathfrak{A}}} = \{c^{\mathfrak{A}}\}$ and $p_\alpha^{\hat{\mathfrak{A}}} = \{\bar{a} \mid \mathfrak{A} \models \alpha[\bar{a}]\}$ (where $\bar{a} \in A^2$). A quick check shows that $\hat{\mathfrak{A}} \models \psi$. For the non-trivial direction, suppose $\mathfrak{B} \models \psi$. For each $c \in C$, let b_c be the unique element of B satisfying p_c in $\mathfrak{B}$. We introduce some more notation. If w is any word in $(C \cup \{x,y\})^*$ of length k, and $\bar{b} = \langle b, b' \rangle$ an ordered pair of elements of B, let $w[\bar{b}]$ denote the the k-tuple from B obtained from w by replacing x by b, y by b', and any individual constant c by b_c. Now define $\check{\mathfrak{B}}$ to be the expansion of $\mathfrak{B}$ obtained by setting $c^{\check{\mathfrak{B}}} = b_c$ for each $c \in C$, and

$$r^{\check{\mathfrak{B}}} := \{w[\bar{b}] \mid w \in (C \cup \{x,y\})^* \text{ and } \bar{b} \in B^2 \text{ s.t } \mathfrak{B} \models p_{r(w)}[\bar{b}]\} \tag{3.6}$$

We claim that $\check{\mathfrak{B}} \models \varphi$, thus completing the proof. The key observation is as follows. Suppose w_1 and w_2 are words in $(C \cup \{x,y\})^*$ and $\bar{b}_1$, $\bar{b}_2$ ordered pairs over B such that $w_1[\bar{b}_1] = w_2[\bar{b}_2]$. Since $\mathfrak{B} \models \Psi$, it follows that $\mathfrak{B} \models p_{r(w_1)}[\bar{b}_1]$ if and only if $\mathfrak{B} \models p_{r(w_2)}[\bar{b}_2]$. Hence, for every $w \in (C \cup \{x,y\})^*$ and every ordered pair $\bar{b}$ over B, it follows from the construction of $\check{\mathfrak{B}}$ in (3.6) that, for any atom $r(w) \in \mathbf{A}^+$, we have

$$\check{\mathfrak{B}} \models r(w)[\bar{b}] \Longleftrightarrow \mathfrak{B} \models p_{r(w)}[\bar{b}].$$

(It is the left-to-right implication that is non-trivial here.) A routine structural induction on sub-formulas of φ now establishes that $\check{\mathfrak{B}} \models \varphi$. □

In view of Lemma 3.4, we shall henceforth assume $\mathcal{FO}^2$-formulas to feature only nullary, unary or binary predicates, since, in terms of satisfiability, individual constants and predicates of higher arity make no difference. This observation does not extend to function-symbols. (See exercise 4.)

We finish this section with some simple model-theoretic concepts which will feature repeatedly in the sequel. Fix some purely relational signature Σ. A *literal* (over Σ) is an atomic formula or the negation of an atomic formula. A *k-type* (over Σ) is a maximal consistent set of equality-free literals over Σ involving only the variables $x_1, \ldots, x_k$. Reference to Σ is suppressed where clear from context. Intuitively, we are invited to think of a k-type τ as a complete specification of the relations determined by some structure on a particular selection of k (distinct) individuals. If $\mathfrak{A}$ is a structure interpreting Σ, and $\bar{a} = a_1, \ldots, a_k$ are *distinct* elements of A, then there exists a unique k-type $\tau(x_1, \ldots, x_k)$ such that $\mathfrak{A} \models \tau[a_1, \ldots, a_k]$. We call τ the *k-type of* $a_1, \ldots, a_k$ in $\mathfrak{A}$, and denote it $\mathrm{tp}^{\mathfrak{A}}[\bar{a}]$; in this case we say also that $\bar{a}$ *realizes* τ in $\mathfrak{A}$. In this book, we do not define $\mathrm{tp}^{\mathfrak{A}}[\bar{a}]$ if $\bar{a}$ contains repeated elements. For convenience, we typically identify τ with the conjunction of its elements; this allows us to write τ in formulas rather than the more correct (but slightly cumbersome) $\bigwedge \tau$. In some contexts, it even helps to think of a k-type τ neither as a set of literals nor as a formula, but rather, as a structure $\mathfrak{X}_\tau$ over the domain $X = \{x_1, \ldots, x_k\}$ defined by setting, for every k-ary predicate r in Σ, $r^{\mathfrak{X}_\tau} = \{w \mid w \text{ a word of length } k \text{ over } X \text{ such that } r(w) \in \tau\}$. Thus, the k-type τ is realized in the structure $\mathfrak{A}$ just in case there is an embedding of $\mathfrak{X}_\tau$ into $\mathfrak{A}$. Of most concern to us in the present chapter (and indeed in most chapters of this book) are 1- and 2-types. Using the variables x and y instead of x_1, and x_2, a *1-type* (over Σ) may be taken to be a maximal consistent set of equality-free literals over Σ involving only the variable x, and a *2-type*, a maximal consistent set of equality-free literals over Σ involving only the variables x and y.

Let τ be a 2-type over a purely relational signature Σ. The result of transposing the variables x and y in τ is also a 2-type, denoted τ^{-1}; the set of literals in τ not featuring the variable y is a 1-type, denoted $\mathrm{tp}_1(\tau)$; likewise, the set of literals in τ not featuring the variable x is also a 1-type, denoted $\mathrm{tp}_2(\tau)$. Thus, if τ is any 2-type over a purely relational signature Σ, then $\mathrm{tp}_2(\tau) = \mathrm{tp}_1(\tau^{-1})$. If $\mathfrak{A}$ is a structure interpreting Σ, and a, b are distinct elements of A such that $\mathrm{tp}^{\mathfrak{A}}[a, b] = \tau$, then $\mathrm{tp}^{\mathfrak{A}}[b, a] = \tau^{-1}$, $\mathrm{tp}^{\mathfrak{A}}[a] = \mathrm{tp}_1(\tau)$ and $\mathrm{tp}^{\mathfrak{A}}[b] = \mathrm{tp}_2(\tau)$. If $B \subseteq A$, we write $\mathrm{tp}^{\mathfrak{A}}[B]$ for the set $\{\mathrm{tp}^{\mathfrak{A}}[b] : b \in B\}$ of 1-types realized by elements of B. Similarly if $C \subseteq A$ (with B, C not necessarily disjoint), we write $\mathrm{tp}^{\mathfrak{A}}[B, C]$ for the set $\{\mathrm{tp}^{\mathfrak{A}}[b, c] : b \in B, c \in C, b \neq c\}$. We identify elements with singletons in this notation, writing, for example $\mathrm{tp}^{\mathfrak{A}}[b, C]$ instead of $\mathrm{tp}^{\mathfrak{A}}[\{b\}, C]$. Observe that, if Σ features only unary and binary predicates, and $|\Sigma| = s$, then there are exactly 2^s 1-types over Σ and at most 2^{4s} 2-types.

3.2 Small sub-structures

The principal task of this section is to prove Lemma 3.6, which establishes a *small model property* for $\mathcal{FO}^2$: any satisfiable $\mathcal{FO}^2$-formula φ has a model of size bounded by a fixed function of $\|\varphi\|$. Indeed, we can find such a function having at most exponential growth. As an immediate corollary, the satisfiability problem for $\mathcal{FO}^2$ is decidable, in NExpTime. Actually, we shall prove a slightly stronger result that we strictly need here: in any model of an $\mathcal{FO}^2$-sentence φ, it is possible to replace *any given sub-structure* with one of size bounded by a fixed exponential function of $\|\varphi\|$, such that the resulting structure remains a model of φ. The extra strength will be useful in Ch. **??**.

Lemma 3.5. *Let $\mathfrak{A}$ be a structure interpreting a signature of unary and binary* *predicates, let B be a subset of A such that $\text{tp}^{\mathfrak{A}}[B] = \{\alpha\}$ for some 1-type α, and let $C = A \setminus B$. Then there is a structure $\mathfrak{A}'$ interpreting the same signature over a domain $A' = B' \cup C$ for some set B' of size bounded by $3 \cdot |\text{tp}^{\mathfrak{A}}[A, A]|^2$, such that:*

(*i*) $\mathfrak{A}' {\restriction} C = \mathfrak{A} {\restriction} C$;

(*ii*) $\text{tp}^{\mathfrak{A}'}[B'] = \text{tp}^{\mathfrak{A}}[B] = \{\alpha\}$, *whence* $\text{tp}^{\mathfrak{A}'}[A'] = \text{tp}^{\mathfrak{A}}[A]$;

(*iii*) $\text{tp}^{\mathfrak{A}'}[B', B'] = \text{tp}^{\mathfrak{A}}[B, B]$ *and* $\text{tp}^{\mathfrak{A}'}[B', C] = \text{tp}^{\mathfrak{A}}[B, C]$, *whence* $\text{tp}^{\mathfrak{A}'}[A', A'] = \text{tp}^{\mathfrak{A}}[A, A]$;

(*iv*) *for each $b' \in B'$ there is some $b \in B$ with* $\text{tp}^{\mathfrak{A}'}[b', A'] \supseteq \text{tp}^{\mathfrak{A}}[b, A]$;

(*v*) *for each $a \in C$,* $\text{tp}^{\mathfrak{A}'}[a, B'] \supseteq \text{tp}^{\mathfrak{A}}[a, B]$.

(*vi*) *for each $b' \in B'$,* $\text{tp}^{\mathfrak{A}'}[b', B'] = \text{tp}^{\mathfrak{A}}[B]$.

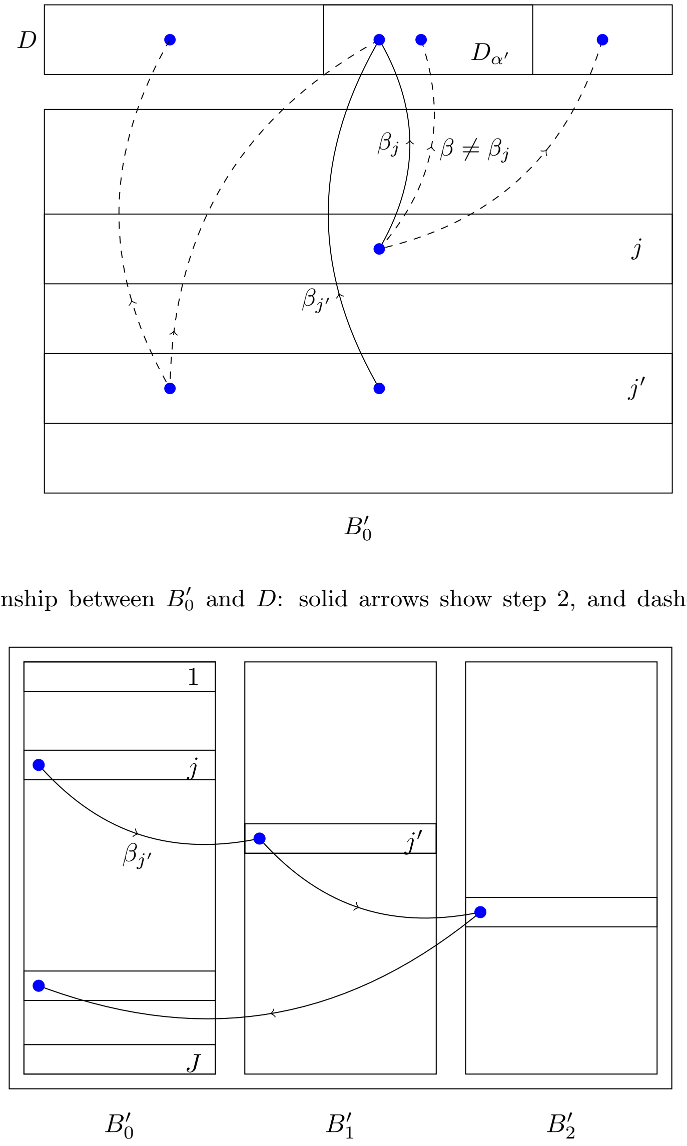
Proof. If $|B| \leq 1$, we may set $B' = B$ and $\mathfrak{A}' = \mathfrak{A}$, and there is nothing to prove. Assume, then, that $|B| \geq 2$.

Let $\text{tp}^{\mathfrak{A}}[A, A] = \{\beta_1, \ldots, \beta_J\}$. (Thus, $J = |\text{tp}^{\mathfrak{A}}[A, A]|$.) For each 1-type $\alpha' \in \text{tp}^{\mathfrak{A}}[A]$, let $\text{tp}^B(\alpha') = \{\text{tp}^{\mathfrak{A}}[b, a] \mid b \in B, a \in C, \text{tp}^{\mathfrak{A}}[a] = \alpha'\}$; and let $N(\alpha') = |\text{tp}^B(\alpha')|$. For each 1-type α' realized by some element of C, let $D_{\alpha'} \subseteq C$ be a set of elements having 1-type α', of size $N(\alpha')$, or all of these elements if there are fewer than $N(\alpha')$. Let D be the union of all the D_α. Evidently, $|D| \leq J$. Now set $B' = D \times \{1, \ldots, J\} \times \{0, 1, 2\}$. Thus, $|B'| \leq 3J^2$. We group the elements of B' into three 'blocks', according to their third coordinate: $B'_k = D \times \{1, \ldots, J\} \times \{k\}$ for all k $(0 \leq k < 3)$. We proceed to define the structure $\mathfrak{A}'$ over domain $A' = B' \cup C$. It helps to think of D as a sufficiently large representative sample of elements of C to serve as witnesses for the elements of B'.

Step 1: Define $\mathfrak{A}' {\restriction} C = \mathfrak{A} {\restriction} C$, This secures (i).

Step 2: For each $a \in D$, let $\alpha' = \text{tp}^{\mathfrak{A}}[a]$, and, for each $\beta_j \in \text{tp}^B(\alpha')$, set $\text{tp}^{\mathfrak{A}'}[(a, j, 0), a] = \beta_j$. This secures (v) for all $a \in D$ (Fig. 3.1(a), solid arrows).

Step 3: For each $b' = (a, j, k) \in B'$, if no 2-type involving b' was set in Step 2, then pick any $b \in B$. For each $\beta \in \text{tp}^{\mathfrak{A}}[b, C]$, let $\alpha' = \text{tp}_2(\beta)$, select a fresh $a \in D_{\alpha'}$, and set $\text{tp}^{\mathfrak{A}'}[b', a] = \beta$. Certainly, there will be enough elements of $D_{\alpha'}$ to go round, and, as a result, b' will be related to the elements of D in $\mathfrak{A}'$ with the same array of 2-types as that with which b is related to the elements of C in $\mathfrak{A}$. If, on the other hand some 2-type involving b' was set in Step 2 (hence $k = 0$), say as $\text{tp}^{\mathfrak{A}'}[b', a] = \beta_j = \text{tp}[a, b]$, with $a \in D$ and $b \in B$, then, for every other $\beta \in \text{tp}^{\mathfrak{A}}[b, C]$ (i.e. $\beta \neq \beta_j$), let $\alpha' = \text{tp}_2(\beta)$, select a fresh $a \in D_{\alpha'}$, and set $\text{tp}^{\mathfrak{A}'}[b', a] = \beta$. Again, there will be enough elements of $D_{\alpha'}$ to go round, and b' will be related to the elements of D in $\mathfrak{A}'$ with the same array of 2-types as that with which b is related to the elements of C in $\mathfrak{A}$. (This arrangement is



(a) Relationship between B'_0 and D: solid arrows show step 2, and dashed arrows, step 3.

(b) Circular witnessing in B', step 6.

Figure 3.1: Proof of Lemma 3.5.

depicted, for the special case of B'_0, by the dashed arrows Fig. 3.1(a).) However $\mathfrak{A}'$ is completed, therefore, (ii) and (iv) are secured.

Step 4: Pick any element $b_0 \in B$. For each $b' \in B'$ and each $a \in D$, such that $\mathrm{tp}^{\mathfrak{A}'}[b', a]$ was not defined in Steps 2 and 3, set $\mathrm{tp}^{\mathfrak{A}'}[b', a] = \mathrm{tp}^{\mathfrak{A}}[b_0, a]$. At this point, all 2-types are defined for $B' \times D$, in such a way that $\mathrm{tp}^{\mathfrak{A}'}[B', D] \subseteq \mathrm{tp}^{\mathfrak{A}}[B, C]$.

Step 5: We now define the 2-types for $B' \times (C \setminus D)$. For each $a \in C \setminus D$, pick $a_0 \in D_{\alpha'}$, where $\alpha' = \mathrm{tp}^{\mathfrak{A}}[a]$, and, for each $b' \in B'$, set $\mathrm{tp}^{\mathfrak{A}'}[b', a] = \mathrm{tp}^{\mathfrak{A}'}[b', a_0]$. (Note that the right-hand side of this assignment was defined in Steps 2–4.) This ensures that a is related to some element of B' in $\mathfrak{A}'$ by every 2-type in $\mathrm{tp}^B(\alpha)$. Hence $\mathrm{tp}^{\mathfrak{A}'}[a, B'] = \mathrm{tp}^B(\alpha) \supseteq \mathrm{tp}^{\mathfrak{A}}[a, B]$, thus securing (iii) and (v).

Step 6: The only 2-types left to define are those involving elements of B. Here we use 'circular witnessing', a technique we shall encounter at various points in this book. If n is an integer, denote $n \mod 3$ by $\lfloor n \rfloor$, and recall that $B' = D \times \{1, \ldots, J\} \times \{0, 1, 2\}$, with $\{\beta_1, \ldots, \beta_J\} = \mathrm{tp}^{\mathfrak{A}}[A, A]$. For any j, j' $(1 \leq j, j' \leq J)$ and any k $(0 \leq k < 3)$, if $\beta_{j'} \in \mathrm{tp}[B, B]$, then set $\mathrm{tp}^{\mathfrak{A}'}[(a, j, k), (a, j', \lfloor k+1 \rfloor)] = \beta_{j'}$. In other words, we provide the required witnesses for the elements of each block of B' by selecting elements from the 'next' block (Fig. 3.1(b)). By inspection, no clashes can arise in this process. Finally, for b' and b'' such that $\mathrm{tp}^{\mathfrak{A}'}[b', b'']$ is not yet defined, pick *any* distinct $b_0, b_1 \in B$ and set $\mathrm{tp}^{\mathfrak{A}'}[b', b''] = \mathrm{tp}^{\mathfrak{A}'}[b_0, b_1]$. (This is where we use the fact that $|B| \geq 2$.) This secures (iii) and (vi). □

Conditions (i)-(vi) of Lemma 3.5 ensure that any prenex $\forall\forall$- or $\forall\exists$-formula of $\mathcal{FO}^2$ satisfied in $\mathfrak{A}$ is also satisfied in $\mathfrak{A}'$.

Lemma 3.6. *Let φ be a satisfiable $\mathcal{FO}^2$-formula in normal form* (3.2) *over a signature Σ. Then φ has a model of size at most $3 \cdot 2^{9|\Sigma|}$. Thus, $\mathcal{FO}^2$ has the finite model property, and* $\mathrm{Sat}(\mathcal{FO}^2)$ *is in* NExpTime.

Proof. Suppose $\mathfrak{A} \models \varphi$. Thus $|\mathrm{tp}^{\mathfrak{A}}[A, A]| \leq 2^{4|\Sigma|}$. For each 1-type α realized in $\mathfrak{A}$ in turn, let $B = \{a \in A \mid \mathrm{tp}^{\mathfrak{A}}[a] = \alpha\}$ and apply Lemma 3.5. After at most $2^{|\Sigma|}$ rounds, we obtain the desired model of φ. For the second sentence of the Lemma, let an $\mathcal{FO}^2$-formula φ be given. By Lemma 3.3, we can compute ψ in polynomial time such that ψ is satisfiable (necessarily over a domain of size at least 2) if and only if φ is satisfiable over a domain of size at least 2, which condition can be checked by guessing and checking a model of size at most $3 \cdot 2^{9|\Sigma|}$, where Σ is the signature of ψ. Satisfiability of φ over a 1-element domain can be checked in non-deterministic polynomial time. □

3.3 Excursus: Tiling problems

This section introduces the apparatus of *tiling problems*, a device frequently used to establishing lower complexity bounds which will make repeated appearances in this book.

A *tiling system* is a triple (C, H, V), where C is a finite set, and H, V are binary relations over C. We call the elements of C *colours*, and we call H and V the *horizontal constraints* and the *vertical constraints*, respectively. A (C, H, V)-*tiling of* $\mathbb{N}^2$ is a function $t : \mathbb{N}^2 \to C$ such that, for all $i, j \in \mathbb{N}$: (i) , $\langle t(i,j), t(i+1,j)\rangle \in H$, and (ii) $\langle t(i,j), t(i,j+1)\rangle \in V$. Intuitively, the elements of C represent types of unit square tile which must be arranged in a grid. The pairs in H list which colours can go immediately to the 'right' of which others; the pairs in V list which colours can go immediately 'above' which others.

The notion of a tiling of $\mathbb{N}^2$ has an obvious finite analogue. For $n > 0$, let $\mathbb{N}_n$ denote the set $\{0, \ldots, n-1\}$. For $m, n > 0$, a (C, H, V)-*tiling of* $\mathbb{N}_m \times \mathbb{N}_n$ is a function $t : \mathbb{N}_m \times \mathbb{N}_n \to C$CFT such that: (i) for all i, j, $(0 \leq i < m-1$, $0 \leq j \leq n-1)$, $\langle t(i,j), t(i+1,j)\rangle \in H$, (ii) for all i, j $(0 \leq i \leq m-1, 0 \leq j < n-1)$, $\langle t(i,j), t(i,j+1)\rangle \in V$. Intuitively, a such a tiling is a set of instructions for coving an $n \times m$ rectangular block (i.e. n rows, m columns) with unit-square tiles having colours from C in such a way as to respect the constraints H and V. We remark that there is no 'toroidal wrap-around' of this rectangle: squares on the extremal columns have only one horizontal neighbour, and squares on the extremal rows have only one vertical neighbour.

Returning to tilings of $\mathbb{N}^2$, we consider the following question. Given a tiling system (C, H, V) and an element $c_\alpha \in C$, does there exist a (C, H, V)-tiling of $\mathbb{N}^2$ in which the origin $(0, 0)$ is assigned the colour $c_\alpha \in C$? We call this problem the *constrained infinite tiling problem* (CIT):

CIT
Given: A tiling system (C, H, V) and an element $c_\alpha \in C$:
Return: Yes if there is a (C, H, V)-tiling t of $\mathbb{N}^2$ s.t. $t(0, 0) = c_\alpha$;
No otherwise.

Thus, CIT asks whether a given tiling system tiles the upper-right quadrant of the plane, subject to the constraint that a given tile lies in the bottom left-hand corner. We can of course pose an analogous problem for tilings of $\mathbb{N}_m \times \mathbb{N}_n$. In this case, however, it will be more convenient to constrain not only the bottom left-hand corner, but also the top right-hand one. Thus, we define the *constrained finite tiling problem* (CFT) as follows:

CFT
Given: A tiling system (C, H, V) and elements $c_\alpha, c_\omega \in C$:
Return: Yes if there is a (C, H, V)-tiling t of $\mathbb{N}_m \times \mathbb{N}_n$ for some
$m, n > 0$, such that $t(0, 0) = c_\alpha$ and $t(m-1, n-1) = c_\omega$;
No otherwise.

Thus, CFT asks whether a given tiling system tiles a rectangular grid (of any dimensions) with specified tiles in the bottom left-hand and top right-hand corners. Sometimes, we consider the more stringent version of CFT in which we restrict consideration to square tilings (i.e., we require $m = n$). The problems CIT and (both versions of) CFT are undecidable, as may be proved easily. We begin with the former.

Lemma 3.7. *The problem* CIT *is co-r.e.-complete under computable reductions.*

Proof. For membership in co-r.e., let an instance (C, H, V, c_α) of CIT be given. Consider the tree whose vertices on level $n \geq 1$ are the (C, H, V)-tilings of $\mathbb{N}^2_n$ such that $(0, 0)$ is assigned the colour c_α, and whose edges are the pairs (t_n, t_{n+1}) such that t_{n+1} tiles $\mathbb{N}^2_{n+1}$, and t_n is the restriction of t_{n+1} to $\mathbb{N}^2_n$. Since the number of vertices on level any n is finite, this tree is finitely branching. Moreover, if it contains an infinite path, then that infinite path defines a tiling of $\mathbb{N}^2$ with c_α assigned to $(0, 0)$. Thus, if there is no such tiling, then the tree is finite by König's infinity lemma, i.e., there exists some n for which there is no tiling of $\mathbb{N}^2_n$ with the bottom-left corner tailed by c_α. Conversely, if there is no tiling of $\mathbb{N}^2_n$ with the bottom left-hand square tiled by c_α, then then there is certainly no such tiling of the whole of $\mathbb{N}^2$. But tilings of $\mathbb{N}^2_n$ can enumerated and checked one by one.

For co-r.e.-hardness, we use the fact that HALTING, namely, the problem of deciding whether a given *deterministic* Turing machine terminates on input ϵ (the empty string), is r.e.-complete. We reduce HALTING to the complement of CIT. Let a Turing Machine $M = (A, Q, s_0, T)$ be given (i.e. with alphabet A, set of states Q, start state s_0 and transition table T). We compute an instance (C, H, V, c_α) of CIT and show that there is a (C, H, V)-tiling of $\mathbb{N}^2$ in which $(0, 0)$ is assigned to c_α if and only if the computation of M on input ϵ runs for ever. The putative tiling of $\mathbb{N}^2$ can be thought of as a 'movie' of that non-terminating run, with the jth row constituting a snap-shot of the tape at time-step j. The tiles C are triples (a, s, η) where a is a symbol, s a state and η a member of the set $\{--, -, 0, +, ++\}$. Assigning tile (a, s, η) to position (i, j) is to be interpreted as stating that, at at time-step j, the ith tape square contains symbol a, the current state is s, and the current position of the head is given by η. In regard to the last of these, $\eta = 0$ means that the head is over square i, $\eta = -$ means that it is over square $i + 1$ (i.e. just to the right), and $\eta = --$ means that it is over square i' for some $i' \geq i + 2$ (i.e. further to the right), with $+$, $++$ interpreted symmetrically. It is then easy to devise relations H and V to force any tiling to represent a run of M. The relation V deals with the effects of the transitions in T. To encode the active effects of transitions, we ensure that V contains a pair $\langle (a, s, 0), (b, s', -) \rangle$ only if T contains the transition $\tau = \langle a, s, b, s', +1 \rangle$ (if the head is reading a while the machine is in state s, then write b, transition to state s', and move the head right); and similarly, *mutatis mutandis*, for $\eta = 0$ or $\eta = -1$. To encode the passive effects of transitions, we ensure that if $\eta \neq 0$, then V contains a pair $\langle (a, s, \eta), (a', s', \eta') \rangle$ only if $a = a'$ (if the head is not over the current square, it's contents do not change). The relation H encodes control information regarding symbol η. Thus, for example, if η is either $-$ or $--$, we ensure that H contains a pair $\langle (a, s, \eta), (a', s', \eta') \rangle$ only if $s' = s$ and $\eta' = --$, and that H contains a pair $\langle (a, s, -), (a', s', \eta') \rangle$ only if $s' = s$ and $\eta' = 0$, and so on (half a dozen or so similar conditions). Finally, to ensure the zeroth row represents the initial condition, we ensure that, if a is either $\triangleleft$ or $\sqcup$ (left-marker or blank, respectively), then H contains a pair

$\langle(a, s_0, \eta), (a', s_0, \eta')$ only if $a' = \sqcup$, and we define c_α to be the tile $\langle(\triangleleft, s_0, 0)$. It is routine to check that the run of M on input ϵ is non-terminating if and only if the whole quadrant is tiled. Note that, if M does terminate, the tiling gets 'stuck' when we encounter a terminating configuration. □

Lemma 3.8. *The problem* CFT *is r.e.-complete under computable reductions. The problem remains r.e.-hard even under the restriction to square tilings.*

Proof. Membership in r.e. is immediate, since (C, H, V)-tilings of $\mathbb{N}_n \times \mathbb{N}_m$ can be enumerated and checked one by one.

To show r.e.-hardness, we reduce Halting to CFT, proceeding in an almost identical way to Lemma 3.7. Let a Turing Machine $M = (A, Q, s_0, T)$ be given. (It does not even matter here that M is deterministic.) Without loss of generality, we may assume that there is a unique state s_* such that T contains no enabled transitions in that state with the head reading $\triangleleft$. (If not, add a new state s_* and additional transitions $(s, \triangleleft, s_*, \triangleleft, 0)$ as necessary.) We construct an instance $(C, H, V, c_\alpha, c_\omega)$ of CFT and show that M terminates on input ϵ if and only if there is a (C, H, V)-tiling of $\mathbb{N}_m \times \mathbb{N}_n$ with c_α assigned to $(0, 0)$ and c_ω assigned to $(m-1, n-1)$. Here, we take the tiles to be quintuples $(a, s, \eta, \rho, \upsilon)$ where (a, s, η) are as in the proof of Lemma 3.7 and ρ, υ are Boolean values indicating, respectively, whether the tile belongs in the right-most column or the upper-most row. The relations H and V are defined almost exactly as in (3.7), except that some more control information is encoded using the components ρ and υ. First, we insist that a pair $\langle(a, s, \eta, \rho, \upsilon), (a', s', \eta', \rho', \upsilon')\rangle$ is in V only if $\rho = \rho'$, and in H only if $\upsilon = \upsilon'$. We also insist that a tile $(a, s, \eta, \rho, \top)$ is not the first member of *any* pair of V (such tiles may only occupy squares on the top row), and similarly, $(a, s, \eta, \top, \upsilon)$ is not the first member of *any* pair of H. To ensure that the tiling represents an accepting computation, we ensure the only tiles of C of the form $(\triangleleft, s, \eta, \top, \upsilon)$ are such that $s = s_*$ and $\eta = 0$ (i.e. M has accepted the empty string). We set c_α to be the tile $\langle(\triangleleft, s_0, 0)$ as before, and c_ω to be the tile $\langle(\sqcup, s, --, \top, \top)$. It is easy to see that $(C, H, V, c_\alpha, c_\omega)$ has the advertised properties.

To prove the final statement, we reduce the general problem to the problem for square tilings, using additional colours to pad out rectangles into squares; the details are left to the reader. □

We know from Proposition 1.3 that there exists a *universal* Turing machine—i.e. one which takes a description of another Turing machine M together with a string x over the alphabet of M as input, and terminates if and only if M terminates on input x, and leaving the same output on the tape as M would have. This fact yields a variant form of the problem CIT. Suppose we have some tiling t of $\mathbb{N}^2$, and $n \geq 0$. We refer to the sequence $t(0, 0), \ldots, t(0, n-1)$ of colours in the bottom left-hand corner as the *initial condition* (of length n) of this tiling. Now fix a tiling system (C, H, V), and consider the following *parametrized* infinite tiling problem, PIT:

PIT$\langle C, H, V\rangle$

Given: A word c over the alphabet C of length $n \geq 0$:
Return: Yes if there is a (C, H, V)-tiling t of $\mathbb{N}^2$ with initial condition c;
No otherwise.

By taking (C, H, V) to be the tiling system describing—in the fashion of the proof of Lemma 3.7—the operation of the *universal* Turing machine, we immediately have:

Corollary 3.9. *There exists a tiling system (C, H, V) such that the problem PIT$\langle C, H, V\rangle$ is is co-r.e.-complete.*

An analogous parametrization of the constrained *finite* tiling problem, CFT, is also possible; details are left to the reader to complete. We seldom require the parametrized forms of these tiling problems; they will in fact appear only in Ch. **??**.

We also mention in passing that the problem CIT has an analogue in which the 'corner-constraint' is dropped altogether. That is, we define the problem *unconstrained infinite tiling* (UIT) as follows:

UIT

Given: A tiling system (C, H, V):
Return: Yes if there is a (C, H, V)-tiling of $\mathbb{N}^2$;
No otherwise.

The problem UIT is also undecidable, a subtler result than Lemma 3.7. Indeed, much more can be said about the computability-theoretic properties of tiling problems. (See Bibliographic notes.) Fortunately, we can avoid these complications: Lemmas 3.7 and 3.8 are adequate for all the undecidability results in this book. In particular, they give us easy proofs of the undecidability results reported in the next section.

3.4 The three-variable fragment

We are now in a position to establish some limits on what can reasonably be hoped for in respect of fragments of first-order logic whose satisfiability problem is decidable. Specifically, we consider $\mathcal{FO}^3$, the set of first-order formulas featuring just three logical variables x, y and z. We show that the satisfiability and finite satisfiability problems for this fragment are both undecidable.

Our proof employs a standard device that we we encounter at various points in the sequel. Recall from Sec. 3.1 that a formula is in prenex form if all the quantifiers are at the front. When dealing with prenex-form formulas, it is often useful to eliminate the existential quantifiers in favour of individual constants and function symbols.

Lemma 3.10. *Let φ be a formula in prenex form. We may compute, in time bounded by a polynomial function of $\|\varphi\|$ a formula ψ in prenex form, featuring no existential quantifiers, such that $\varphi \lhd \psi$.*

Proof. Let $\varphi = Q_1 x_1 \dots Q_n x_n \psi$ with ψ quantifier-free. If φ is purely universal, there is nothing to do. Otherwise, let i be the largest integer $(1 \leq i \leq n)$ such that $Q_1, \dots, Q_{i-1}$ are all $\forall$. If $i = 1$, replace each occurrence of x_1 in ψ by a fresh individual constant c_1; otherwise, replace each occurrence of x_i in ψ by the term $f(x_1, \dots, x_{i-1})$, where f is a fresh $(i-1)$-ary function symbol. Letting the result be φ', it is easy to see that $\varphi \triangleleft \varphi'$, and that φ' has one existential quantifier fewer than φ. Repeating this process, obtain the desired formula ψ. □

The process outlined in the proof of Lemma 3.10 is referred to as *Skolemization*, and the constants and function-symbols introduced, as *Skolem constants* and *Skolem functions*.

Theorem 3.11. *The problem* $\text{Sat}(\mathcal{FO}^3)$ *is co-r.e.-complete. The problem* $\text{FinSat}(\mathcal{FO}^3)$ *is r.e.-complete.*

Proof. That $\text{Sat}(\mathcal{FO}^3)$ is co-r.e. follows from the fact that $\text{Sat}(\mathcal{FO})$ is co-r.e. (see Sec. 1.3). Thus, we need only show co-r.e.-hardness, proceeding by reduction from the constrained infinite tiling problem, CIT. Let a tiling system (C, V, H) and a colour $c_\alpha \in C$ be given. We compute an $\mathcal{FO}^3$-sentence φ_T, satisfiable if and only if the CIT-instance $T = (C, H, V, c_\alpha)$ is positive. It follows from Lemma 3.7 that $\text{Sat}(\mathcal{FO}^3)$ is co-r.e.-hard.

To make the formulas in question more readable, we construct a finite set of sentences Ψ_T over a signature Σ featuring a single individual constant 0, a single unary function-symbol, s and one binary predicate for each element of C. Indeed, we may as well regard the elements of C themselves as the binary predicates in question. We then show how s can be eliminated, yielding a an $\mathcal{FO}^3$-sentence φ_T satisfiable if and only if Ψ_T is satisfiable.

We take Ψ_T to contain the following sentence:

$$\forall x \forall z \left(\bigvee_{c \in C} c(x,z) \wedge \bigwedge_{c,d \in C}^{c \neq d} \neg (c(x,z) \wedge d(x,z)) \right) \tag{3.7}$$

As an aide to intuition, think of x and z as integers, and read $c(x,z)$ as "the grid-square with coordinates (x,z) is tiled with colour c. Formula (3.7) then 'says' that each grid-square is tiled with exactly one colour. We further take Ψ_T to contain the following sentences:

$$\forall x \forall z \left(\bigvee_{(c,d) \in H} (c(x,z) \wedge d(s(x),z)) \wedge \bigvee_{(c,d) \in V} (c(z,x) \wedge d(z,s(x))) \right). \tag{3.8}$$

Again, as an aide to intuition, read the term $s(x)$ as denoting $x+1$. Formula (3.8) then 'says' that horizontally neighbouring squares are tiled in accordance with H, and vertically neighbouring squares are tiled in accordance with V. Finally, we take Ψ_T to contain the sentence, $c_\alpha(0,0)$, which 'says' that the origin of the grid is tiled with c_α. This completes the construction of Ψ_T.

If there exists a (C, H, V)-tiling $t : \mathbb{N}^2 \to C$ such that $t(0, 0)$, define the structure $\mathfrak{A}_t$ over the domain $\mathbb{N}$ by setting $0^{\mathfrak{A}_t} = 0$, $s^{\mathfrak{A}_t}(i) = i + 1$, and $\mathfrak{A}_t \models c[i, j]$ if and only if $t(i, j) = c$, for all $i, j \in \mathbb{N}$ and $c \in C$. It is easy to check that $\mathfrak{A}_t \models \Psi_T$. That is: if T is a positive instance of CIT, then Ψ_T is satisfiable. Conversely, suppose $\mathfrak{A} \models \Psi_T$. Define the sequence of elements $a_0, a_1, \ldots$ of A by setting $a_0 = 0^{\mathfrak{A}}$, and $a_{i+1} = (s^{\mathfrak{A}})(a_i)$ for all $i \geq 0$. (There is no requirement that these elements be distinct.) Now define the function $t_{\mathfrak{A}} : \mathbb{N}^2 \to C$ by letting $t_{\mathfrak{A}}(i, j)$ be the unique $c \in C$ such that $\mathfrak{A} \models c[a_i, a_j]$. This is well-defined by (3.7); by (3.8), $t_{\mathfrak{A}}$ is a (C, H, V)-tiling, and from the formula $c_\alpha(0, 0) \in \Psi_T$, it satisfies the corner-constraint c_α. Hence, if Ψ_T is satisfiable, then $\mathcal{T}$ is a positive instance of CIT.

Let ψ be the result of removing all the universal quantifiers from the formulas in Ψ_T, and taking the conjunction. Thus $\bigwedge \Psi$ is logically equivalent to $\forall x \forall z.\psi$. Notice that all occurences of the function-symbol s in ψ have argument x. Let ψ^* be the result of replacing every occurrence of $s(x)$ in ψ by the variable y. Then $\forall x \forall z.\psi$ is the result of Skolemizing the formula $\varphi_T = \forall x \exists y \forall z.\psi^*$. In particular, $\forall x \forall z.\psi$ and φ_T are satisfiable over the same domains. It follows that φ_T is satisfiable if and only if T is a positive instance of CIT. But $\varphi_T \in \mathcal{FO}^3$, and we are done.

Turning now to the finite satisfiability problem, that FinSat($\mathcal{FO}^3$) is r.e. follows simply from the fact that finite models can be enumerated. Thus, we need only show r.e.-hardness, proceeding by computable reduction from the constrained finite tiling problem, CFT. Let a tiling system (C, V, H) and colours $c_\alpha, c_\omega \in C$ be given. We compute an $\mathcal{FO}^3$-sentence $\hat{\varphi}_T$, satisfiable if and only if the CFT-instance $T = (C, H, V, c_\alpha, c_\omega)$ is positive. Again, we proceed indirectly, making use initially of a unary function symbol. Define the set of formulas $\hat{\Psi}_T$ in the same way as Ψ_T except that we replace (3.8) by

$$\forall x \forall z \left(\left(\neg e(x) \to \bigvee_{(c,d) \in H} (c(x, z) \wedge d(s(x), z)) \right) \wedge \right.$$
$$\left. \left(\neg e(x) \to \bigvee_{(c,d) \in V} (c(z, x) \wedge d(z, s(x))) \right) \right), \quad (3.9)$$

and add the formulas

$$\forall x \forall y ((\neg e(x) \to \neg \ell(x, x)) \wedge \ell(x, s(x)) \wedge [\ell(y, x) \to \ell(y, s(x))]) \quad (3.10)$$
$$\forall x \forall z (e(x) \wedge e(z) \to c_\omega(x, z)), \quad (3.11)$$

where e is a new unary predicate and ℓ a new binary predicate. As an aide to intuition, read $e(x)$ as "the finite grid is of size $m = x + 1$" (alternatively: "the largest value of the horizontal or vertical coordinates is $m - 1 = x$"), and read $\ell(x, y)$ as "$x < \min(y, m)$". If there is a (C, H, V)-tiling of $\mathbb{N}_m^2$ for some $m \geq 1$ satisfying the corner-constraints c_α and c_ω, let $A = \{0, \ldots, m - 1\}$, and define $\mathfrak{A}_t$ over A by interpreting the individual constant 0 and the predicates e, ℓ and c (for $c \in C$) as suggested above, and setting $s^{\mathfrak{A}}(i) = \min(i+1, m-1)$. A simple check then shows that $\mathfrak{A}_t \models \hat{\Psi}_T$. Conversely, suppose $\mathfrak{A}$ is a *finite* model of $\hat{\Psi}_T$, and define the sequence $a_0, a_1, \ldots$ as before. Formula (3.9) works in much the same way as (3.8), except that the sequence $a_0, a_1, \ldots$ stops at some a_{i-1} satisfying the predicate e. Indeed, (3.10) guarantees that there exists such an $i \geq 1$, since otherwise, we would have an infinite sequence of distinct elements. Let the least such i be m. Defining $t_{\mathfrak{A}}(i,j)$ to be the unique $c \in C$ such that $\mathfrak{A} \models c[a_i, a_j]$ for all i, j $(0 \leq i, j < m)$ we obtain a (C,H,V)-tiling $t_{\mathfrak{A}} : \mathbb{N}^2_m \to C$. Indeed, since $\hat{\Psi}_T$ contains both $c_\alpha(0,0)$ and (3.11), the corner-constraints are satisfied, and so T is a positive instance of CFT. Converting $\hat{\Psi}_T$ to an $\mathcal{FO}^3$-sentence satisfiable over the same domains proceeds as before. □

3.5 The two-variable fragment: lower bound

We now establish a matching lower bound to Lemma 3.6 by showing that Sat($\mathcal{FO}^2$) (=FinSat($\mathcal{FO}^2$)) is NExpTime-hard. Again, we employ tiling systems on bounded rectangular grids; but this time we consider a rather different sort of problem. We adapt the notion of *initial condition* to the context of finite tilings. Suppose we have some tiling t of the square grid $\mathbb{N}^2_m$, and $0 \leq n \leq m$. We refer to the sequence $t(0,0), \ldots, t(0, n-1)$ of colours in the bottom left-hand corner as the *initial condition* (of length n) of this tiling. Let $f : \mathbb{N} \to \mathbb{N}$ be inflationary ($f(n) \geq n$), let and P be a problem over some alphabet A such that $P \in$ NTime(f). Thus, there is a Turing Machine M, with time-bound f, such that, for all $\mathrm{x} \in A^*$, M has a terminating run on input x if and only if x is a positive instance of P. By the standard encoding of Turing machines runs as tilings of $\mathbb{N}^2$, it is easy to see that there exists a tiling system (C,H,V) and a logarithmic-space computable, length-preserving mapping $\tau : A^* \to C^*$ such that, for all $\mathrm{x} \in A^*$, the square grid $\mathbb{N}^2_{f(|\mathrm{x}|)}$ has a (C,H,V)-tiling with initial condition $\tau(\mathrm{x})$ if and only if $\mathrm{x} \in P$.

This motivates the concept of (*parametrized*) *bounded tiling problems*. Let (C,H,V) be a tiling system and $f : \mathbb{N} \to \mathbb{N}$. Define the problem

BTP$\langle C, H, V, f \rangle$
Given: A finite word c over the alphabet C:
Return: Yes if there is a (C,H,V)-tiling of $\mathbb{N}^2_{f(|\mathsf{c}|)}$ with initial condition c;
No otherwise.

Notice that the tiling system (C,H,V) as well as the function f are *parameters* here: we have a separate problem for each setting of them. That problem is to determine whether a given word c over the alphabet C can be an initial condition (of length $n = f(|\mathsf{c}|)$) for some (C,H,V)-tiling of a square grid of size $f(n)$. From the foregoing discussion, we have

Lemma 3.12. *Let $f : \mathbb{N} \to \mathbb{N}$ be inflationary. Any problem in* NTime(f) *is many-one log-space reducible to some problem of the form* BTP$\langle C, H, V, f \rangle$.

Proof. If $P \in \text{NT}\textsc{ime}(f)$, let M be a non-deterministic Turing machine recognizing P and halting in time f. Construct the tiling system (C, H, V) simulating runs of M as discussed above. Note that any instance x of P can be straightforwardly mapped to an appropriate sequence c of colours from C, regardless of M. □

Hence, to show that a problem P is—say—NExpTime-hard, it suffices to show that any problem of the form $\text{BTP}\langle C, H, V, 2^{f(n)}\rangle$, where $f : \mathbb{N} \to \mathbb{N}$ is a polynomial, can be reduced to P. Actually, since it is known that there are NExpTime-complete problems in $\text{NTime}(2^n)$, it suffices to show that any problem of the form $\text{BTP}\langle C, H, V, 2^n\rangle$ can be reduced to P. (Similar observations apply to other non-deterministic time-complexity classes.) This will be one of our favourite methods for showing NExpTime-hardness in this book.

Lemma 3.13. *The problem* $\text{Sat}(\mathcal{FO}^2)$ *is* NExpTime*-hard.*

Proof. Fix some bounded tiling problem $P = \text{BTP}\langle C, H, V, 2^n\rangle$. Let $\mathsf{c} = c_0, \ldots, c_{n-1}$ be any instance of T. We construct a formula φ_{c} of $\mathcal{FO}^2$ such that φ_{c} is satisfiable if and only if c is a positive instance of P. Before we give the construction in detail, recall the procedure for incrementing a number stored as a bit string. Let k be an integer ($0 \leq k < 2^n - 1$) with standard n-bit representation $d_{n-1}, \ldots, d_0$ (d_0 is least significant). Then $k' = k + 1$ has standard n-bit representation $d'_{n-1}, \ldots, d'_0$, where, for all i $0 \leq i < n$:

$$d'_i = d_i \text{ if and only if } d_i = 0 \text{ for some } j \ (0 \leq j < i).$$

To motivate the definition of φ_{c}, we first assume that $\mathsf{c} = c_0, \ldots c_{n-1}$ is a positive instance of P. Writing $m = 2^n$, let $t : (\mathbb{N}_m)^2 \to C$ be a witnessing tiling. We simultaneously build the formula φ_{c} and a model $\mathfrak{A} \models \varphi_{\mathsf{c}}$.

Let $A = \mathbb{N}_m^2$. We think of the elements of A as positions on an $m \times m$ grid, and treat the elements of C as unary predicates interpreted over $\mathfrak{A}$ as suggested by t:

$$c^{\mathfrak{A}} = \{a \in A \mid t(a) = c\}$$

for all $c \in C$. We further interpret the additonal unary predicates

$$X_0, \ldots, X_{n-1}, Y_0, \ldots, Y_{n-1}$$

as follows. Recalling that, if $a = \langle k, \ell\rangle \in A$, the numbers k, ℓ are in the range $[0, 2^n - 1]$, define

$$\begin{aligned} X_i^{\mathfrak{A}} &= \{\langle k, \ell\rangle \mid \text{ the } i\text{th digit in the binary representation of } k \text{ is } 1\} \\ Y_i^{\mathfrak{A}} &= \{\langle k, \ell\rangle \mid \text{ the } i\text{th digit in the binary representation of } \ell \text{ is } 1\} \end{aligned}$$

for all i ($0 \leq i < n$), where, as usual, the zeroth digit is taken to be the least significant. Now define

$$\begin{aligned}
\epsilon_X(x,y) &:= \bigwedge_{i=0}^{n-1} (X_i(x) \leftrightarrow X_i(y)) \\
\iota_X(x,y) &:= \bigwedge_{i=0}^{n-1} \left((X_i(x) \leftrightarrow X_i(y)) \leftrightarrow \neg \bigwedge_{j=0}^{i-1} X_j(x) \right) \\
\lambda_X(x) &:= \bigvee_{i=0}^{n-1} \neg X_i(x)
\end{aligned}$$

and define ϵ_Y, ι_Y and λ_Y similarly, but with "X" replaced throughout by "Y". Intuitively, $\epsilon_X(x,y)$ 'says' that x and y have the same X-coordinates, $\lambda_X(x)$ that the X-coordinate of x is less than $2^n - 1$, and $\iota_X(x,y)$ that (under that assumption) the X-coordinate of y is one greater than that of x; similarly for Y. Finally, define

$$\begin{aligned}
\eta(x,y) &:= \iota_X(x,y) \wedge \epsilon_Y(x,y) \wedge \lambda_X(x) \\
\nu(x,y) &:= \iota_Y(x,y) \wedge \epsilon_X(x,y) \wedge \lambda_X(x).
\end{aligned}$$

A moment's thought shows that $\mathfrak{A} \models \eta[a,b]$ if and only if b is immediately to the right of a in the grid, and $\mathfrak{A} \models \nu[a,b]$ if and only if b is immediately above a in the grid. Recalling that $t : A \to C$, the following formula is true in $\mathfrak{A}$:

$$\forall x \bigvee_{c \in C} c(x) \wedge \forall x \bigwedge_{c,c' \in C}^{c \neq c'} \neg(c(x) \wedge c'(x)). \tag{3.12}$$

Since $\mathfrak{A}$ contains the origin $(0,0)$, the formula

$$\exists x[(\neg X_0(x) \wedge \cdots \wedge \neg X_n(x)) \wedge (\neg Y_0(x) \wedge \cdots \wedge \neg Y_n(x))] \tag{3.13}$$

is true in $\mathfrak{A}$. Moreover, And since t has initial segment c, the following n-conjunct formula is true in $\mathfrak{A}$:

$$\begin{aligned}
&\forall x([(\neg X_0(x) \wedge \cdots \wedge \neg X_n(x)) \wedge (\neg Y_0(x) \wedge \cdots \wedge \neg Y_n(x))] \to c_0(x)) \wedge \\
&\forall x([(X_0(x) \wedge \cdots \wedge \neg X_n(x)) \wedge (\neg Y_0(x) \wedge \cdots \wedge \neg Y_n(x))] \to c_1(x)) \qquad (3.14) \\
&\cdots ,
\end{aligned}$$

where the ith conjunct $(0 \leq i \leq n-1)$ specifies that the grid position $\langle i, 0 \rangle$ is assigned colour c_i. From the construction of η and ν, the following formulas are true in $\mathfrak{A}$:

$$\forall x\,(\lambda_X(x) \to \exists y.\eta(x,y)) \tag{3.15}$$

$$\forall x\,(\lambda_Y(x) \to \exists y.\nu(x,y))\,. \tag{3.16}$$

Moreover, since t respects the constraints H and V, the following formulas are also true in $\mathfrak{A}$:

$$\bigwedge_{\langle c,c'\rangle \notin H} \forall x \forall y(\eta(x,y) \wedge c(x) \rightarrow \neg c'(y)) \tag{3.17}$$

$$\bigwedge_{\langle c,c'\rangle \notin V} \forall x \forall y(\nu(x,y) \wedge c(x) \rightarrow \neg c'(y)). \tag{3.18}$$

Let φ_{c} be the conjunction of the formulas in (3.12)–(3.18). We have just seen that, if there is a tiling of $\mathbb{N}^2_m$ with initial condition c, then φ_{c} has a model. However, the construction of φ_{c} depends only on the sequence c (and not on the tiling t, if indeed such a tiling exists). Moreover, that construction requires space $O(\log n)$. This is easily seen by observing that, apart from the conjuncts (3.14), the only way in which φ_{c} depends on c is in terms of the quantity $n = |\mathsf{c}|$.

Suppose, then, that φ_{c} has a model, $\mathfrak{A}$. We show that there is a tiling of $\mathbb{N}^2_m$ with initial condition c. Pick an element $a_{0,0}$ which is a witness for x in (3.13), and pick $a_{1,0}, \ldots, a_{m-1,0}$ such that, for all k $(0 \leq k < m-1)$, $a_{k+1,0}$ is a witnesses for y in (3.15) when x takes the value $a_{k,0}$. Similarly, for all k $(0 \leq k < m)$, pick $a_{k,1}, \ldots, a_{k,m-1}$ such that, for all ℓ $(0 \leq \ell < m-1)$, $a_{k,\ell+1}$ is a witnesses for y in (3.16) when x takes the value $a_{k,\ell}$. It is obvious that the elements $a_{k,\ell}$ form a square grid of size m under the relations defined by the formulas η and ν. By (3.12), we can define the function $t : \mathbb{N}^2_m \rightarrow C$ by setting $t(k,\ell)$ to be the unique $c \in C$ such that $\mathfrak{A} \models c[a_{k,\ell}]$. Formula (3.14) guarantees that c is an initial condition of t. Formulas (3.17) and (3.18) guarantee that t respects the constraints H and V. Hence c is a positive instance of P. This completes the reduction. □

Combining Lemmas 3.6 and 3.13, we obtain tight complexity bounds for the satisfiability problem for the two-variable fragment:

Theorem 3.14. Sat($\mathcal{FO}^2$) *is* NExpTime-*complete.*

3.6 The monadic fragment

In Sec. 3.4, we showed the undecidability of Sat($\mathcal{FO}^k$) and FinSat($\mathcal{FO}^k$) for all $k \geq 3$. In this section, we show that the presence of binary relations is essential for this result. In doing so, we reach back to what is arguably the first ever result identifiably directed at a special case of the *Entscheidungsproblem* (see Bibliographical notes). Define the *monadic fragment* of first-order logic, $\mathcal{FO}_{\text{Mon}}$, to be the set of first-order formulas (with equality) over a signature consisting entirely of unary predicates. Thus, individual constants, function-symbols of any arity and predicates of arity greater than 1 are not allowed. (The results below are unaffected by allowing individual constants.) We shall show that $\mathcal{FO}_{\text{Mon}}$, like $\mathcal{FO}^2$, has an exponential-sized model property, and that its satisfiability (= finite satisfiability) problem is NExpTime-complete.

Suppose $\mathfrak{A}$ is a structure interpreting a signature consisting entirely of unary predicates. Recall the notion of the 1-type $\mathrm{tp}^{\mathfrak{A}}[b]$ of an element b of $\mathfrak{A}$, defined in Sec. 3.1. If b, b' are elements of A such that $\mathrm{tp}^{\mathfrak{A}}[b] = \mathrm{tp}^{\mathfrak{A}}[b']$, then the map $f : A \to A$ which exchanges b and b', and leaves all other elements of A unaffected is clearly an automorphism of $\mathfrak{A}$. In particular, if $\bar{a} = a_1, \ldots, a_k$ is a tuple elements of A distinct from both b and b', (so that f is constant on $\bar{a}$) then, for all formulas $\psi(\bar{x}, y)$ over the signature of $\mathfrak{A}$, $\mathfrak{A} \models \psi[\bar{a}, b]$ if and only if $\mathfrak{A} \models \psi[\bar{a}, b']$. We combine this observation with the following lemma, which many readers will recognize as a variant of the Tarski-Vaught test for elementary sub-structures.

Lemma 3.15. *Let $\mathfrak{A}$, $\mathfrak{B}$ be structures with $\mathfrak{A} \subseteq \mathfrak{B}$, and suppose that, for any tuple $\bar{a}$ from A, any formula $\psi[\bar{x}, y]$ involving at most n variables (free or bound), and any $b \in B$ such that $\mathfrak{B} \models \psi[\bar{a}, b]$, there exists $b' \in A$ such that $\mathfrak{B} \models \psi[\bar{a}, b']$. Then, for any tuple $\bar{a}$ from A, and any formula $\varphi(\bar{x})$ involving at most n variables (free or bound), $\mathfrak{A} \models \varphi[\bar{a}]$ if and only if $\mathfrak{B} \models \varphi[\bar{a}]$.*

Proof. A simple structural induction. The conditions of the lemma are required for the case where $\varphi(\bar{x})$ is of the form $\exists y.\psi(\bar{x}, y)$ and we wish to argue that $\mathfrak{B} \models \varphi[\bar{a}]$ implies $\mathfrak{A} \models \varphi[\bar{a}]$. □

We are now in a position to prove a small model property for $\mathcal{FO}_{\mathrm{Mon}}$, much as we did for $\mathcal{FO}^2$ in Lemma 3.6. This time, however, the proof is much simpler.

Lemma 3.16. *Let φ be a satisfiable $\mathcal{FO}_{\mathrm{Mon}}$-formula over a signature Σ of unary predicates, and involving at most n variables. Then φ has a model of size at most $n \cdot 2^{|\Sigma|}$. Thus, $\mathcal{FO}_{\mathrm{Mon}}$ has the finite model property, and* $\mathrm{Sat}(\mathcal{FO}_{\mathrm{Mon}})$ *is in* NExpTime.

Proof. Suppose $\mathfrak{B} \models \varphi$. Define a subset $A \subseteq B$ as follows: for each 1-type realized in $\mathfrak{A}$, select n distinct elements having that 1-type, or all such elements if there are fewer than n. Thus, $|A| \leq n \cdot 2^{|\Sigma|}$. Let $\mathfrak{A} = \mathfrak{B} \restriction A$. Now, if $\bar{a}$ is a k-tuple of elements from A ($k < n$), and $b \in B \setminus A$ such that $\mathfrak{B} \models \psi[\bar{a}, b]$, we can pick $b' \in A$ distinct from $\bar{a}$ such that $\mathrm{tp}^{\mathfrak{A}}[b] = \mathrm{tp}^{\mathfrak{A}}[b']$. Thus there is an automorphism of $\mathfrak{B}$ fixing $\bar{a}$ and taking b to b', whence $\mathfrak{B} \models \psi[\bar{a}, b']$. By Lemma 3.15, $\mathfrak{A} \models \varphi$. □

For the lower bound, there is no more work to do.

Lemma 3.17. *The problem* $\mathrm{Sat}(\mathcal{FO}_{\mathrm{Mon}})$ *is* NExpTime*-hard.*

Proof. The formula φ_{c} constructed in the proof of Lemma 3.13 is in $\mathcal{FO}_{\mathrm{Mon}}$. □

In fact, the formula φ_{c} does not feature the symbol =. Thus, Lemma 3.13 applies even to two-variable, monadic logic without equality. Combining Lemmas 3.16 and 3.17, we obtain tight complexity bounds for the satisfiability problem for the monadic fragment:

Theorem 3.18. $\mathrm{Sat}(\mathcal{FO}_{\mathrm{Mon}})$ *is* NExpTime*-complete.*

3.7 Expressive power

In Sec. 3.2, we showed that limiting attention to formulas with two variables yields a logic with the finite model property, and with satisfiability problem in NExpTime. But what can one say in this fragment? There is no single answer to this question, of course; however, we shall present a very natural semantic characterization of the expressive power of the fragments $\mathcal{FO}^k$ in this section. Readers interested only in (finite) satisfiability can skip it without loss.

We require some notions from model theory. Fix a relational signature Σ, and let $\bar{a}$ be a tuple (of any finite length) of distinct individual constants not occurring in Σ. An *elementary k-type* (*over* Σ) *with parameters* $\bar{a}$ is any maximal consistent set of first-order formulas over the signature $\Sigma \cup \bar{a}$ with free variables among $x_1, \ldots, x_k$. Do not confuse elementary k-types with what we earlier called k-types (see Sec. 3.1): elementary k-types are always infinite; k-types over finite signatures are always finite. A *pointed* structure over a signature Σ is a pair $(\mathfrak{A}, \bar{a})$, where $\mathfrak{A}$ is a structure interpreting Σ and $\bar{a}$ a tuple of distinct elements from $\mathfrak{A}$. We treat the elements $\bar{a}$ as individual constants denoting themselves, so that $(\mathfrak{A}, \bar{a})$ interprets formulas over the signature $\Sigma \cup \bar{a}$. We write $\text{Th}(\mathfrak{A}, \bar{a})$—rather than the more correct $\text{Th}((\mathfrak{A}, \bar{a}))$—to denote the set of true sentences in $(\mathfrak{A}, \bar{a})$. If $(\mathfrak{A}, \bar{a})$ is a pointed structure and Γ an elementary k-type with parameters $\bar{a}$, we say Γ is *realized* in $\mathfrak{A}$ if, for some k-tuple $\bar{b}$ of elements of A, $(\mathfrak{A}, \bar{a}) \models \Gamma[\bar{b}]$. A structure $\mathfrak{A}$ interpreting Σ is *ω-saturated* if, for any tuple $\bar{a}$ from A, every elementary 1-type $\Gamma(x_1)$ with parameters $\bar{a}$ such that $\Gamma(x_1)$ consistent with $\text{Th}(\mathfrak{A}, \bar{a})$ is realized in $(\mathfrak{A}, \bar{a})$. It is a simple matter to prove that, in that case, for every $k \geq 1$, every elementary k-type $\Gamma(\bar{x})$ consistent with $\text{Th}(\mathfrak{A}, \bar{a})$ is realized in $(\mathfrak{A}, \bar{a})$. We remark that saturation is definable for other cardinalities than ω. But ω will do. We need one final notion for the next lemma. If $\mathfrak{A}$ is a structure, then the *elementary diagram* of $\mathfrak{A}$, denoted $\text{elDiag}(\mathfrak{A})$, is the set of sentences $\text{Th}(\mathfrak{A}^*)$, where $\mathfrak{A}^*$ is the expansion of $\mathfrak{A}$ obtained by taking every element of A to be a fresh individual constant, interpreted as itself. It is easy to show that, if $\mathfrak{B} \models \text{elDiag}(\mathfrak{A})$, and $\mathfrak{B}^-$ is the reduct of $\mathfrak{B}$ to the signature of $\mathfrak{A}$, then $\mathfrak{A} \preceq \mathfrak{B}^-$. In fact, if $\Gamma(x_1)$ is an elementary 1-type with parameters $\bar{a}$ such that $\Gamma(x_1)$ is consistent with $\text{Th}(\mathfrak{A}, \bar{a})$, a simple compactness argument shows that $\Gamma(c)$ is consistent with $\text{elDiag}(\mathfrak{A})$, where c is a fresh individual constant, whence $\mathfrak{A}$ has an elementary extension realizing $\Gamma(x_1)$. The key fact we need is:

Lemma 3.19. *Every structure has an ω-saturated elementary extension.*

Proof. We first remark that, if $\mathfrak{A} \preceq \mathfrak{B}_1$ and $\mathfrak{A} \preceq \mathfrak{B}_2$, then there exists a common elementary extension of $\mathfrak{B}_1$ and $\mathfrak{B}_2$. To see this, assume without loss of generality that $B_1 \cap B_2 = A$, and consider the elementary diagrams $\text{elDiag}(\mathfrak{B}_1)$ and $\text{elDiag}(\mathfrak{B}_2)$. The union of these two theories is consistent, since if not, by compactness, there exists $\psi_1 \in \text{elDiag}(\mathfrak{B}_1)$ and $\psi_2(\bar{b}_2) \in \text{elDiag}(\mathfrak{B}_2)$ such that $\models \neg(\psi_1 \wedge \psi_2(\bar{b}_2))$. (Here, $\bar{b}_2$ are all the individual constants from $B_2 \setminus A$ occurring in ψ_2.) Thus, $\mathfrak{B}_2 \models \exists \bar{x}.\psi_2(\bar{x})$, whence $\mathfrak{B}_1 \models \exists \bar{x}.\psi_2(\bar{x})$, whence $\psi_1 \wedge \psi_2$ is satisfiable by re-interpreting the $\bar{b}_2$ as elements of B_1, a contradiction. Thus, $\text{elDiag}(\mathfrak{B}_1) \cup \text{elDiag}(\mathfrak{B}_2)$ has a model, which must be an elementary extension of

$\mathfrak{B}_1$ and $\mathfrak{B}_2$. By repeated applications of this fact, any finite set of elementary extensions of $\mathfrak{A}$ has a common elementary extension, and therefore, by compactness again, any set of elementary extensions of $\mathfrak{A}$ has a common elementary extension.

Let $\mathfrak{A}_0 = \mathfrak{A}$. Having defined $\mathfrak{A}_i$, if $\bar{a}$ is any tuple of distinct elements from $\mathfrak{A}_i$, and $\Gamma(x_1)$ an elementary 1-type with parameters $\bar{a}$ and consistent with $\mathrm{Th}(\mathfrak{A}_i, \bar{a})$, there is an elementary extension of $\mathfrak{A}_i$ realizing $\Gamma(x_1)$. Let $\mathfrak{A}_{i+1}$ be a common elementary extension of these elementary extensions of $\mathfrak{A}_i$ (as $\bar{a}$ and Γ vary over all possible values). Then $\mathfrak{A}_i \preceq \mathfrak{A}_{i+1}$. Letting $\mathfrak{B} = \bigcup_{i \geq 0} \mathfrak{A}_i$, we see that $\mathfrak{A} \preceq \mathfrak{B}$, and $\mathfrak{B}$ is ω-saturated. □

Now let us bring the discussion back to $\mathcal{FO}^k$. Let $\mathfrak{A}$ and $\mathfrak{B}$ be structures interpreting a common relational signature Σ. A *partial isomorphism* from $\mathfrak{A}$ to $\mathfrak{B}$ is a function $f : D \to E$, where $D \subseteq A$, $E \subseteq B$ and $f : \mathfrak{A}{\restriction}D \simeq \mathfrak{B}{\restriction}E$ is a structure isomorphism. We call f *finite* if D (hence E) is; and we call f a k-*isomorphism* if $|D| = |E| \leq k$. A k-*bisimulation* from $\mathfrak{A}$ to $\mathfrak{B}$ is a set F of k-isomorphisms satisfying the following two properties:

1. if $f : D \to E \in F$, and $D' \subseteq A$ with $|D'| \leq k$, then there exist $E' \subseteq B$ and $f' : D' \to E' \in F$ such that $f{\restriction}(D \cap D') = f'{\restriction}(D \cap D')$;

2. if $f : D \to E \in F$, and $E' \subseteq B$ with $|E'| \leq k$, then there exist $D' \subseteq A$ and $f' : D' \to E' \in F$ such that $f^{-1}{\restriction}(E \cap E') = {f'}^{-1}{\restriction}(E \cap E')$.

We say that the pointed structures $(\mathfrak{A}, \bar{a})$ and $(\mathfrak{B}, \bar{b})$, with $|\bar{a}| = |\bar{b}| \leq k$, are k-*bisimilar* if there exists a k-bisimulation F from $\mathfrak{A}$ to $\mathfrak{B}$ and with some $f \in F$ taking $\bar{a}$ to $\bar{b}$. Conditions 1 and 2 in the above definition are typically referred to as the *forth*- and *back*-conditions, respectively, because together they allow one can move back and forth between the structures $\mathfrak{A}$ and $\mathfrak{B}$. The situation is often treated as a game between a 'spoiler', who wants to show that $\mathfrak{A}$ and $\mathfrak{B}$ are different, and a 'duplicator', who wants to show that they are the same. If $\bar{c}$ is a tuple of at most k distinct elements of A and $\bar{d}$ a tuple of at most k distinct elements of B such that $\bar{c}$ and $\bar{d}$ look the same (in their respective structures), the spoiler chooses one of the structures—say $\mathfrak{A}$—throws away some elements of $\bar{c}$ and chooses some new elements of A to form a collection $\bar{c}'$ (with cardinality at most k); the duplicator must then respond by making matching choices in the other structure—in this case $\mathfrak{B}$—so that the resulting tuples still look the same in their respective structures. The back-and-forth conditions just say that, if we start with some $f \in F$, the duplicator can keep going whatever the spoiler does. A first-order formula $\varphi(\bar{x})$ with at most k free variables is *invariant* under k-bisimulations if, whenever $(\mathfrak{A}, \bar{a})$ and $(\mathfrak{B}, \bar{b})$ are k-bisimilar, $\mathfrak{A} \models \varphi[\bar{a}]$ implies $\mathfrak{B} \models \varphi[\bar{b}]$.

The critical observation is that, for ω-saturated structures, the relation of satisfying the same formulas of $\mathcal{FO}^k$ *is* a k-bisimulation. More formally, let $(\mathfrak{A}, \bar{a})$ and $(\mathfrak{B}, \bar{b})$ be pointed structures with $|\bar{a}| = |\bar{b}| \leq k$. We write $(\mathfrak{A}, \bar{a}) \equiv^k (\mathfrak{B}, \bar{b})$ if, for every $\mathcal{FO}^k$-formula $\psi(\bar{x})$ of the appropriate arity, $\mathfrak{A} \models \psi[\bar{a}] \Rightarrow \mathfrak{B} \models$

$\psi[\bar{b}]$. (Since $\mathcal{FO}^k$ is closed under negation, $\Rightarrow$ could of course be replaced by $\Leftrightarrow$.)

Lemma 3.20. *If $\mathfrak{A}$ and $\mathfrak{B}$ are ω-saturated structures such that $(\mathfrak{A}, \bar{a}) \equiv^k (\mathfrak{B}, \bar{b})$, then $(\mathfrak{A}, \bar{a})$ and $(\mathfrak{B}, \bar{b})$ are k-bisimilar.*

Proof. Let $\bar{c} = c_1, \dots c_\ell$ be any tuple of distinct elements of A, and $\bar{d} = d_1, \dots d_\ell$ any tuple of distinct elements of B, with $\ell \leq k$. If $(\mathfrak{A}, \bar{c}) \equiv^k (\mathfrak{B}, \bar{d})$, then the mapping $c_i \mapsto d_i$ $(1 \leq i \leq \ell)$ is certainly a k-isomorphism. Let F be the set of all k-isomorphisms from $\mathfrak{A}$ to $\mathfrak{B}$ constructed in this way. Then F is non-empty: in particular, it contains a k-isomorphism taking $\bar{a}$ to $\bar{b}$. We claim that F is a k-bisimulation, from which the lemma follows.

It is immediate that F is closed under subsets, in the sense that, if $(\mathfrak{A}, \bar{c}) \equiv^k (\mathfrak{B}, \bar{d})$, $\bar{c}_1 \subseteq \bar{c}$, and $\bar{d}_1$ is the corresponding subset of $\bar{d}$, then $(\mathfrak{A}, \bar{c}_1) \equiv^k (\mathfrak{B}, \bar{d}_1)$. Thus, to establish the forth- condition for F, it suffices to show that if $(\mathfrak{A}, \bar{c}_1) \equiv^k (\mathfrak{B}, \bar{d}_1)$, and $\bar{c}_2$ is a tuple from A disjoint from $\bar{c}_1$ such that $|\bar{c}_1\bar{c}_2| \leq k$, then there exists a tuple $\bar{d}_2$ from B disjoint from $\bar{d}_1$ such that $(\mathfrak{A}, \bar{c}_1\bar{c}_2) \equiv^k (\mathfrak{B}, \bar{d}_1\bar{d}_2)$. Consider then the pointed structure $(\mathfrak{B}, \bar{d}_1)$, and the set of $\mathcal{FO}^k$-formulas (with individual constants from $\bar{d}_1$) given by

$$\Gamma(\bar{x}_2) = \{\psi(\bar{d}_1, \bar{x}_2) : \psi(\bar{x}_1, \bar{x}_2) \in \mathcal{FO}^k \text{ and } \mathfrak{A} \models \psi[\bar{c}_1, \bar{c}_2]\}.$$

We first claim that $\Gamma(\bar{x}_2)$ is consistent with $\mathrm{Th}(\mathfrak{B}, \bar{d}_1)$. For suppose not. By compactness, there exists a finite subset of $\Gamma(\bar{x}_2)$ not consistent with $(\mathfrak{B}, \bar{d}_1)$. Denoting the conjunction of this finite subset by $\gamma(\bar{d}_1, \bar{x}_2)$, we have $(\mathfrak{B}, \bar{d}_1) \models \neg\exists\bar{x}_2.\gamma(\bar{d}_1, \bar{x}_2)$. On the other hand, by construction of Γ, we have $\mathfrak{A} \models \gamma[\bar{c}_1, \bar{c}_2]$ and thus $(\mathfrak{A}, \bar{c}_1) \models \exists\bar{x}_2.\gamma(\bar{c}_1, \bar{x}_2)$, contradicting the assumption that $(\mathfrak{A}, \bar{c}_1) \equiv^k (\mathfrak{B}, \bar{d}_1)$. Since $\Gamma(\bar{x}_2)$ is consistent with $\mathrm{Th}(\mathfrak{B}, \bar{d}_1)$, and $\mathfrak{B}$ is ω-saturated, there exists a tuple $\bar{d}_2$ such that $\mathfrak{B} \models \Gamma[\bar{d}_1, \bar{d}_2]$. But then $(\mathfrak{A}, \bar{c}_1\bar{c}_2) \equiv^k (\mathfrak{B}, \bar{d}_1\bar{d}_2)$ as required. This establishes the forth-condition for F; the back-condition is established symmetrically. □

We are now in a position to give our semantic characterization of the expressive power of $\mathcal{FO}^k$.

Theorem 3.21. *A formula of first-order logic is logically equivalent to a formula of $\mathcal{FO}^k$ if and only if it is invariant under k-bisimulations.*

Proof. The only-if condition is proved by a simple structural induction on formulas, and is left to the reader. For the if-condition, suppose $\varphi(\bar{x})$ is invariant under k-bisimulations. Let Ψ be the set of $\mathcal{FO}^k$-consequences of φ:

$$\Psi = \{\psi \in \mathcal{FO}^k : \ \models \varphi(\bar{x}) \to \psi(\bar{x})\}.$$

It suffices to show that that $\Psi(\bar{x})$ entails $\varphi(\bar{x})$. The theorem then follows by compactness, since some finite subset of $\Psi(\bar{x})$ must entail $\varphi(\bar{x})$, say $\models \psi_1(\bar{x}) \wedge \cdots \wedge \psi_n(\bar{x}) \to \varphi(\bar{x})$, with the entailment $\models \varphi(\bar{x}) \to \psi_1(\bar{x}) \wedge \cdots \wedge \psi_n(\bar{x})$ holding by the definition of Ψ.

Suppose then that $\mathfrak{A} \models \Psi[\bar{a}]$ for some tuple $\bar{a} \subseteq A$. Let $\Gamma(\bar{x})$ be the set of $\mathcal{FO}^k$-variable formulas true in $\mathfrak{A}$ at $\bar{a}$:

$$\Gamma = \{\gamma(\bar{x}) \in \mathcal{FO}^k : \quad \mathfrak{A} \models \gamma[\bar{a}]\}.$$

We claim that $\Gamma \cup \{\varphi\}$ is consistent. For if not, by compactness, there exist $\gamma_1, \ldots, \gamma_n \in \Gamma$ such that $\models \gamma_1(\bar{x}) \wedge \cdots \wedge \gamma_n(\bar{x}) \rightarrow \neg\varphi(\bar{x})$. Writing $\gamma(\bar{x})$ for the conjunction $\gamma_1(\bar{x}) \wedge \cdots \wedge \gamma_n(\bar{x})$, we have $\models \varphi(\bar{x}) \rightarrow \neg\gamma(\bar{x})$, whence $\neg\gamma \in \Psi$, whence $\mathfrak{A} \models \neg\gamma[\bar{a}]$, contradicting $\mathfrak{A} \models \Gamma[\bar{a}]$. But if $\Gamma \cup \{\varphi\}$ is consistent, there exist some structure $\mathfrak{B}$ and tuple $\bar{b}$ of elements from B such that $\mathfrak{B} \models \Gamma[\bar{b}]$ (whence $(\mathfrak{A}, \bar{a}) \equiv^k (\mathfrak{B}, \bar{b})$) and $\mathfrak{B} \models \Gamma[\bar{b}]$. By Lemma 3.19, let $\mathfrak{A}^*$, and $\mathfrak{B}^*$, be ω-saturated elementary extensions of $\mathfrak{A}$ and $\mathfrak{B}$, respectively. Certainly, then $(\mathfrak{A}^*, \bar{a}) \equiv^k (\mathfrak{B}^*, \bar{b})$, whence, by Lemma 3.20, $(\mathfrak{A}^*, \bar{a})$ and $(\mathfrak{B}^*, \bar{b})$ are k-bisimilar. Since φ is, by hypothesis, preserved under k-bisimulations, we have $\mathfrak{A}^* \models \varphi[\bar{a}]$, and hence $\mathfrak{A} \models \varphi[\bar{a}]$, which is what we were required to show. □

Bi-simulation invariance is not the only way to characterize expressive power. We end the chapter with a simple observation that nicely reveals the limited expressiveness of $\mathcal{FO}^2$. Let φ be a satisfiable normal-form $\mathcal{FO}^2$-formula, as given in (3.2), and suppose that $\mathfrak{A} \models \varphi$. Fix some $a \in A$, and let $\pi = \text{tp}^{\mathfrak{A}}[a]$. We call a a *king* if it is the unique element of $\mathfrak{A}$ having 1-type π, and we refer to π as a *royal* 1-type. Note that, if φ contains the conjunct $\forall x \forall y (x = y \vee \neg p(x) \wedge \neg p(y))$, then any 1-type π containing the atom $p(x)$, if realized in $\mathfrak{A}$, must be royal. Thus, φ can ensure that kings exist. But suppose now that $b \in A$ is not a king, and that b' is some element of $A \setminus \{b\}$ having the same 1-type as b in $\mathfrak{A}$. Let b^* be a fresh object ($b^* \notin A$) and define the structure $\mathfrak{A}^*$ over domain $A^* = A \cup \{b^*\}$, by writing $\mathfrak{A}^* \restriction A = \mathfrak{A}$ and setting the remaining 2-types as follows:

$$\text{tp}^{\mathfrak{A}^*}[a, b^*] = \begin{cases} \text{tp}^{\mathfrak{A}^*}[a, b] & \text{if } a \in A \setminus \{b\} \\ \text{tp}^{\mathfrak{A}^*}[b, b'] & \text{if } a = b\ . \end{cases}$$

Thus, b^* is essentially a copy of b, related to b in just the way that b' is. A moment's reflection shows that $\mathfrak{A}^* \models \varphi$. Thus, we have, in effect, a pumping lemma for models of normal-form $\mathcal{FO}^2$-formulas: non-royal elements of $\mathcal{FO}^2$ can be duplicated. This limitation on the expressive power of $\mathcal{FO}^2$ is closely related to the technique of 'circular witnessing' employed in Lemma 3.5. We shall employ this idea in Ch. **??**. Here we employ it for a quite different purpose.

The *spectrum* of a formula φ is the set of cardinalities of the finite models of φ. A subset $S \subseteq \mathbb{N}^m$ is *co-finite* if $\mathbb{N} \setminus S$ is finite.

Theorem 3.22. *The spectra of $\mathcal{FO}^2$-formulas are exactly the finite and co-finite subsets of $\mathbb{N} \setminus \{0\}$.*

Proof. We show that the spectrum of any $\mathcal{FO}^2$-formula φ is finite or co-finite. By Lemma 3.3, we may confine attention to normal form formulas, and obviously, when considering the spectrum of φ, we may confine attention to structures interpreting the signature of φ only. If φ has any (finite) model in which some element is not a king, then, by duplicating that element any number of times, we obtain models of all larger finite cardinalities, so that the spectrum of φ is cofinite. On the other hand, if all elements of all models of φ are kings, then there is certainly a finite bound on the sizes of those models. Conversely, if $S \subseteq \mathbb{N}\backslash\{0\}$ is finite or cofinite, it is a simple matter to construct an $\mathcal{FO}^2$-formula φ whose spectrum is S. The details are left to the reader. □

Concluding remarks

The principal result of this chapter is Theorem 3.14, which states that the satisfiability problem (= finite satisfiability problem) for the two-variable fragment of first-order logic is NExpTime-complete, and which serves as a springboard for Parts II and III of this book. Since this seemed an appropriate juncture, we also took the opportunity to prove Theorem 3.18, which gives corresponding results for the monadic fragment of first-order logic, arguably the first example of a non-trivial fragment of first-order logic for which the *Entscheidungsproblem* was ever settled. (See Bibliographic notes.) We also obtained a semantic characterization of the expressive power of the fragments $\mathcal{FO}^k$, in Theorem 3.21.

But more importantly, perhaps, we have introduced four basic techniques that will recur throughout this book: (i) the use of normal forms as exemplified in Lemma 3.3; (ii) the technique of replacing (sub-) structures with 'small' equivalents given in Lemma 3.5, and which may be used to obtain upper complexity bounds; (iii) the use of infinite and finite tiling problems to establish the undecidability results, as in Theorem 3.11; and (iv) the use of bounded tiling problems to establish lower complexity bounds for decidable satisfiability (and finite satisfiability) problems, as in Lemma 3.13.

Theorem 3.11 shows us that, in terms of limiting the numbers of variables, $\mathcal{FO}^2$ is the end of the line: we cannot hope to find more decidable satisfiability (or finite satisfiability) problems in that direction. But, as we shall see, there are other ways of achieving decidability; and it is to some of these that we turn in the remaining two chapters of Part 1.

Exercises

1. Let φ be a formula of first-order logic not involving the Boolean connective $\leftrightarrow$. Show that we can compute, in time bounded by a polynomial function of $\|\varphi\|$, a first-order formula ψ in prenex form, logically equivalent to φ.

2. Complete the construction of Ψ in the proof of Lemma 3.4 and fill in all remaining details of the proof.

3. Give a careful proof of Lemma 3.12.

4. Show that the satisfiability and finite satisfiability problems for $\mathcal{FO}^2$ extended with a single unary function symbol (even without equality) are undecidable.

5. Show that the fragment $\mathcal{FO}_{\mathrm{Mon}}$ retains a small model property even when individual constants are allowed.

Bibliographic notes

The fragment $\mathcal{FO}^2$ has an intriguing history. Lemma 3.3, which is due to D. Scott [70] in 1962, reduces the problem $\mathrm{Sat}(\mathcal{FO}^2)$ to the satisfiability problem for the so-called *Gödel fragment* with equality: the set of first-order formulas in prenex-form having quantifier prefix matching $\exists^*\forall\forall\exists^*$. K. Gödel [27] had shown in 1933 that this fragment (*without* equality) has the finite model property, and is thus decidable. Unfortunately, Gödel also claimed, in the very last sentence of his paper, that adding equality would not affect this result, a claim which was only later shown to be false by W. Goldfarb [28] in 1984. (See Secs. **??** and **??** in this book.) Relying on Gödel's incorrect assertion, Scott claimed to have a proof that $\mathcal{FO}^2$ is decidable; what he actually showed was the decidability of satisfiability for the sub-fragment of $\mathcal{FO}^2$ without equality. That the full two-variable fragment does indeed have the finite model property was eventually established in 1975 by M. Mortimer [59]. Interestingly, Mortimer's paper contains no hint that Gödel's claim might be suspect, or that Scott's reduction must therefore be confined to $\mathcal{FO}^2$ without equality. The tight NExpTime upper bound was first obtained by E. Grädel, P. Kolaitis and M. Vardi [31]. Our Lemma 3.5 employs essentially the same proof tactic, originally generalized by E. Kieroński and M. Otto [47] and then (slightly) further by E. Kieroński, J. Michaliszyn, I. Pratt-Hartmann and L. Tendera [46].

The use of constrained domino problems to prove undecidability for logical fragments was pioneered by H. Wang [77], who realized that it is often more convenient to code runs of Turing machines indirectly, via tilings, relying on Lemma 3.7. (See also the Bibliographic notes to Ch. 5.) The undecidability of the *un*constrained infinite tiling problem defined in Sec. 3.3 (here called UIT) was first shown by R. Berger [12]. Our proof of Theorem 3.11 re-uses a construction from [15]; but this sort of argument is now standard. It turns out that undecidability proofs for satisfiability and finite satisfiability problems can be obtained rather more elegantly by adopting a more sophisticated approach to tilings. Say that a tiling of $\mathbb{N}^2$ if *periodic* is it consists of a repeated rectangular $n \times m$ block, and define the *unconstrained periodic tiling problem* (PerUIT) as follows: given a tiling system (C, H, V), determine if there is a periodic (C, H, V)-tiling of $\mathbb{N}^2$. It can be shown that PerUIT and the complement of UIT are *recursively inseparable*: there is no Turing machine which, when given *either* a positive instance of PerUIT *or* a negative instance of UIT, can reliably determine which of these it is. This result typically allows 'simultaneous' proofs of undecidability of both satisfiability and finite satisfiability (see E. Börger, E. Grädel and Y. Gurevich [15, Ch. 3]). In practice, however, giving separate proofs in for the satisfiablity and finite satisfiability problems involves relatively little extra work, and allows us to rely on the more perspicuous Lemmas 3.7 and 3.8, concerning *constrained* tiling problems. The study of lower complexity bounds for decidable fragments of logic goes back to H. Lewis [13]. The proof of the complexity lower bound of Lemma 3.13 using finite tilings is due to M. Fürer [24].

The finite model property for $\mathcal{FO}_{\mathrm{Mon}}$ was originally established by L. Löwenheim in 1915 [53], probably the first non-trivial fragment of first-order logic for which such a result was obtained. Löwenheim's paper is extremely hard to read; however, D. Hilbert and W. Ackermann give a much clearer presentation in the *Grundzüge* [38, pp. 77 ff.].

Our proof of Theorem 3.21 is taken directly from H. Andréka, I. Németi and J. van Benthem [3, p. 236]. The background material on ω-saturation is capable of considerable generalization; see, for example, W. Hodges [40, p. 490]. The very compact treatment here follows B. Poizat [64, p. 57].

Chapter 4

Guards

Most quantification encountered in everyday talk is *relativized* by means of some noun or noun-phrase: thus, we say that all *men* are mortal, that every *boy* loves some *girl*, and so on. The realization that such statements could be understood as conditional assertions about the *entire universe*—as in the familiar gloss "For every thing x, *if* x is a man, *then* x is mortal"—arguably represents, along with the accompanying apparatus of logical variables, one the of great conceptual strides in the history of logic. Paradoxically, recent developments in the field of compuational logic, in part reverse this development. For, when using logic to describe structured data in computing applications, the instances of quantification that most naturally arise are again relativized—though in a subtly different way from the patterns encountered in natural language. The key insight here is that, by instituting quantifier relativization in first-order logic along tightly controlled lines, the satisfiability problem becomes decidable. That is the subject of the present chapter.

More specifically, we consider those first-order formulas in which universal quantifiers are applied only to a conditional formula where the antecedent of the conditional is an atomic formula featuring all the free variables of the formula in question. That is, all universal quantification has the form $\forall \bar{x}(A \rightarrow \psi)$, where A is an atomic formula featuring all the free variables of ψ. Similarly, all existential quantification has the form $\exists \bar{x}(A \wedge \psi)$ with the same conditions on A and ψ. The atomic formulas A in this context are referred to as *guards*, and the quantification is said to be *guarded*. The following feature only guarded quantification:

Some artist admires only beekeepers
$\exists x(\text{artst}(x) \wedge \forall y(\text{adm}(x,y) \rightarrow \text{bkpr}(y)))$

Every artist envies every bekeeper he admires
$\forall x(\text{artst}(x) \rightarrow \forall y(\text{adm}(x,y) \rightarrow (\text{bkpr}(y) \rightarrow \text{env}(x,y))))$.

The following formulas feature non-guarded quantification:

Every artist admires every beekeeper
$\forall x(\text{artst}(x) \to \forall y(\text{bkpr}(y) \to \text{adm}(x, y)))$

Some artist envies every bekeeper he does not admire
$\exists x(\text{artst}(x) \wedge \forall y(\text{bkpr}(y) \wedge \neg\text{adm}(x, y) \to \text{env}(x, y)))$.

The *guarded fragment*, here denoted $\mathcal{G}$, is the set of all function-free, first-order formulas (but with individual constants and equality both allowed) in which all quantification is guarded. A formal definition will be given presently. The k-variable *guarded fragment*, denoted $\mathcal{G}^k$, is the subset of $\mathcal{G}$-formulas containing no more than k variables (free or bound). That is: $\mathcal{G}^k = \mathcal{G} \cap \mathcal{FO}^k$.

It is worth outlining in slightly more detail the connection to structured data mentioned above. The relational database paradigm constitutes by far the most popular methodology for mass information storage and retrieval in present-day computing. A *relational database* is a collection of named *tables*. Each table in the database has a small collection of named *columns* (which we should think of as being immutable), and a—typically very large (and mutable)—collection of *rows*. Each of these rows specifies a datum, or *value*, for each of the named columns. In practice, data values may be of various types, but we shall assume, for simplicity, that all values are (uninterpreted) character strings. Fig. 4.1 displays a toy example featuring an organization setting. The database has two tables: one named `orgPeople`, with columns `personID`, `firstName` and `lastName`, and another named `orgMgmnt`, with columns `empID`, `mgrID`. Each row in `orgPeople` lists some person known to the organization, identified by the that row's `personID`-value, and recording his first and last names as values in the appropriate columns. Each row in `orgMgmnt` lists a separate employee (assumed to be a person known to the organization), again identified by the that row's `personID`-value, and records that person's manager, similarly identified. Altogether, the database records five persons, of whom four are employees. Jean Dupont is managed by Mario Rossi, who is in turn managed by Torsten Jung (who manages himself), while John Brown is managed by Torsten Jung; Alicja Kowalska (not an employee, but known to the organization) is not listed as being managed by anyone. Other tables might link the individuals mentioned here (identified by the tags `0001`–`0005`) to entries in an addresses table with attributes such as house-number, street name, postal code, or perhaps to tables giving social security numbers or other personal details.

We may wish to write constraints describing the organization of these tables. Thus, for example, we might stipulate that every employee has a manager, lives at some address and earns some salary; the manager of a any employee is classified a manager or an executive, the salary an integer, and the postal code of any address is a valid alphanumeric sequence of some sort. As it turns out, the great majority of these statements are expressible (under natural choices of

orgPeople		
personID	firstName	lastName
0001	Jean	Dupont
0002	Mario	Rossi
0003	Torsten	Jung
0004	Alicja	Kowalska
0005	John	Brown

orgMgmnt	
empID	mgrID
0001	0002
0002	0003
0003	0003
0005	0003

Figure 4.1: Some tables in a relational database.

logical primitives) in the (2-variable)guarded fragment:

$$\forall x(\mathrm{empl}(x) \rightarrow \exists y(\mathrm{manages}(x,y) \wedge \mathrm{empl}(y))) \tag{4.1}$$

$$\forall x(\mathrm{empl}(x) \rightarrow \forall y(\mathrm{manages}(x,y) \rightarrow (\mathrm{mgr}(y) \vee \mathrm{exec}(y)))). \tag{4.2}$$

A considerable proportion of current research into decidable fragments of first-order logic is motivated by the use of such fragments to describe structured data in just this way, a topic to which we shall return in Ch. **??**.

Since the logical variables typically carry little or no information in such applications, a more compact, variable-free syntax is typically preferred. Thus, in the logic known as $\mathcal{ALC}$(a sub-fragment of $\mathcal{G}^2$), formulas (4.1)–(4.2) are written as follows:

$$\begin{aligned}\mathrm{empl} &\sqsubseteq \exists \mathrm{manages} \cdot \mathrm{empl}\\ \mathrm{empl} &\sqsubseteq \forall \mathrm{manages} \cdot (\mathrm{mgr} \sqcup \mathrm{exec}).\end{aligned}$$

The language $\mathcal{ALC}$ is the most basic member of a family of decidable fragments of (mainly first-order) logic collectively known as *description logics.* As a general rule, description logics are based on the two-variable guarded fragment and its extension, the two-variable guarded fragment *with counting* (see Ch. **??**). They have all been developed with a view to obtaining efficient practical algorithms for checking satisfiability and derivative problems. We mention that $\mathcal{ALC}$ is a syntactic variant of the multi-modal logic $\mathrm{K}^{\mathrm{m}}\mathrm{U}$ (see Ch. **??**). While the structure of a company database might, with its few dozens of tables and attributes, hardly benefit from logical formalization, the problem of systematizing and maintaining networks of interrelated concepts represents a significant challenge in many areas of human activity. In medicine, for example, the numerous anatomical features, symptoms, pathologies and treatments form a collection which no human expert can comprehend in its entirety. The SNOMED clinical terminology, widely employed as a standard within healthcare systems, contains over three hundred thousand concepts. Its semantic basis is $\mathcal{ALC}$.

We begin in Sec. 4.1 with a formal definition of the guarded fragment, $\mathcal{G}$, together with its k-variable sub-fragments $\mathcal{G}^k = \mathcal{G} \cap \mathcal{FO}^k$, for $k \geq 2$. Sec. 4.2 gives upper complexity bounds for the satisfiability problems for these fragments. Sec. 4.3 establishes matching lower bounds. The guarded fragment, like the two-variable fragment, possesses the finite model property. However—and somewhat unusually—while it is relatively easy to establish the decidability of the satisfiability problem for $\mathcal{G}$, the finite model property requires a more sophisticated approach; we consider this problem in Sec. 4.4. We return to the satisfiability problem in Sec. 4.5 and Sec. 4.6, showing how it be algorithmically solved with the help of resolution theorem-proving. We round off in Sec. 4.7 with a characterization of the expressive power of $\mathcal{G}$.

4.1 The fragment $\mathcal{G}$

Recall that, if φ is a first-order formula, vars(φ) denotes the list of free variables of φ (in some order). The *guarded fragment* of first-order logic, $\mathcal{G}$, is defined to be the smallest set of function-free first-order formulas satisfying the following conditions:

1. every atomic formula is in $\mathcal{G}$;
2. $\mathcal{G}$ is closed under Boolean connectives;
3. if $\psi \in \mathcal{G}$ and α is an atomic formula such that $\text{vars}(\alpha) \supseteq \text{vars}(\psi)$ and $\bar{x}$ a non-empty tuple of variables such that $\text{vars}(\alpha) \supseteq \bar{x}$, then $\forall \bar{x}(\alpha \rightarrow \psi) \in \mathcal{G}$ and $\exists \bar{x}(\alpha \wedge \psi) \in \mathcal{G}$;
4. if $\psi \in \mathcal{G}$ with $\text{vars}(\psi) = \{x\}$, then $\forall x.\psi \in \mathcal{G}$ and $\exists x.\psi \in \mathcal{G}$.

For $k > 1$, we define the *k-variable guarded fragment*, denoted $\mathcal{G}^k$, to be the subset of $\mathcal{G}$ involving at most the variables $x_1, \ldots x_k$. Thus, $\mathcal{G}^k = \mathcal{G} \cap \mathcal{FO}^k$, for all k. Clearly, $\mathcal{G}^k$ is interesting only when $k \geq 2$.

The formulas α appearing in clause 3 of the definition of $\mathcal{G}$ are called *guards*. The equality predicate is permitted in guards. One motivation for clause 4 of the definition is that the formulas it permits are natural simplifications of formulas with equality guards. Thus, for example, if $\chi(x, y)$ is a quantifier-free formula with the indicated free variables, then, the guarded formula $\forall xy(x = y \rightarrow \chi)$ is equivalent to $\forall x.\chi(x, x)$, which might lack a guard, but is in $\mathcal{G}$ by clause 4. A formula formed without recourse to clause 4 will be called *strictly guarded*. According to the above definition, vacuous quantification is not possible in $\mathcal{G}$; this restriction is simply a technical convenience, and could be relaxed without changing any of the results that follow. Notice that function symbols are not allowed in $\mathcal{G}$ (though individual constants are). We therefore silently assume, until the end of Sec. 4.4, that no function-symbols are present; in Sec. 4.5, they re-appear in an auxiliary role.

Our first task is to prove some easy simplification lemmas for $\mathcal{G}$.

Lemma 4.1. *Let φ be a formula of $\mathcal{G}^k$. We may compute, in time bounded by a linear function of $\|\varphi\|$, a strictly guarded formula of $\mathcal{G}^k$ in which equality does not appear in guards, and which is (finitely) satisfiable if and only if φ is (finitely) satisfiable.*

Proof. By replacing any sub-formula of either of the forms $\exists y(x = y \wedge \chi(x, y))$ or $\forall y(x = y \rightarrow \chi(x, y))$ by the corresponding formula $\chi(x, x)$, we may assume that equality does not appear in guards. Now let q be a fresh unary predicate. Re-write any guarded quantification in φ as follows:

$$\exists \bar{y}(\alpha \wedge \dots) \Rightarrow \exists \bar{y}\left(\alpha \wedge \bigwedge_{y \in \bar{y}} q(y) \wedge \dots\right);$$

$$\forall \bar{y}(\alpha \rightarrow \dots) \Rightarrow \forall \bar{y}\left(\alpha \rightarrow \left(\bigwedge_{y \in \bar{y}} q(y) \rightarrow \dots\right)\right).$$

And re-write any non-guarded quantification in φ as follows:

$$\exists x(\dots) \Rightarrow \exists x(q(x) \wedge \dots); \qquad \forall x(\dots) \Rightarrow \forall x(q(x) \rightarrow \dots).$$

Let the resulting formula be ψ', and let $\psi := \psi' \wedge \exists x.(q(x) \wedge \top)$. Thus, ψ is strictly guarded, and can be computed in linear time. We remark that φ and ψ' have, essentially, the same structure. A simple structural induction shows that, if $\mathfrak{A} \models \varphi$, then we can expand $\mathfrak{A}$ to a structure $\mathfrak{A}^+ \models \psi$ by setting $q^{\mathfrak{A}^+} = A$; likewise, if $\mathfrak{B} \models \psi$, then, taking $A = q^{\mathfrak{B}} \neq \emptyset$, we have $\mathfrak{B} \restriction A \models \varphi$. □

Call a formula *sentential* if it has no free variables.

Lemma 4.2. *Let φ be a strictly guarded sentence of $\mathcal{G}^k$ containing no equality guards over a signature featuring $n \geq 1$ individual constants. We can construct, in time bounded by a polynomial function of $\|\varphi\|$, a strictly guarded sentence φ_0 of $\mathcal{G}^{k+n}$ such that φ and φ_0 are satisfiable over the same domains, and φ_0 contains no equality guards and no individual constants. Moreover, we can ensure that φ_0 has an existential quantifier as its major connective, and contains no proper, sentential, quantified sub-formulas.*

Proof. Let the individual constants of φ be $\bar{c} = c_1, \dots, c_n$ (in some order, without repeats). For every non-equality predicate p occurring in φ, let $\hat{p}$ be a fresh predicate with n additional arguments, and let q_0 be a fresh n-ary predicate. Now let $\hat{\varphi}$ be the result of replacing every non-equality atomic subformula $p(t_1, \dots, t_m)$ of φ by the corresponding atom $\hat{p}(t_1, \dots, t_m, \bar{c})$. It is easy to see that $\hat{\varphi}$ is satisfiable over the same domains as φ: if $\mathfrak{A} \models \varphi$, we may construct $\hat{\mathfrak{A}} \models \hat{\varphi}$ by setting $\hat{\mathfrak{A}} \models \hat{p}[\bar{a}, \bar{d}]$ if and only if $\mathfrak{A} \models p[\bar{a}]$ and $\bar{d} = \bar{c}^{\mathfrak{A}}$; and if $\mathfrak{B} \models \hat{\varphi}$, we may construct $\check{\mathfrak{B}} \models \varphi$ by setting $\check{\mathfrak{B}} \models p[\bar{a}]$ if and only if $\mathfrak{B} \models \hat{p}[\bar{a}, \bar{c}^{\mathfrak{B}}]$. Now treating the symbols $\bar{c}$ in $\hat{\varphi}$ as variables (rather than as individual constants), we see that $\varphi_0 := \exists \bar{c}(q_0(\bar{c}) \wedge \hat{\varphi})$ is satisfiable over the same domains as $\hat{\varphi}$, and hence as φ. Since φ_0 is strictly guarded with no equality guards, so is φ_0. Moreover, all proper quantified sub-formulas of φ_0 involve the free variables $\bar{c}$. □

As with the two-variable fragment, so too with the guarded fragment, it is useful to work with formulas in standard forms. A sentence of $\mathcal{G}$ containing no individual constants is in (*guarded*) *normal form* if it has the shape

$$\exists\bar{x}.\chi(\bar{x})\wedge \bigwedge_{h=1}^{\ell} \forall\bar{x}_h(A_h(\bar{x}_h)\rightarrow\exists\bar{y}_h.\chi_h(\bar{x}_h,\bar{y}_h))\wedge\bigwedge_{h=1}^{m}\forall\bar{z}_h(C_h(\bar{z}_h)\rightarrow\omega_h(\bar{z}_h)), \quad (4.3)$$

where: (i) ℓ and m are non-negative integers; (ii) $\bar{x}$ and the $\bar{x}_h$, $\bar{y}_h$ and $\bar{z}_h$ are non-empty tuples of variables; (iii) the $A_h(\bar{x}_h)$ and $C_h(\bar{z}_h)$ are non-equality atoms with the indicated variables; and (iv) $\chi(\bar{x})$, the $\chi_h(\bar{x}_h,\bar{y}_h)$ and the $\omega_h(\bar{z}_h)$ are quantifier-free formulas with the indicated variables. We refer to $\exists\bar{x}.\chi(\bar{x})$ as the *initial* conjunct of φ, to the $\forall\bar{x}_h(A_h(\bar{x}_h)\rightarrow\exists\bar{y}_h.\chi_h(\bar{x}_h,\bar{y}_h)))$ as the *universal-existential* conjuncts of φ, and to the $\forall\bar{z}_h(C_h(\bar{z}_h)\rightarrow\omega_h(\bar{z}_h))$ as the *universal* conjuncts of φ. Strictly speaking, the formulas χ and χ_h have to incorporate guards for (4.3) to be in $\mathcal{G}$. However, this is not actually required for the ensuing treatment, and we therefore ignore the matter.

Guarded normal form is similar in spirit to Scott normal form encountered in Sec. 3.1. We remind the reader of the notation introduced there: if φ, ψ are first-order sentences, $\varphi\lhd\psi$ means that $\models\psi\rightarrow\psi$, and every structure $\mathfrak{A}$ has an expansion $\mathfrak{A}^+$ such that $\mathfrak{A}\models\varphi[\bar{a}]$ implies $\mathfrak{A}^+\models\psi[\bar{a}]$.

Lemma 4.3. *Let φ_0 be as guaranteed by Lemma 4.2. We can construct, in time bounded by a polynomial function of $\|\varphi\|$, a normal-form formula ψ of $\mathcal{G}^k$, such that $\varphi_0\lhd\psi$.*

Proof. We proceed as in the proof of Lemma 3.1, but this time taking care to ensure that the constructed formula is guarded. Suppose φ_0 has a quantified proper subformula $\theta=\exists\bar{v}(G\wedge\chi)$, with χ quantifier-free, and $\text{vars}(G)=\bar{u}\bar{v}$. Let q be a new unary predicate of arity $|\bar{u}|$, let φ_1 be $\varphi[\theta/q(\bar{u})]$, and let

$$\psi_1 := \forall\bar{u}(q(\bar{u})\rightarrow\exists\bar{v}(G\wedge\chi))\wedge\forall\bar{u}\bar{v}(G\rightarrow(\chi\rightarrow q(\bar{u}))). \quad (4.4)$$

Notice that the variable tuples $\bar{u}$ and $\bar{v}$ are non-empty, because φ contains no proper, sentential, quantified sub-formulas. We claim that $\varphi_0\lhd(\varphi_1\wedge\psi_1)$. Observing that $\psi_1\equiv\forall\bar{u}(q(\bar{u})\leftrightarrow\theta)$, it is immediate that $\varphi_1\wedge\psi_1$ entails φ_0; moreover, any model $\mathfrak{A}$ of φ_0 may be expanded to a model of $\varphi_1\wedge\psi_1$ by interpreting q to be satisfied by a tuple $\bar{a}$ from A if and only if $\mathfrak{A}\models\theta[a]$. Similarly, if φ_0 has a quantified proper subformula $\theta=\forall\bar{v}(G\rightarrow\chi)$, with χ quantifier-free, and $\text{vars}(G)=\bar{u}\bar{v}$, let φ_1 be $\varphi[\theta/\neg q(\bar{u})]$, and set

$$\psi_1 := \forall\bar{u}\forall\bar{v}(G\rightarrow(\neg q(\bar{u})\rightarrow\chi))\wedge\forall\bar{u}(q(\bar{u})\rightarrow\exists\bar{v}(G\wedge\neg\chi). \quad (4.5)$$

Observing that $\psi_1\equiv\forall\bar{u}(\neg q(\bar{u})\leftrightarrow\theta)$, it is immediate that $\varphi_1\wedge\psi_1$ entails φ_0; moreover, any model $\mathfrak{A}$ of φ_0 may be expanded to a model of $\varphi_1\wedge\psi_1$ by interpreting q to be satisfied by a tuple $\bar{a}$ from A if and only if $\mathfrak{A}\not\models\theta[a]$.

Now process φ_1 in the same way to obtain φ_2 and ψ_2, and continue until we reach some sentence φ_m having the form $\exists\bar{x}.\chi$, with χ quantifier-free. Setting

$$\psi := (\varphi_m\wedge\psi_m\wedge\psi_{m-1}\wedge\cdots\wedge\psi_1),$$

we have $\varphi_0 \lhd \psi$. By inspection, ψ is in normal form. □

Lemmas 4.1–4.3 state that, for the purposes of determining (finite) satisfiability in $\mathcal{G}^k$, we may without loss of generality restrict attention to normal-form formulas over purely relational signatures. Indeed, since, as we shall see below (Lemma 4.9), the problems in question are all at least ExpTime-hard, we may further assume that these formulas involve no proposition letters, since, given any formula φ, we can simply enumerate the $2^{\|\varphi\|}$ combinations of truth-values of any proposition letters.

We end these preliminaries with a simple observation that may help to convey a sense of the character of guarded quantification. Suppose φ is a satisfiable guarded formula not containing individual constants. If $\mathfrak{A}$ is any model of φ, let $\mathfrak{A}_1$ and $\mathfrak{A}_2$ be disjoint copes of $\mathfrak{A}$, and form the structure $\mathfrak{B} = \mathfrak{A}_1 \cup \mathfrak{A}_2$ with domain $B = A_1 \cup A_2$ and the interpretations of any predicate p given by $p^{\mathfrak{B}} = p^{\mathfrak{A}_1} \cup p^{\mathfrak{A}_2}$. Thus, $\mathfrak{B}$ consists of two copies of $\mathfrak{A}$ not connected to each other by any predicate. It is simple to see that $\mathfrak{B}$ is also a model of φ. Of course, this is not true for $\mathcal{FO}$ in general: just consider the formula $\forall x(\text{artst}(x) \rightarrow \forall y(\text{bkpr}(y) \rightarrow \text{adm}(x, y)))$. It follows in particular that, if a guarded formula has a model of size n, then it has a model of size kn for all $k \geq 1$, a matter to which we shall return in Ch. **??**.

4.2 The guarded fragment: upper bounds

Let a $\mathcal{G}^k$-formula φ of the form (4.6), over a signature Σ, be given. Grouping the universal conjuncts of φ into a single formula Υ, we may write φ more conveniently as

$$\exists \bar{x}.\chi(\bar{x}) \wedge \bigwedge_{h=1}^{\ell} \forall \bar{x}_h(A_h(\bar{x}) \rightarrow \exists \bar{y}_h.\chi_h(\bar{x}_h\bar{y}_h)) \wedge \Upsilon. \tag{4.6}$$

Recall that $\chi(\bar{x})$ is a quantifier-free formula whose free variables are exactly $\bar{x}$, $A_h(\bar{x}_h)$ an atomic formula whose free variables are exactly $\bar{x}_h$, and each $\chi_h(\bar{x}_h\bar{y}_h)$ a quantifier-free formula whose free variables are among $\bar{x}_h\bar{y}_h$. We may further assume, as just agreed, that φ features no propositional letters; thus, the signature Σ consists only of predicates of arity 1 to k. We describe a procedure which takes φ as input and returns `Y` or `N` according as φ is satisfiable.

Recall the notion of a k-type τ (over the relational signature Σ) for $k \geq 1$: a maximal, consistent set of non-equality Σ-literals in the variables $x_1, \ldots, x_k$. Observe that, by regarding $x_1, \ldots, x_k$ as objects (rather than as variables), we can think of τ as a *structure* over the domain $\{x_1, \ldots, x_k\}$ in the obvious way. Specifically: for any word w over the alphabet $\{x_1, \ldots, x_k\}$ and any predicate $r \in \Sigma$ with arity $|w|$, we take it that $\tau \models r[w]$ just in case the positive literal $r(w)$ is in τ. In this section (Sec. 4.2), we shall refer to an ℓ-type over Σ, for any ℓ $(1 \leq \ell \leq k)$ simply as a *type*. Since each atom occurring in such a type features one predicate and at most k arguments chosen from V_k, there are at most $|\Sigma| \cdot k^k$ such atoms, and therefore at most $2^{|\Sigma|\cdot k^k}$ types. Referring to (4.6), suppose σ is a type and $\bar{w}$ a tuple (repeats allowed) chosen from among the variables of σ, such that $\sigma \models A_h[\bar{w}]$, where $1 \leq h \leq m$. Let us say that a type τ is an *h-witness for $\bar{w}$ in σ* if: (i) every element of $\bar{w}$ is a variable occurring in τ, (ii) $\sigma \restriction \bar{w} = \tau \restriction \bar{w}$, and (iii) $\tau \models \chi_h[\bar{w}, \bar{v}']$ for some tuple $\bar{v}'$ chosen from $\bar{v}$.

The procedure `satGFk(`φ`)` starts by setting T to be the set of all types τ such that $\tau \models \Upsilon$. (Remember: we are regarding types as structures.) By the above-mentioned bound on the number of types, $|T| \leq 2^{|\Sigma|k^k}$. We call any $\sigma \in T$ *bad* if, for some tuple $\bar{w}$ chosen from vars(σ), and for some h ($1 \leq h \leq \ell$), $\sigma \models A_h[\bar{w}]$, but T contains no h-witness for $\bar{w}$ in σ. Now consider each $\sigma \in T$ and remove all bad ones. If any bad σ has been found, repeat this step (with the new value of T), until eventually there are no bad types. Finally, if T contains a type σ such that $\sigma \models \exists \bar{x}.\chi$, return `Y`; otherwise, return `N`. In pseudo-code:

```
1. begin satGFk($\varphi$)
2.     let $T := \{\tau \mid \tau$ a type such that $\tau \models \Upsilon\}$
3.     until $T$ contains no bad types
4.         let $S = \{\sigma \in T \mid \sigma$ is bad$\}$
5.         let $T := T \setminus S$
6.     if $T$ contains a type $\sigma$ such that $\sigma \models \exists \bar{x}.\chi$
7.         return Y
8.     return N
9. end
```

Lemma 4.4. *Let φ be a normal-form $\mathcal{G}^k$-sentence over a purely relational signature Σ. The procedure* `satGFk(`φ`)` *terminates in time $O(f(\|\varphi\|)k^{2k} \cdot 2^{3|\Sigma|k^k})$ for some fixed polynomial f.*

Proof. We may check the condition $\tau \models \Upsilon$ in time $f(\|\varphi\|)$, for some polynomial f. Hence line 2 requires time at most $O(f(\|\varphi\|) \cdot 2^{|\Sigma|k^k})$. Similarly, for fixed σ, τ, $\bar{w}$ and $\bar{z}$, the condition $\tau \models \chi_h[\bar{w}, \bar{z}]$ can be checked in time $f(\|\varphi\|)$. Hence we may check whether τ is an h-witness for $\bar{w}$ in σ in time $O(f(\|\varphi\| \cdot k^k)$, since at most k^k tuples $\bar{z}$ need be considered. Since $|T| \leq 2^{|\Sigma|k^k}$, we may check whether T contains an h-witness for a fixed $\bar{w}$ in σ in time $O(f(\|\varphi\|) \cdot k^k \cdot 2^{|\Sigma|k^k})$. Hence, we may check whether σ is bad in time $O(f(\|\varphi\|) \cdot \ell k^{2k} \cdot 2^{|\Sigma|k^k})$, since at most k^k tuples $\bar{w}$ and ℓ values of h need to be considered. Again, since $|T| \leq 2^{|\Sigma|k^k}$, we may select a bad σ from T (or verify that there is none) in time $O(f(\|\varphi\|) \cdot \ell k^{2k} \cdot 2^{2|\Sigma|k^k})$. Moreover, the number of removals from T is likewise bounded by $2^{|\Sigma|k^k}$), from which it follows that the `until`-loop (lines 3–5) runs in time $O(f(\|\varphi\|) \cdot \ell k^{2k} \cdot 2^{3|\Sigma|k^k})$, which of course dominates the time taken in lines 2 and 6–8. The parameter ℓ can be absorbed into f. □

Thus, the running time of `satGFk(`φ`)` is doubly exponentially bounded in $\|\varphi\|$, and only singly exponentially bounded in $\|\varphi\|$ if the number of variables k is treated as a constant. We now show that `satGFk(`φ`)` delivers the expected results.

Lemma 4.5. *Let φ be a normal-form $\mathcal{G}^k$-sentence. If φ is satisfiable, then* `satGFk(`φ`)` *returns* `Y`*.*

Proof. Suppose $\mathfrak{A} \models \varphi$, and let T_0 be the set of types realized in $\mathfrak{A}$. We claim that the invariant $T \supseteq T_0$ is maintained throughout the execution of the `until`-loop (lines 3–5). This is obvious when T is initialized in line 2, since $\mathfrak{A} \models \Upsilon$. Now suppose some $\sigma(\bar{u}) \in T_0$ is removed from T, and consider the first such removal. For some tuple $\bar{w}$ chosen from vars(σ), and for some h $(1 \leq h \leq \ell)$, we have $\sigma \models A_h[\bar{w}]$, while $\bar{w}$ has no h-witness in T. Since σ is realized in $\mathfrak{A}$, there is a tuple $\bar{a}$ such that $\mathfrak{A} \models A_h[\bar{a}]$, and hence, since $\mathfrak{A} \models \forall \bar{x}_h(A_h(\bar{x}) \rightarrow \exists \bar{y}_h.\chi_h(\bar{x}_h\bar{y}_h))$, a tuple $\bar{b}$ such that $\mathfrak{A} \models \chi_h[\bar{a}, \bar{b}]$. But then $\bar{w}$ certainly has a witness in T_0 and hence in T, because σ is by hypothesis the first removal of a type in T_0, a contradiction. Hence, at the end of the `until`-loop, T includes T_0. And since $\mathfrak{A} \models \exists \bar{x}.\chi$, there is a $\sigma \in T_0$ such that $\sigma \models \exists \bar{x}.\chi$. □

Lemma 4.6. *Let φ be a normal-form $\mathcal{G}^k$-sentence. If* `satGFk(`φ`)` *returns* `Y`*, then φ is satisfiable.*

Proof. Consider the value of the variable T at the end of the execution. The initial assignment (line 2) ensures that $\tau \models \Upsilon$ for all $\tau \in T$. The `until`-loop (lines 3–5) ensures that T contains no bad types. Moreover, the final test (line 6) ensures that there exists some $\tau_0 \in T$ such that $\tau_0 \models \exists \bar{x}.\chi$. We proceed to build a (possibly infinite) tree V based on T; each vertex μ of V will be an isomorphic copy of some type $\sigma_\mu \in T$: we write $f_\mu : \sigma_\mu \simeq \mathfrak{A}_\mu$. Furthermore, we shall partition the domain A_μ of this structure into sets of *new* and *old* objects.

We define V level by level. Starting with the root, μ_0, we let A_{μ_0} be τ_0 (regarded as a structure), set f_{μ_0} to be the identity, and declare all the variables of τ_0 to be new elements of A_{μ_0}. (Incidentally, μ_0 is the only vertex for which we shall allow the set of old objects to be empty.) Now suppose that the vertex μ has been constructed, with $f_\mu : \sigma_\mu \rightarrow \mathfrak{A}_\mu$. For all tuples $\bar{a}$ chosen from the domain of A_μ, and all h $(1 \leq h \leq m)$, if $\mathfrak{A}_\mu \models A_h[\bar{a}]$, let $\tau \in T$ be a witness for $\bar{w} = f_\mu^{-1}(\bar{a})$ in σ_μ. If $\tau \subseteq \sigma_\mu$, there is nothing to do; otherwise, define a mapping f_ν on τ by taking f_ν to be the same as f_μ on the variables $\bar{w}$, and mapping all other variables of τ to some fresh objects. Let $\mathfrak{A}_\nu$ be the resulting structure, and declare the elements $\bar{a} = f_\nu(\bar{w})$ to be the old elements of A_ν. Since $\sigma_\mu \restriction \bar{w} = \tau \restriction \bar{w}$, we have $\mathfrak{A}_\mu \restriction \bar{a} = \mathfrak{A}_\nu \restriction \bar{a}$, and since $\tau \not\subseteq \sigma$, the set of new elements of A_ν is non-empty. Finally, we add ν to the tree, as a daughter of μ. Thus, for any vertex μ of T, each daughter ν of μ agrees with μ on the old elements of ν, while the new elements of ν appear only in ν and (possibly) the descendants of ν. As we might say: daughters agree with their mothers, but are otherwise free.

We now build a structure $\mathfrak{A}$ as follows. The domain A is the union $\bigcup_{\mu \in V} A_\mu$, and for any vertex μ, we set $\mathfrak{A} \restriction A_\mu = \mathfrak{A}_\mu$. Since daughters agree with their mothers, but are otherwise free, these assignments are evidently consistent. Finally, if $\bar{a}$ is any tuple from A whose type has not been assigned by this process, we take $\bar{a}$ to satisfy no predicates (of arity $|\bar{a}|$) at all. This completes the definition of $\mathfrak{A}$. We claim that $\mathfrak{A} \models \varphi$. Indeed, since $\tau_0 \models \exists \bar{x}.\chi$, we have $\mathfrak{A}_{\mu_0} \models \exists \bar{x}.\chi$

and hence $\mathfrak{A} \models \exists\bar{x}.\chi$. To see that $\mathfrak{A} \models \forall\bar{x}_h(A_h(\bar{x}) \rightarrow \exists\bar{y}_h.\chi_h(\bar{x}_h\bar{y}_h))$, suppose $\mathfrak{A} \models A_h[\bar{a}]$. Then $\bar{a}$ belongs to some subset A_μ such that, writing $\bar{w} = f_\mu^{-1}[\bar{a}]$, $\sigma_\mu \models A_h[\bar{w}]$. If $\sigma_\mu \models \chi_h[\bar{w},\bar{v}]$ for some tuple $\bar{v}$, then, writing $\bar{b} = f_\mu(\bar{v})$, we have $\mathfrak{A}_\mu \models \chi_h[\bar{a},\bar{b}]$. Otherwise, by construction of V, μ has a daughter ν with $\mathfrak{A}_\nu\restriction\bar{w} = \mathfrak{A}_\mu\restriction\bar{w}$ and for which $\sigma_\nu \models \chi_h[\bar{w},\bar{v}]$, for some tuple $\bar{v}$, whence, writing $\bar{b} = f_\nu(\bar{v})$, we have $\mathfrak{A}_\nu \models \chi_h[\bar{a},\bar{b}]$. Thus, $\mathfrak{A} \models \forall\bar{x}_h(A_h(\bar{x}) \rightarrow \exists\bar{y}_h.\chi_h(\bar{x}_h,\bar{y}_h))$, as claimed. Finally, to see that $\mathfrak{A} \models \Upsilon$, let $\forall\bar{z}(C_h(\bar{z}_h) \rightarrow \omega_h(\bar{z}_h))$ be a conjunct of Υ. If $\mathfrak{A} \models C[\bar{a}]$, then $\bar{a}$ belongs to some subset A_μ, and by construction, $\mathfrak{A}_\mu \models \omega_h[\bar{a}]$, since $\mathfrak{A} \models \Upsilon$. □

Lemma 4.7. *The problem* Sat($\mathcal{G}$) *is in* 2-ExpTime. *For every k, the problem $\mathcal{G}^k$-Sat is in* ExpTime.

Proof. Let $\varphi \in GFk$ be given. By Lemmas 4.1–4.3, we may convert φ, in polynomial time, to an equisatisfiable formula $\varphi' \in \mathcal{G}^{k+1}$ in normal form. Now call `satGFk(`φ'`)`. The result follows from Lemmas 4.4, 4.5 and 4.6. □

We remark at this point that the model constructed in Lemma 4.6 is in general infinite. Thus, Lemma 4.7 concerns Sat($\mathcal{G}$) and not FinSat($\mathcal{G}$). In fact—and rather unusually for decidable fragments of first-order logic—establishing the finite model property for $\mathcal{G}$ is considerably harder than establishing the complexity of the satisfiability problem. We return to this topic in Sec. 4.4. But for now, we can content ourselves with the following easy special case.

Lemma 4.8. *The fragment $\mathcal{G}^2$ has the finite model property.*

Proof. Let φ be a satisfiable normal-form $\mathcal{G}^2$-formula. Let $\mathfrak{A}$ be the model of φ constructed in the proof of Lemma 4.6, based on the (possibly infinite) tree V. Without loss of generality, we may assume that the initial conjunct $\exists\bar{x}.\chi$ features just one variable, so that A_{μ_0} contains a single element (which is declared new). Furthermore, since every other vertex of V features at least one old and at least one new element, and we are working in $\mathcal{G}^2$, it follows that it features exactly one new element and (with the exception of μ_0) exactly one old element. So label each vertex μ of V with the 1-element structure $\mathfrak{A}_\mu\restriction v$, where v is the new element of A_μ, and label each edge (μ,ν) of V with the 2-element structure $\mathfrak{A}_\nu$. This labelling constitutes a complete specification of $\mathfrak{A}$, in the obvious sense. Since the elements of A are in 1–1 correspondence with the vertices of V (each element is mapped to the vertex where it is new), we may henceforth identify the sets A and V.

To make a finite model $\mathfrak{A}$ of φ, say that an element b of A is *blocked* if it has a proper ancestor a (in V) such that $\mathrm{tp}^{\mathfrak{A}}[a] = \mathrm{tp}^{\mathfrak{A}}[b]$. Let N be the number of 1-types realized in $\mathfrak{A}$, let $\mathfrak{A}_0$ be the result of deleting every blocked element and all of its descendants; and let $\mathfrak{A}_h$ $(1 \leq h \leq 3N-1)$ be a fresh copy of $\mathfrak{A}_0$. It is immediate that $\mathfrak{A}_0$ realizes all the 1-types realized in $\mathfrak{A}$. Let $\mathfrak{B}$ be the union of the structures $\mathfrak{A}_h$ $(0 \leq h \leq N-1)$, divided into disjoint substructures, $\mathfrak{B}_0$, $\mathfrak{B}_1$ and $\mathfrak{B}_3$, each containing N copies of $\mathfrak{A}_0$ (Fig. 4.2). Suppose $b \in B_i$ $(0 \leq i < 3)$ is a copy of $a \in A$. For each blocked daughter a' of a in $\mathfrak{A}$, select an element

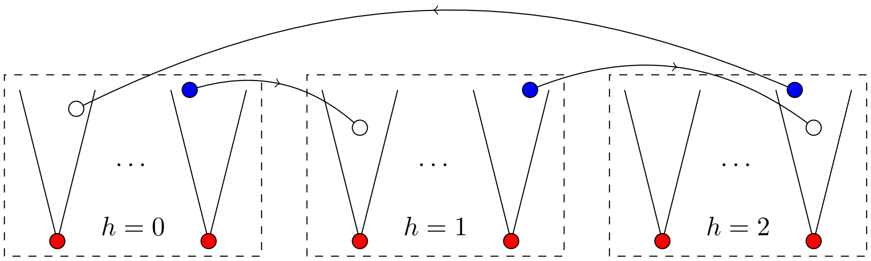


Figure 4.2: Proof of Lemma 4.8

b' in a fresh copy of $\mathfrak{A}_0$ in $\mathfrak{B}'_{i+1}$ (arithmetic in subscripts modulo 3), such that $\mathrm{tp}^{\mathfrak{B}'_{i+1}}[b'] = \mathrm{tp}^{\mathfrak{A}}[a']$ and redefine $\mathrm{tp}^{\mathfrak{B}}[b, b']$ to be $\mathrm{tp}^{\mathfrak{A}}[a, a']$. We see by inspection of Fig. 4.2 that no clashes arise. It is, moreover routine to check that $\mathfrak{B} \models \varphi$. Since $\mathfrak{B}$ is finite, this completes the proof. □

4.3 The guarded fragment: lower bounds

In this section, we obtain lower bounds for the fragments $\mathrm{Sat}(\mathcal{G}^k)$ ($k \geq 2$) and $\mathcal{G}$, matching the upper bounds of Lemma 4.7. For the problem $\mathrm{Sat}(\mathcal{G}^k)$, there is no more work to do. A quick glance at the forms defining the fragment given in Lemma 2.27 shows that they are all in $\mathcal{G}^2$, whence $\mathrm{Sat}(\mathcal{G}^2)$ is ExpTime-hard. It therefore remains only to establish a lower bound for $\mathrm{Sat}(\mathcal{G})$ to match that of Lemma 4.7.

Lemma 4.9. *The problem $\mathcal{G}$-Sat is* 2-ExpTime-*hard.*

Proof. Let M be an alternating Turing machine, with alphabet A and set of states S, and running in space bound 2^f, for some polynomial f. As usual, we take a *symbol* of M to be any element of A together with $\sqcup$ (blank) and $\triangleleft$ (start-of-tape). Recall that the run of M (on a particular input) is a tree of machine configurations, with the initial configuration as the root. Our strategy is to encode the run of M on some input string $\mathrm{x} \in A^*$ by means of a set of guarded formulas, $\Phi_{M,\mathrm{x}}$. We shall ensure that: (i) if $\Phi_{M,\mathrm{x}}$ has a model, then M accepts x; (ii) if M accepts x, then $\Phi_{M,\mathrm{x}}$ has a model. We ensure that, for fixed M, the construction requires only space $O(\log |\mathrm{x}|)$. Thus there is a many-one log-space reduction from the language recognized by M to $\mathrm{Sat}(\mathcal{G})$. This proves that $\mathrm{Sat}(\mathcal{G})$ is AExpSpace-hard; the theorem follows by Proposition 1.11. We may assume without loss of generality that that the states encountered in runs of M alternate strictly between universal and existential, starting with a universal state, and that, for any state s and any symbol a, M has at most two transitions available. Let these transitions be arbitrarily assigned to the sets T^- (left) and T^+ (right), so that for any state s and symbol a, T^- contains at most one transition $\langle s, a, \dots \rangle$, and similarly for T^+. Recall that a universal configuration is accepting just in case all its successors are, and an existential configuration is accepting just in case some successor is. Thus, a terminating configuration (with no enabled transitions) is accepting just in case it is universal.

The idea of the encoding is that accepting configurations will be represented by elements in models of $\Phi_{M,\mathrm{x}}$. Write $n = |\mathrm{x}|$ and $m = f(n)$, assuming, without loss of generality, that $m \geq n$. Let every symbol $a \in A \cup \{\sqcup, \triangleleft\}$ be an $(m+1)$-ary predicate, and every state $s \in S$, a unary predicate. Let h be an $(m+1)$-ary predicate, and c, t^+, t^-, unary predicates and r a binary predicate. As a guide to intuition, we give the intended interpretation of these predicates as follows:

$c(x)$	x is a configuration
$a(x, \bar{u})$	in configuration x, square $\bar{u}$ contains symbol a
$r(x, y)$	configuration y is a successor of configuration x
$h(x, \bar{u})$	in configuration x, the head is over square $\bar{u}$
$s(x)$	in configuration x, the current state is s
$t^{\pm}(x)$	in configuration x, enabled transitions in $T^{\pm}$ will be executed.

The problem is to encode runs of M using *guarded* formulas. Let B_1 an $(m+1)$-ary predicate $(m \geq 1)$ and 0, 1 individual constants. For ease of reading, we use the symbols 0 and 1 ambiguously for the individual constants themselves and their interpretations in some structure under consideration. We think of an m-tuple from $\{0, 1\}$ as a bit-string $\bar{d} = d_{m-1}, \ldots, d_0$ representing an integer on the range $[0, 2^m - 1]$ in the standard way (d_0 least significant), and for any i in this range, we denote the m-bit string representing it by "i". (Thus, for example, "0" is a string of m 0s.) We now add to $\Phi_{M,\mathrm{x}}$ formulas ensuring that every configuration (i.e. element a satisfying c) stands in the relation B_1 to *any* m-tuple chosen from the elements $\{0, 1\}$. That is, we would *like* to ensure

$$\forall x(c(x) \to B_1(x, \bar{d})). \tag{4.7}$$

for all bit-strings $\bar{d}$ of length m. Unfortunately, we cannot add all these conjuncts to $\Phi_{M,\mathrm{x}}$, because we cannot generate them in space $O(\log(|\mathrm{x}|))$. However, we can achieve the same effect *succinctly* by instead adding to $\Phi_{M,\mathrm{x}}$ the formula

$$\forall x(c(x) \to B_1(x, \bar{0})) \wedge \bigwedge_{i=0}^{m-1} \forall x(c(x) \to \forall \bar{u}(B_1(x, \bar{u}) \to B_1(x, \bar{u}\theta_i))), \tag{4.8}$$

where $\bar{u} = u_{m-1}, \ldots u_0$, and, for all i $(0 \leq i < m)$, θ_i is the substitution defined by $u_i\theta_i = 1$ and $u_j\theta_i = u_j$ for all $j \neq i$. It is easy to see that (4.8) entails (4.7), and can be computed within the required space bound. Similarly, let B_2 be a $(2m+2)$-ary predicate. Using the same technique, we also add to $\Phi_{M,\mathrm{x}}$ formulas *succinctly* ensuring that

$$\forall x \forall y(r(x, y) \to B_1(x, y, \bar{u}, \bar{v}))$$

for all bit-strings $\bar{u}$, $\bar{v}$ of length m; i.e. any two configurations satisfying the binary predicate r also stand in the relation B_2 to any *pair* of m-tuples chosen from the elements $\{0, 1\}$.

Recall again the procedure for incrementing a number stored as a bit string. Let k be an integer ($0 \leq k < 2^n - 1$) with standard n-bit representation $d_{n-1}, \ldots, d_0$ (d_0 is least significant). Then $k' = k + 1$ has standard n-bit representation $d'_{n-1}, \ldots, d'_0$, where, for all i $0 \leq i < n$:

$$d'_i = d_i \text{ if and only if } d_i = 0 \text{ for some } j \ (0 \leq j < i).$$

Let d^+ be a $2m$-ary predicate. We add to $\Phi_{M,\mathrm{x}}$ a formula stating that, if $\bar{u}$, $\bar{v}$ range over bit-strings of length m and satisfy $B_2(x, y, \bar{u}, \bar{v})$ with respect to some configurations x and y, then the relation $d^+(\bar{u}, \bar{v})$ holds just in case the number denoted by $\bar{v}$ is the successor of the number denoted by $\bar{u}$:

$$\forall x y \bar{u} \bar{v} \left(B_2(x, y, \bar{u}, \bar{v}) \rightarrow \left(d^+(\bar{u}, \bar{v}) \leftrightarrow \bigwedge_{i=0}^{m-1} \left((u_i = v_i) \leftrightarrow \left(\bigvee_{j=0}^{i-1} u_j = 0 \right) \right) \right) \right).$$

We also add similar formulas allowing us to read $d^-(\bar{u}, \bar{v})$ as "$\bar{v}$ is the predecessor of $\bar{u}$" and $d^0(\bar{u}, \bar{v})$ as "$\bar{v}$ is equal to $\bar{u}$".

With these preliminaries behind us, we add to $\Phi_{M,\mathrm{x}}$ a formula stating that in every configuration x, M is in exactly one state $s \in S$; if s is *existential*, x satisfies *either* t^+ or t^- (either a positive or a negative transition is executed), and if s is *universal*, x satisfies *both* t^+ and t^-. This is straightforward. We also add to $\Phi_{M,\mathrm{x}}$ the following formula stating that every square contains at most one symbol:

$$\forall x \forall \bar{u} \left(B_1(x, \bar{u}) \rightarrow \bigvee_{a, b \in A \cup \{\sqcup, \triangleleft\}}^{a \neq b} \neg(a(x, \bar{u}) \wedge b(x, \bar{u})) \right).$$

Notice that the guarding by B_1 does not compromise the effectiveness of the formula, since $\Phi_{M,\mathrm{x}}$ ensures that every configuration is related by B_1 to every bit-string of length m. We add to $\Phi_{M,\mathrm{x}}$ similar formulas ensuring that, in every configuration, the head is over at most one tape-square. We also add the following formulas setting up the initial configuration:

$$\exists x \left(c(x) \wedge s_0(x) \wedge h(x, \bar{0}) \wedge \triangleleft(x, \bar{0}) \wedge \bigwedge_{i=1}^{n} \mathrm{x}^i(x, \text{“}i\text{”}) \right)$$

$$\forall x \forall \bar{u} \left(B_1(x, \bar{u}) \rightarrow \left(s_0(x) \wedge \bigwedge_{i=0}^{n} (\neg d^0(\bar{u}, \text{“}i\text{”}) \rightarrow \sqcup(x, \bar{u})) \right) \right).$$

Here, we take it that $\mathrm{x} = \mathrm{x}^1 \cdots \mathrm{x}^n$, and that s_0 is the initial state of M. We are assuming without loss of generality that M never re-visits s_0.

We must add to $\Phi_{M,\mathrm{x}}$ formulas ensuring that transitions are executed when appropriate. Suppose, for example that $\tau = \langle a, s, b, t, +1 \rangle$ is a transition in T^+. ("If M is in state s reading a, then write b, move the head right and transition to state t.") Remembering that T^+ contains at most one transition enabled by state s and symbol a, we add to $\Phi_{M,\mathrm{x}}$ the formula

$$\begin{aligned}\forall x \forall \bar{u}(B_1(x, \bar{u}) \to (h(x, \bar{u}) \wedge a(x, \bar{u}) \wedge s(x) \wedge t^+(x) \to \\ \exists y(B_2(x, y, \bar{u}, \bar{u}) \wedge r(x, y) \wedge b(y, \bar{u}) \wedge \\ \forall \bar{v}(B_2(x, y, \bar{u}, \bar{v}) \to (d^+(\bar{u}, \bar{v}) \to h(y, \bar{v})))))),\end{aligned}$$

stating that if a is written on tape square $\bar{u}$ in configuration x, with the head visiting that square, and the configuration is in a state s and will execute any enabled transition in T^+, then there is a successor configuration y in which b has been written on tape square $\bar{u}$, and the head moved right. Notice the use of the B_2-atoms to ensure guardedness. This is harmless, because $\Phi_{M,\mathrm{x}}$ ensures that every pair of configurations satisfying the binary predicate r is related by B_2 to every pair of bit-strings of length m. Other transitions are treated similarly. We must also add to $\Phi_{M,\mathrm{x}}$ 'inertial' formulas stating that the tape does not change away from the head; but this can be done in a similar fashion.

A routine argument shows, in the same way as for Lemma 2.27, that, if $\Phi_{M,\mathrm{x}}$ has a model, M accepts x, and, conversely, that that, from an an accepting run on of M on input x, one obtains a (finite) model of $\Phi_{M,\mathrm{x}}$ by interpreting the predicates involved as indicated above. □

Combining lemmas 4.7, 4.9, as well as 2.27, we obtain:

Theorem 4.10. *The problem $\mathcal{G}$-Sat is* 2-ExpTime-*complete. For every $k \geq 2$, the problem $\mathcal{G}^k$-Sat is in* ExpTime-*complete.*

4.4 The finite model property

Sec. 4.2 says nothing about the finite satisfiability problem for $\mathcal{G}^k$ ($k \geq 3$). Indeed, we pointed out in connection with our discussion of the procedure `satGFk` that the structures yielded by the proof of Lemma 4.6 are not, in general, finite. Only in the case $k = 2$, where we could draw the models in question as trees, did we show how to replace these by finite models.

Such a simple approach is not available if $k > 2$. Consider, for example the $\mathcal{G}$-formulas:

$$\exists x \exists z.s(x, z) \wedge \forall \mathrm{x} \forall z(s(x, z) \to \exists w.r(x, z, w)) \wedge \forall x \forall y \forall x(r(x, y, z) \to s(x, z)).$$

The the proof of Lemma 4.6 constructs a tree embedding a sequence of types $\mu_1(v_0, v_1), \mu_1(v_0, v_1, v_2), \mu_2(v_0, v_2, v_3), \ldots$, where, for $i > 1$, μ_i contains the atoms $r(v_0, v_i, v_{i+1})$, $s(v_0, v_i)$ and $s(v_0, v_{i+1})$. In particular, the collections of elements of $\mathfrak{A}$ corresponding to vertex μ_i all contain v_0. It is unclear how the 'looping back' used in the proof of Lemma 4.8 is guaranteed to be meaningful in such a case.

Nevertheless, the finite model property does hold for $\mathcal{G}$. The proof relies on a sophisticated combinatorial theorem which we do not prove here. We first remind ourselves of some standard terminology. Recall from Sec. 3.7 that if $\mathfrak{A}$ and $\mathfrak{B}$ are structures, a *partial isomorphism* from $\mathfrak{A}$ to $\mathfrak{B}$ is a function $f : D \to E$, where $D = \mathrm{dom}(f) \subseteq A$, $E = \mathrm{rng}(f) \subseteq B$ and $f : \mathfrak{A}\restriction D \simeq \mathfrak{B}\restriction E$ is a structure isomorphism; we call f *finite* if D (hence E) is. Furthermore, an *automorphism* of $\mathfrak{A}$ is a structure isomorphism $f : \mathfrak{A} \to \mathfrak{A}$.

Proposition 4.11 (Herwig)**.** *Let $\mathfrak{A}$ be a finite model. There exists a finite model $\mathfrak{A}^+ \supseteq \mathfrak{A}$ such that, for any partial isomorphism p from $\mathfrak{A}$ to itself, there exists an automorphism f of $\mathfrak{A}^+$ such that f and p agree on dom(p). Furthermore, if q is a predicate interpreted by these structures and $\bar{a}$ a tuple from A^+ such that $\mathfrak{A}^+[\bar{a}]$, then there is an* automorphism *g of $\mathfrak{A}^+$ such that $g(\bar{a}) \subseteq A$.*

Theorem 4.12. *The guarded fragment has the finite model property.*

Proof. Let φ be a satisfiable sentence of $\mathcal{G}$. By Lemma 4.3, we may assume without loss of generality that φ is in normal form (4.3), repeated here for convenience as

$$\exists x.p_0(x) \wedge \bigwedge_{h=1}^{\ell} \forall \bar{x}_h(A_h \to \exists \bar{y}_h(B_h \wedge \beta_h)) \wedge \bigwedge_{h=1}^{m} \forall \bar{z}_h(C_h \to \gamma_h).$$

Suppose $\mathfrak{A} \models \varphi$. By Lemmas 4.5 and 4.6, we may assume without loss of generality that $\mathfrak{A}$ has the form yielded by the proof of Lemma 4.6, based on the tree of vertices V. Let $\mathfrak{A}_0$ be a finite initial segment V_0 of V such that, for all vertices ν in V, there exists a vertex μ labelled by the same type, and let $\mathfrak{A}_0$ be the restriction of $\mathfrak{A}$ to the elements occurring in the vertices of V_0. Finally, let $\mathfrak{A}_0^+$ be the extension of $\mathfrak{A}_0$ guaranteed by Proposition 4.11.

We claim that $\mathfrak{A}_0^+ \models \varphi$. That $\mathfrak{A}_0^+ \models \exists x.p_0(x)$ is immediate. Now consider any conjunct $\forall \bar{x}_h(A_h \to \exists \bar{y}_h(B_h \wedge \beta_h))$, and suppose $\mathfrak{A}_0^+ \models A_h[\bar{a}]$. We must find a tuple $\bar{d}$ such that $\bar{b}\bar{d}$ satisfies $B_h \wedge \beta_h$ in $\mathfrak{A}_0^+$. Now, then there is an automorphism g of $\mathfrak{A}_0^+$ mapping $\bar{a}$ to some tuple $\bar{b}$ from A_0 (and hence from A). Thus, $\mathfrak{A} \models A_h[\bar{b}]$, and so $\bar{b}$ is chosen from the elements $\bar{u}$ for some vertex μ with type $\sigma(\bar{u})$. If σ provides a witness for $\exists \bar{y}_h(B_h \wedge \beta_h)$ (with $\bar{x}$ taking the values $\bar{b}$), then there is nothing to show. Otherwise, let ν be a vertex of V with type $\tau(\bar{v})$ which provides a proper h-witness for $\bar{b}$. Thus, $\bar{b}$ has an h-witness in τ. Now let ν' be a vertex of V_0 with type $\tau'(\bar{v}')$, such that there exists an isomorphism isomorphic to $p : \tau(\bar{v}) \to \tau'(\bar{v}')$. Such a ν' exists by the choice of V_0. Let $\bar{b}'$ be the image of $\bar{b}$ under p, and let f be an automorphism of $\mathfrak{A}_0^+$ extending p. Thus, the composition $f \circ g$ maps $\bar{a}$ to $\bar{b}'$. Since $\bar{b}'$ has an h-witness in τ', let $\bar{d}'$ be such that $\bar{b}'\bar{d}'$ satisfies $B_h \wedge \beta_h$ in $\mathfrak{A}_0^+$, and let $\bar{d} = (f \circ g)^{-1}(\bar{d}')$. But then $\bar{b}\bar{d}$ satisfies $B_h \wedge \beta_h$ in $\mathfrak{A}_0^+$ as required. Finally, consider any conjunct $\forall \bar{z}_h(C_h \to \gamma_h)$, and suppose $\mathfrak{A}_0^+ \models A_h[\bar{a}]$. We must shows that $\bar{a}$ satisfies γ_h in $\mathfrak{A}_0^+$. But again, there is an automorphism g of $\mathfrak{A}_0^+$ mapping $\bar{a}$ to some tuple $\bar{b}$ from A_0 (and hence from A). Thus, $\mathfrak{A} \models C_h[\bar{b}]$, whence $\mathfrak{A} \models \gamma_h[\bar{b}]$, whence $\mathfrak{A}_0 \models \gamma_h[\bar{b}]$. Applying the automorphism g^{-1}, $\mathfrak{A}^+ \models \gamma_h[\bar{a}]$, as required. □

4.5 Excursus: Resolution theorem proving

In this section, we give an outline of *resolution theorem proving*, an approach to first-order satisfiability checking with important practical as well as theoretical applications. The immediate goal is an alternative proof of Lemma 4.7 for the special case of the equality-free guarded fragment, which we obtain in Sec. 4.6. But we shall return repeatedly to resolution-based techniques in subsequent chapters. Since resolution theorem proving is not specific to guarded formulas, we shall deal in this section with arbitrary first-order formulas. Throughout our treatment, signatures will in general be assumed to contain both individual constants and function-symbols. The presentation is structured as follows: Sec. 4.5.1 explains the conversion of first-order formulas to so-called clause form; Sec.4.5.2 then presents resolution theorem-proving for variable-free clauses; and Sec.4.5.3 generalizes this presentation to clauses with variables.

A term, expression or formula containing no variables is said to be *ground*.

4.5.1 Clausal form

Recall from Sec. 3.1 that a formula is in *prenex form* if none of its Boolean connectives out-scopes any of its quantifiers. (The formula ψ yielded by the proof of this lemma will in general not be guarded, even if φ is.) And recall from Sec. 3.4 that any prenex-form formula may be Skolemized, so that only universal quantifiers remain. From Lemmas 3.2 and 3.10, if φ is a sentence of first-order logic, then we can compute, in time bounded by a polynomial function of $\|\varphi\|$ a quantifier-free formula χ (in general involving function-symbols), such that $\varphi \lhd (\forall \bar{x}.\chi)$, where $\bar{x} = \text{vars}(\chi)$.

We remind ourselves that a *literal* is an atom (= atomic formula) or the negation of an atom, and that a *clause* is a disjunction of literals. When discussing resolution, we shall generally use the letters C and D to range over atoms, L and M to range over literals, γ and δ to range over clauses and Γ and Δ over sets of clauses. If L is a literal, we write $\bar{L}$ to denote the opposite literal: i.e. $\bar{L} = \neg C$, for $L = C$, an atom, and $\bar{L} = C$ for $L = \neg C$. Since the order of literals in clauses is not important, we may regard clauses as finite *multisets* of literals. We allow the falsum $\bot$ to be a clause, the disjunction of the empty multiset of literals.

Lemma 4.13. *Let χ be a quantifier-free formula (possibly containing variables). We may compute, in time bounded by a polynomial function of $\|\chi\|$, a set of clauses Γ such that $\chi \lhd \bigwedge \Gamma$. Moreover, the literals occurring in Δ are the literals appearing in ψ together with some collection of literals whose arguments are all variables.*

Proof. Using standard Boolean equivalences, we may assume without loss of generality that the negation symbol in χ applies only to atoms. Let $\chi_0 = \chi$ and $\Gamma_0 = \emptyset$. Thus $\chi \equiv \chi_0 \wedge \bigwedge \Gamma$. Suppose χ contains some subformula $\theta = L \circ M$, with L and M are literals and $\circ$ a Boolean connective. Writing $\bar{x} = \text{vars}(\theta)$, let p be a fresh predicate with arity $|\bar{x}|$. If $\circ$ is $\vee$, let $\chi_1 = \chi_0[\theta/p(\bar{x})]$ and let $\Gamma_1 = \Gamma_0 \cup \{\neg p(\bar{x}) \vee L \vee M, \bar{L} \vee p(\bar{x}), \bar{M} \vee p(\bar{x})\}$; then it is easy to see that $(\chi_0 \wedge \bigwedge \Gamma_0) \lhd (\chi_1 \wedge \bigwedge \Gamma_1)$. The case where $\circ$ is any of $\wedge$, $\rightarrow$ or $\leftrightarrow$ is handled similarly. Now repeat the procedure until we obtain χ_m and Γ_m with $(\chi_0 \wedge \bigwedge \Gamma_0) \lhd \cdots \lhd (\chi_m \wedge \bigwedge \Gamma_m)$, where χ_m is a literal L. Let $\Gamma = \{L\} \cup \Gamma_m$. □

If γ is a clause, we write $\forall^*\gamma$ for the formula $\forall \bar{x}.\gamma$, where $\bar{x} = \text{vars}(\gamma)$, and if Γ is a set of clauses we write $\forall^*\Gamma = \{\forall^*\gamma \mid \gamma \in \Gamma\}$. We refer to $\forall^*\gamma$ as the *universal closure* of γ; similarly for $\forall^*\Gamma$. A set of clauses is *universally satisfiable* if its universal closure is satisfiable. Universal satisfiability obviously implies satisfiability, but not conversely. For example, the set of three clauses $\{p(a), \neg p(x) \vee p(f(x)), \neg p(f(a))\}$ is satisfiable, but not universally satisfiable. From Lemmas 3.2, 3.10 and 4.13, we see that, if φ is a sentence of first-order logic, then we can compute, in time bounded by a polynomial function of $\|\varphi\|$, a set of clauses Γ such that φ is satisfiable if and only if Γ is universally satisfiable. We speak in this case of *putting φ into clause form*.

An example may be helpful. Consider again the valid argument given in (2.8), which we may render in first-order logic as follows:

$$\begin{array}{l} \exists x(\text{artst}(x) \wedge \forall y(\text{bkpr}(y) \rightarrow \neg\text{hate}(x,y))) \\ \underline{\forall x(\text{bkpr}(x) \rightarrow \exists y(\text{artst}(y) \wedge \text{hate}(x,y)))} \\ \exists x(\text{artst}(x) \wedge \neg\text{bkpr}(x))). \end{array} \tag{4.9}$$

This argument is of course valid just in case the sentence φ formed by conjoining the two premises with *the negation of* the conclusion is unsatisfiable. Putting φ into clause form, we obtain Γ given by:

$$\begin{aligned} \{&\text{artst}(s),\ \neg\text{bkpr}(y) \vee \neg\text{hate}(s,y),\ \neg\text{bkpr}(x) \vee \text{artst}(f(x)), \\ &\qquad\qquad \neg\text{bkpr}(x) \vee \text{hate}(x,f(x)),\ \neg\text{artst}(x) \vee \text{bkpr}(x)\}, \end{aligned} \tag{4.10}$$

where s is a Skolem constant and f a Skolem function. Hence to show that the argument (4.9) is valid, it suffices to show that the set of clauses (4.10) is *not* universally satisfiable. Resolution theorem proving is a method for showing that a set of clauses is not universally satisfiable.

For ground clauses, satisfiability and universal satisfiability are obviously the same thing. The following very simple result, often known as *Herbrand's Theorem*, connects these two notions more generally. If Γ is a set of clauses over a signature Σ, we consider the set $\hat{\Gamma}$ of its *ground instances*, that is, the (usually infinite) set of clauses obtained by applying all possible substitutions of ground terms over Σ to any clause in Γ.

Lemma 4.14. *A set of clauses is universally satisfiable if and only if the set of its ground instances is satisfiable.*

Proof. The only-if direction is trivial. For the if-direction, suppose $\mathfrak{A} \models \hat{\Gamma}$. Let I be the ground atoms (over Σ) true in $\mathfrak{A}$. Note that I satisfies the following condition: for each clause $\gamma \in \hat{\Gamma}$, either γ contains a positive literal C such that $C \in I$, or γ contains a negative literal $\neg C$ such that $C \notin I$. We write $I \models \hat{\Gamma}$ to denote this condition.

Given the set I, we build a model $\mathfrak{H}$ as follows. The domain of quantification is simply the set H of ground terms over Σ. For any individual constant c, we set $c^{\mathfrak{H}} = c$; for any n-ary function symbol f ($n \geq 1$), we set $f^{\mathfrak{H}} = \{\langle t_1, \ldots, t_n, f(t_1, \ldots, t_n)\rangle \mid t_1, \ldots, t_n \in H\}$; and for any n-ary predicate p ($n \geq 1$), we set $p^{\mathfrak{H}} = \{\langle t_1, \ldots, t_n\rangle \mid p(t_1, \ldots, t_n) \in I\}$. It is simple to check that $I \models \Gamma$ implies $\mathfrak{H} \models \forall^*\Gamma$. □

The structure $\mathfrak{H}$ guaranteed by Lemma 4.14 is sometimes called an *Herbrand* model. The notation $I \models \Delta$, where I is a set of ground atoms and Δ a set of ground clauses, will be used below.

4.5.2 Ground resolution

Resolution theorem proving provides a method to test the universal satisfiability of a set of clauses. Strangely, almost all of the work we have to do here involves the variable-free case. For the remainder of this section (Sec. 4.5.2), all atoms, clauses and literals are assumed to be ground.

The method consists of two rules, *resolution* and *factoring*, given by the respective schemata:

$$\frac{\gamma \vee C \quad \delta \vee \neg C}{\gamma \vee \delta} \qquad\qquad \frac{\gamma \vee L \vee L}{\gamma \vee L} \tag{4.11}$$

where γ, δ are clauses, C is an atom, and L is a literal. These rules are clearly valid: the sentences above the line entail the sentence below it. A *derivation* of a clause γ from a set of clauses Γ is a tree created by successive applications of these rules, starting with premises in Γ (at the leaves), and terminating in a single conclusion γ (at the root). In particular, if there is a derivation of the empty clause $\perp$ from Γ, then Γ is unsatisfiable. As we shall show in this section, the converse holds: if Γ is unsatisfiable, there is a derivation, using resolution and factoring, of $\perp$ from Γ. That is: resolution and factoring form a (sound and) refutation-complete inference system for ground clauses. This result holds even under sever restrictions on when these inference rules can be applied, based on the concepts of *orderings* and *selection functions*.

Fix a signature Σ of predicates of any arities, individual constants and function-symbols of any positive arities; we assume that Σ contains at least one individual constant. We consider the set of atoms over Σ, and suppose that $\prec$ is any strict partial order on this set. (We shall encounter examples presently.) We silently extend $\prec$ to a partial orderings on literals as follows: if L, M are literals featuring the respective atoms C and D, we write $L \prec M$ just in case $C \prec D$ or $L = C$ and $M = \neg C$. That is: literals are compared according to their atoms, subject to the additional stipulation that $C \prec \neg C$. We call a literal C *maximal in its clause* (*under* $\prec$) if there is no literal D in that clause such that $C \prec D$.

By a *selection function*, we understand any function which assigns to any clause γ over Σ some subset of the *negative* literals occurring in γ; we call the literals in this set *selected*. There are no constraints at all on selection functions, except that all selected literals are negative; in particular, it is allowed to select no literals from a clause, or indeed every negative literal in a clause. Suppose, then, $\mathfrak{s}$ is a selection function. Fixing $\prec$ and $\mathfrak{s}$, say that a literal L occurring in a clause γ is *eligible* if either $L \in \mathfrak{s}(\gamma)$, or $\mathfrak{s}(\gamma) = \emptyset$ and L is maximal in its clause under $\prec$. The rules of $(\prec, \mathfrak{s})$-*resolution* and -*factoring* are exactly as given in (4.11), but subject to the restriction that the displayed literals C, $\neg C$ and L are eligible in their respective clauses. Where $\prec$ and $\mathfrak{s}$ are clear from context, we suppress reference to them and simply speak of *ordered resolution/factoring.* Put simply, we are only allowed to resolve/factor on selected literals (if there are any), and only only on maximal literals if there are no selected literals. Notice that the function $\mathfrak{s}_0$ defined by $\mathfrak{s}(\gamma) = \emptyset$ is a selection function, and that the empty relation $\prec_0 = \emptyset$ is a well-founded partial order. With respect to $\prec_0$ and $\mathfrak{s}_0$, then, every literal is eligible; hence the notion of ordered resolution and factoring with selection includes unrestricted resolution and factoring as a special case. Since we are dealing with restrictions of valid rules, ordered resolution and factoring forms a sound deductive system. We show that, under minimal assumptions on $\prec$, it is in fact refutation-complete.

Recall that a strict partial order $\prec$ on any set X is *well-founded* if, for any infinite sequence of distinct elements $a_0, a_1, a_2, \ldots$ from X, there exist $i < j$ such that $a_i \prec a_j$, and that a *well-order* is a well-founded, total order. It is a standard fact that very well-founded partial order can be extended to a well-order. (Exercise 1.) It should be obvious that, if $\prec$ is a well-founded order on ground atoms, then the extension of $\prec$ to ground literals is also well-founded (and thus extends to a well-order).

Lemma 4.15. *Let Γ be a set of ground clauses without equality, $\prec$ a well-founded, strict partial order on the set of ground atoms, and $\mathfrak{s}$ a selection function. Then Γ is satisfiable if and only if there is no derivation of the empty clause from Γ using $\prec$-ordered resolution and -factoring with selection function $\mathfrak{s}$.*

To reduce notational clutter, if γ and δ are clauses, we write $\gamma \prec \delta$ rather than $\gamma \prec_{\text{mult}} \delta$.

To prove this result, we first extend the ordering $\prec$ from literals to ground clauses. Recalling that clauses are essentially just multisets of literals, we may employ the following standard construction. If $\prec$ is a strict partial order on a set X, and S_1, S_2 are multisets over X, we write $S_1 \prec_{\text{mul}} S_2$ just in case $S_1 \neq S_2$ and, for all $a \in X$, there exists $b \in M$ such that $a \preceq b$ and $S_1(b) \leq S_2(b)$. We refer to $\prec_{\text{mul}}$ as the *multiset extension* of $\prec$. The following lemma is standard in the literature on term re-writing. (The proof is a nice exercise; see, e.g. [8, pp. 22–24].)

Lemma 4.16. *If $\prec$ is a partial order on X, then $\prec_{\text{mul}}$ is a partial order on the set of multisets over X. Moreover, if $\prec$ is total, then so is $\prec_{\text{mul}}$, and if $\prec$ is well-founded, then so is $\prec_{\text{mul}}$.*

Let us return to Lemma 4.15. The only-if direction is trivial; we need only establish the if-direction. Let Γ, $\prec$ and $\mathfrak{s}$ be as in the statement of the lemma.

Since every well-founded order can be extended to a well-order, and since extending $\prec$ further restricts the available derivations, it suffices to establish the result in the special case that $\prec$ is a total order. Thus, we assume that $\prec$ is a well-order on ground atoms, and hence on ground literals, and hence, by Lemma 4.16, on ground clauses. We attempt to construct a set of atoms I with the aim of securing the condition $I \models \Gamma$ (in the sense of the proof of Lemma 4.14). This construction may or may not succeed, depending on Γ: if it fails, we refer to any $\gamma \in \Gamma$ for which $I \not\models \gamma$ as a *counterexample*. Since Γ is well-ordered by $\prec$, there is in this case a smallest counterexample. The construction proceeds as follows. We define, for each ground clause δ (not necessarily in Γ) sets of atoms I_δ and e_δ, proceeding by simultaneous transfinite induction. Once this is done, we let $I_\Gamma = \bigcup\{I_\gamma \mid \gamma \in \Gamma\}$. For the induction, assuming e_γ and I_γ have been defined for all $\gamma \prec \delta$, set $I_\delta = \bigcup_{\gamma \prec \delta} e_\gamma$. In particular, if δ_0 is the smallest clause of all in Γ with respect to $\prec$, then $I_{\delta_0} = \emptyset$. To define e_δ, test whether the following four conditions apply: (i) $\delta \in \Gamma$, (ii) $I_\delta \not\models \delta$; (iii) the maximal literal of δ is positive, say C; and (iv) $\mathfrak{s}(\delta) = \emptyset$. If so, set $e_\delta = \{C\}$; if not, set $e_\delta = \emptyset$. We say that δ is *productive* if $e_\delta \neq \emptyset$.

Intuitively, the set I_δ tries to make true all clauses $\gamma \in \Gamma$ such that $\gamma \prec \delta$, so that $I = \bigcup_{\delta \in I} I_\delta$ has the desired property that $I \models \Gamma$. We remark in respect of the definition of e_δ: (i) we do not make true any clause not in Γ; (ii) we do not make true any clause which is already true, as it will remain so throughout the construction; (iii) we do not make true any clause by satisfying non-maximal literals; and (iv) we do not make true any clause which has any selected (negative) literals.

Since $\gamma \prec \delta$ implies $I_\gamma \subseteq I_\delta$, it is then immediate that $I_\Gamma \models \gamma$ for any productive clause γ. In particular, for $\gamma \in \Gamma$ such that the maximal selected literal of γ is positive, and γ contains no selected literals, we have $I_\Gamma \models \gamma$. The key technical observation is that, if $\gamma = C \vee \gamma'$, and with $e_\gamma = \{C\}$, no literals of γ' are true in I_Γ. For the negative literals this is obvious: they must be false in I_γ and thus will remain false. So suppose some positive literal $D \prec C$ true in I_Γ. Since D is false in I_γ, we have $e_\delta = \{D\}$ for some $\delta \succ \gamma$. And since D is maximal in δ, we have $L \preceq D$ for all $L \in \delta$. But then δ results from γ by deleting (zero or more) elements and replacing C by a multiset of strictly smaller literals, hence $\delta \prec \gamma$, a contradiction.

We require one more technical lemma.

Lemma 4.17. *Suppose Γ is a set of ground clauses closed under application of $(\prec, \mathfrak{s})$-resolution and $(\prec, \mathfrak{s})$-factoring, and suppose γ is a counterexample for Γ. If γ contains a selected literal $\neg B$, there is a counterexample ϵ such that $\epsilon \prec \gamma$.*

Proof. Let $\gamma = \gamma' \vee (\neg D \vee \cdots \vee \neg D)$. Since γ is a counterexample, $D \in I_\gamma$; let $\delta = \delta' \vee (D \vee \cdots \vee D) \in \Gamma$ be such that $e_\delta = \{D\}$. Thus, Γ contains the conclusion of the inference

$$\dfrac{\dfrac{\begin{array}{c}\gamma' \vee (\neg D \vee \cdots \vee \neg D)\\ \vdots \\ \gamma' \vee (\neg D \vee \neg D)\end{array}}{\gamma' \vee \neg D} \qquad \dfrac{\begin{array}{c}\delta' \vee (D \vee \cdots \vee D)\\ \vdots \\ \delta' \vee (D \vee D)\end{array}}{\delta' \vee D}}{\gamma' \vee \delta'} \;. \tag{4.12}$$

We claim that $\epsilon = \gamma' \vee \delta'$ has the required properties. By assumption, all literals in γ' are false in I_Γ; moreover, to have $e_\delta = \{B\}$, we require all literals in δ' (which are non-maximal in δ) to be false in I_δ; and, as just observed these must remain false in I_Γ. Hence, ϵ is a counterexample. Finally, we see that, for all $L \in \delta'$, $L \prec D \prec \neg D$. Hence, ϵ results from γ by substituting a multiset of strictly smaller elements for $\neg D$, whence $\epsilon \prec \gamma$. □

Now we have all the ingredients for a proof of the refutation-completeness of $(\prec, \mathfrak{s})$-resolution and -factoring in the ground case.

Proof of Lemma 4.15. Let Γ_∞ be the smallest set of (ground) clauses including Γ, and closed under $(\prec, \mathfrak{s})$-resolution and $(\prec, \mathfrak{s})$-factoring, and write $I = I_{\Gamma_\infty}$. We show that $\bot \notin \Gamma_\infty$ entails $I \models \Gamma_\infty$. Suppose, for contradiction that Γ_∞ contains counterexample; let γ be the smallest one (under $\prec$). By Lemma 4.17, γ does not contain a selected literal. And since $\gamma \neq \bot$, let D be the maximal *atom* occurring in γ. We observed earlier that, if γ contains no selected literal and the maximal literal of γ is positive, then $I \models \gamma$. Hence D occurs negatively in γ, and we can write $\gamma = \gamma' \vee (\neg D \vee \cdots \vee \neg D)$. Since $I \not\models \gamma$, there is a $\delta \in \Gamma_\infty$ such that $e_\delta = \{D\}$; write $\delta = \delta' \vee (D \vee \cdots \vee D)$. Applying the inference (4.12) again, have $\epsilon = \gamma' \vee \delta' \in \Gamma_\infty$. The argument is then much as for Lemma 4.17: all the literals of δ' are false in I and so remain false in I, whence ϵ is a counterexample; and ϵ is the result of replacing $\neg D$ in γ by a multiset of strictly smaller elements, whence $\epsilon \prec \gamma$. This contradicts the assumed minimality of γ. □

4.5.3 Resolution for arbitrary clauses

Generalizing resolution theorem-proving to arbitrary (non-ground) clauses is surprisingly easy, though we need to say a little more about orderings on atoms.

By an *expression*, we mean either a term or an atom. A *substitution* is a function θ assigning to each variable x a term $x\theta$; if e is an expression, $e\theta$ denotes the result of replacing (in parallel) every variable x in e by $x\theta$. For example, if $x\theta = f(y)$ and $y\theta = b$, then $p(x, g(y, y))\theta = p(f(y), g(b, b))$. If e and e' are expressions, a *unifier* of e and e' is a substitution θ such that $e\theta = e'\theta$. If such a θ exists, we say that e and e' are *unifiable*. It is a standard result that, if e and e' are unifiable, then there is a substitution θ such that $e\theta = e'\theta$, and such that any θ' satisfying $e\theta' = e'\theta'$ can be factored as $\theta' = \theta\sigma$, where σ is a substitution. We call θ the *most general unifier* (*mgu*) of e and e'. The mgu of two unifiable expressions is unique up to renaming of variables. Given e and e', we may decide in polynomial (in fact, linear) time whether e and e' are

$$\dfrac{\dfrac{\dfrac{\neg\mathrm{a}(x)\vee\mathrm{b}(x)\quad\mathrm{a}(s)}{\mathrm{b}(s)}\quad\neg\mathrm{b}(x)\vee\mathrm{h}(x,f(x))}{\mathrm{h}(s,f(s))}\quad\dfrac{\dfrac{\dfrac{\dfrac{\neg\mathrm{a}(x)\vee\mathrm{b}(x)\quad\mathrm{at}(s)}{\mathrm{b}(s)}\quad\neg\mathrm{b}(x)\vee\mathrm{a}(f(s))}{\mathrm{a}(f(s))}\quad\neg\mathrm{a}(x)\vee\mathrm{b}(x)}{\mathrm{b}(f(s))}\quad\neg\mathrm{b}(y)\vee\mathrm{h}(s,y)}{\neg\mathrm{h}(s,f(s))}}{\bot}\ .$$

Figure 4.3: Resolution derivation of empty clause from the clause set (4.10): predicate names have been shortened to save space.

unifiable, and compute (a compact representation of) θ. In general, however, the size of the unificant $e\theta = e'\theta$ may be exponentially large as a function of the sizes of e and e'.

Let $\prec$ be a strict partial order on atoms (not just ground atoms). We say that $\prec$ is *liftable* if $e \prec f \Rightarrow e\theta \prec f\theta$ for all expressions e, f and all substitutions θ; and we say that $\prec$ is *admissible* if it is liftable and well-founded. Trivially, the empty order is admissible; we shall give a more interesting example in the next section. We extend $\prec$ from atoms to literals and clauses as before. A *selection function* is, as before, any function which returns, for any clause, a subset of its negative literals, and a literal is eligible in its clause if it is either selected by $\mathfrak{s}$, or there are no selected literals and it is maximal under $\prec$. Fix some admissible ordering $\prec$ and selection function $\mathfrak{s}$. By ($\prec\ fs$)-resolution and -factoring, we understand the following inference rules:

$$\frac{\gamma\vee C \quad \delta\vee\neg D}{\gamma\vee\delta} \qquad\qquad \frac{\gamma\vee C\vee D}{(\gamma\vee C)\theta}\ , \tag{4.13}$$

where θ is the mgu of C and D, and the literals C and $\neg D$ are both eligible in their respective clauses. When applying resolution, we assume that the antecedents have no variables in common: this can always be achieved, of course, by variable re-naming.

These inference rules are again valid in the sense that the universal closure of the antecedent(s) entails the universal closure of the consequent. Thus, they yield a sound derivation system in the usual sense: in particular, if the empty clause $\bot$ can be derived from a set of clauses Γ by means of these rules, then Γ is not universally satisfiable. Thus, for example, let Γ be the clause set given in (4.10). Taking $\mathfrak{s}$ to be the empty selection function, and $\prec$ the empty partial order (i.e. all literals are eligible), we can construct the derivation of $\bot$ shown in Fig. 4.3. showing that Γ is, as claimed, not universally satisfiable, and hence that argument (4.9) is valid.

We now show that the derivation system based on ordered resolution and factoring with selection is refutation-complete. We rely on a lemma of a type commonly referred to as a *lifting lemma*. Let Γ be a set of clauses and $\hat{\Gamma}$ the set of ground instances of Γ. Let $\mathfrak{s}$ be a selection function. We define another selection function $\hat{\mathfrak{s}}$ as follows. If $\gamma \in \hat{\Gamma}$, pick some (abitrary) $\check{\gamma} \in \Gamma$ and ground substitution θ such that $\check{\gamma}\theta = \gamma$, and define $\hat{\mathfrak{s}}(\gamma)) = \mathfrak{s}(\check{\gamma})\theta$; otherwise, if $\gamma \notin \Gamma$, define $\hat{\mathfrak{s}}(\gamma)$ in any way at all.

Lemma 4.18. *Fix some admissible ordering $\prec$ and selection function $\mathfrak{s}$. Let Γ be a set of clauses, and $\hat{\Gamma}$ the set of ground instances of clauses in Γ. Construct the selection function $\hat{\mathfrak{s}}$ as above. If $\gamma, \delta \in \hat{\Gamma}$ resolve under $\prec$ and $\hat{\mathfrak{s}}$ to form a clause ϵ, then there exist $\check{\gamma}, \check{\delta} \in \Gamma$, a clause $\check{\epsilon}$ and ground substitutions θ, η, ρ with $\gamma = \check{\gamma}\theta$, $\delta = \check{\delta}\eta$ and $\epsilon = \check{\epsilon}\rho$, such that $\check{\gamma}$ and $\check{\delta}$ resolve under $\prec$ and $\hat{\mathfrak{s}}$ to form $\check{\epsilon}$. Similarly, if $\gamma \in \hat{\Gamma}$ factorizes under $\prec$ and $\hat{\mathfrak{s}}$ to form a clause ϵ, then there exists $\check{\gamma} \in \Gamma$, a clause $\check{\epsilon}$ and ground substitutions θ, ρ with $\gamma = \check{\gamma}\theta$ and $\epsilon = \check{\epsilon}\rho$ such that $\check{\gamma}$ factorizes under $\prec$ and $\hat{\mathfrak{s}}$ to form $\check{\epsilon}$.*

Proof. We consider only the first statement (resolution); the second (factorization) is dealt with similarly. By the definition of $\hat{\mathfrak{s}}$, there exist clauses $\check{\gamma}, \check{\delta} \in \Gamma$ and ground substitutions θ, η such that $\check{\gamma}\theta = \gamma$ and $\check{\delta}\eta = \delta$, with the selected literals of $\check{\gamma}$ (under $\mathfrak{s}$) being precisely the instances of the the selected literals of γ (under $\hat{\mathfrak{s}}$); and similarly for δ. Moreover, if L is a maximal literal of γ, then the literal $\check{L}$ of $\check{\gamma}$ which gives rise to it is maximal in $\check{\gamma}$ by the liftability of $\prec$; and similarly for δ. Indeed, since γ and δ are assumed to share no variables, we may as well suppose $\eta = \theta$.

Without loss of generality, write $\check{\gamma} = A \vee \gamma$ and $\check{\delta} = \neg B \vee \delta'$, where $A\theta = B\theta$ is the resolved-on atom in the ground resolution step. Since A and B unify, let $\zeta = \mathrm{mgu}(A, B)$, and let ρ be such that $\theta = \sigma\rho$. Thus, we have the inference

$$\frac{\gamma \qquad \delta}{(\gamma' \vee \delta')\zeta} \; .$$

Thus, setting $\check{\epsilon} = (\gamma' \vee \delta')\zeta$, we have $\check{\epsilon}\rho = \epsilon$ as required. □

Finally, we have reached the goal of this section (Sec. 4.5.3).

Theorem 4.19. *Let Γ be a set of clauses without equality. Then Γ is universally satisfiable if and only if there is no derivation of the empty clause from Γ using an admissible ordering $\prec$ and a selection function $\mathfrak{s}$.*

Proof. The only-if direction is trivial. For the if-direction, let Γ_∞ be the smallest set of clauses including Γ and closed under $(\prec, \mathfrak{s})$-ordered resolution and factoring; and let $\hat{\Gamma}_\infty$ be the set of ground instances of Γ_∞. Let $\hat{\mathfrak{s}}$ be the selection function defined above (with respect to Γ_∞). By Lemma 4.14, if Γ_∞ is unsatisfiable, so is $\hat{\Gamma}_\infty$. By Lemma 4.15, therefore, there is a proof of $\bot$ from $\hat{\Gamma}_\infty$ under the ordering $\preceq$ and the selection function $\hat{\mathfrak{s}}$. Viewing this derivation as a tree (in which vertices are labelled with clauses and edges connect antecedents of inference steps to their consequents), we argue that, for every vertex in this tree, with label γ:

there is a clause $\gamma' \in \Gamma_\infty$ and a substitution ρ such that $\gamma'\rho =$ γ, (I) and hence (as an immediate consequence, given the construction of $\hat{\mathfrak{s}}$),

> there is a clause $\check{\gamma} \in \Gamma_\infty$ and a substitution θ such that $\check{\gamma}\theta = \gamma$ *and* $\mathfrak{s}(\check{\gamma})\theta = \hat{\mathfrak{s}}(\gamma)$. (II)

We prove these statements by induction, working from the leaves to the root. If γ is at a leaf, statement (II) holds by definition. Now suppose ϵ is the conclusion of a resolution step with premises γ and δ. By inductive hypothesis, statement (II) holds for γ and δ. By Lemma 4.18, statement (I) holds for ϵ; therefore statement (II) does as well. This completes the induction. It follows that $\perp$ is in Γ, as required. □

It is worth remarking on the rather sly character of this last induction. Lemma 4.18 tells us that the clause ϵ at a particular point in the ground proof is the ground instance of some clause ε' in Γ_∞. But this fact itself does not allow the induction to proceed, because the literals selected by $\hat{\mathfrak{s}}$ need not in general be instances of the literals of ϵ' selected by $\mathfrak{s}$. No matter: it guarantees that ε is an instance of *some* clause $\check{\epsilon} \in \Gamma_\infty$ (with a possibly quite different derivation to that of ϵ'), such that the literals in ϵ selected by $\hat{\mathfrak{s}}$ *are* the corresponding instances of the literals of $\check{\epsilon}$ selected by $\mathfrak{s}$. Thus, the 'lifted' derivation of $\perp$ from Γ yielded by the induction need not in general look anything like the ground derivation from which its existence is inferred.

4.6 Deciding the guarded fragment by resolution

Theorem 4.19, together with its escort of Lemmas 3.2, 3.10 and 4.13, give us a method for checking the satisfiability of any sentence φ of first-order logic without equality: convert φ to clause form Γ, and use resolution and factoring to try to derive $\perp$. However, this is not an algorithmic solution to the *Entscheidungsproblem*, since the saturation of a given finite clause set Γ under resolution and factoring is in general infinite. When searching for a derivation of $\perp$, one never knows whether victory is just around the corner.

On the other hand, a judicious choice of admissible ordering can sometimes keep clause sets finite. As an example, consider the following atom ordering, $\prec_d$, which will prove useful below. If s and t are terms, define the *depth* of s in t, denoted $d(s,t)$, as follows:

$$d(s,t) = \begin{cases} 0 & \text{if } s = t \\ \max_i d(s,t_i) + 1 & \text{if } s \neq t \text{ and } t = f(t_1, \ldots t_m) \\ -\infty & \text{otherwise;} \end{cases}$$

and define the *depth* of t, denoted $d(t)$, to be the maximum depth of any term occurring in t. For example, if $t = f(x, g(y, y, h(a)))$, then $d(x,t) = 1$, $d(y,t) = 2$, and $d(t) = d(a,t) = 3$. Note that $d(s,t) = -\infty$ if s does not occur in t. If

$A = p(t_1, \ldots, t_m)$ is a an atom and u a variable, define $d(u, A) = \max_i d(u, t_i)$ and $d(A) = \max_i d(t_i)$. Now define

$$A \prec_d B \text{ if } A \neq B,\ d(A) \leq d(B) \text{ and } d(x, A) \leq d(x, B) \text{ for all } x \in \text{vars}(A).$$

It is immediate that $\prec_d$ is an admissible ordering on atoms. Observe that $A \prec_d B$ implies $\text{vars}(A) \subseteq \text{vars}(B)$.

Now consider the clause set $\Gamma = \{p(a), \neg p(x) \vee p(f(x))\}$, and using the ordering $\prec_d$ and the empty empty selection function $\mathfrak{s}_0$, we see that no resolution steps are possible, since the only eligible literal in the clause $\neg p(x) \vee p(f(x))$ is $p(f(x))$. As $\perp$ is therefore not derivable, Theorem 4.19 gives us an assurance that Γ is universally satisfiable.

An atom A is *simple* if no function-symbol appears in the scope of any other; a clause is *simple* if each of its atoms is simple. Note that, if A is simple, then $d(A) \leq 1$, and $d(x, A) \leq 1$ for all variables x. An atom A is *covering* if every functional term in A contains all the variables occurring in A; a clause γ is *covering* if every functional term in contains all the variables in γ. For example, the atom $p(f(x, y), x)$ is covering, but the atoms $p(f(x, a), y)$ and $p(u, g(b))$ are not.

Lemma 4.20. *Suppose A and B are covering literals with* mgu θ*. If A and B are simple, then so is $A\theta = B\theta$.*

Proof. Write $A = p(s_1, \ldots, s_n)$, $B = p(t_1, \ldots, t_n)$, and suppose $C = A\theta = B\theta$ is not simple. Then there exists i $(1 \leq i \leq n)$ such that s_i is a variable and t_i is a functional term (or vice versa). But then A must also contain some functional term, say s_j, with $j \neq i$. Since s_i is a proper sub-term of s_j, $t_i\theta = s_i\theta$ is a proper sub-term of $s_j\theta = t_j\theta$. But t_j is a either a variable which occurs in the functional term t_i, or a functional term having the same variables as t_i. Either way, it is impossible for $t_i\theta$ to be a proper sub-term of $t_j\theta$. □

Lemma 4.21. *Let $A_1 \vee \gamma_1$ and $\neg A_2 \vee \gamma_2$ be simple, covering clauses which resolve to form $\delta = (\gamma_1 \vee \gamma_2)\theta$, where $\theta = \text{mgu}(A_1, A_2)$. Suppose that, for $i = 1, 2$:* (i) $\text{vars}(A_i) \supseteq \text{vars}(\gamma_i)$*, and* (ii) *if γ_i is functional, then so is A_i. Then the clause δ is simple and covering. Similarly, let $A_1 \vee A_2 \vee \gamma$ be a simple, covering clause which factors to form $\delta = (A_1 \vee \gamma)\theta$. Then, under the same suppositions concerning A_1, A_2 and γ, the clause δ is simple and covering.*

Proof. We prove the first statement only; the second follows by similar reasoning. Suppose, for contradiction, that δ contains a term t' with embedded function symbols. Then θ either assigns t' to a variable u occurring in γ_1 or γ_2 (and hence occurring in A_1 or A_2), or assigns a functional term to a variable occurring in a functional context in γ_1 or γ_2 (and hence occurring in a functional context in A_1 or A_2). Either way, $A_1\theta = A_2\theta$ is not simple, contradicting Lemma 4.20.

Suppose now that t' is a functional term occurring in δ. Then either θ assigns t' to some variable occurring in γ_1 or γ_2 (and hence occurring in A_1 or A_2), or $t' = t\theta$ for some functional term occurring in γ_1 or γ_2. In the former case, $t' = s\theta$ for some functional term s occurring in either A_1 or A_2; in the latter case, there must be a functional term s in either A_1 or A_2 with $\text{vars}(s) = \text{vars}(t)$. Either way, we have $\text{vars}(t') = \text{vars}(s)\theta = \text{vars}(A_1)\theta = \text{vars}(A_2)\theta \supseteq \text{vars}(\delta)$, so that δ is covering. □

In the above proof, we stated that if $\theta = \text{mgu}(A_1, A_2)$, then any functional term which θ assigns to a variable must arise as $s\theta$, for some functional term in either A_1 or A_2. Intuitively, this should be obvious by imagining the expressions A_1 and A_2 drawn as trees. However, a careful (if rather wearisome) proof can be given through an analysis of the unification algorithm. The reader is referred to [22, p. 100] for details.

One more definition. A clause γ is *properly guarded* if: (i) γ is simple and covering; and (ii) γ is either ground or contains a non-functional, negative literal, called a *guard*, which contains all the variables of γ. Now consider a formula in guarded normal-form, as given by (4.3) and let φ' be the corresponding formula

$$\chi(\bar{c}) \wedge \bigwedge_{h=1}^{\ell} \forall \bar{x}_h (A_h(\bar{x}_h) \to \chi(\bar{x}_h, f(\bar{x}_h))) \wedge \bigwedge_{h=1}^{m} \forall \bar{z}_h (C_h(\bar{z}_h) \to \omega_h(\bar{z}_h)),$$

where the $\bar{c}$ are individual constants and the f_h are function symbols. It is obvious that $\varphi \lhd \varphi'$. This transformation is in essence just the process of Skolemization mentioned in Lemma 3.10, except that we have eliminated the existential quantifiers 'intelligently', rather than first converting to prenex form. Now remove the universal quantifiers from φ' and transform into clause-form, Γ, as in Lemma 4.13. Thus, φ is satisfiable if and only if Γ is universally satisfiable.

Lemma 4.22. *If Γ is the clause form of a formula in guarded normal-form, obtained in the way just outlined, then every clause in Γ is properly guarded.*

Proof. Immediate. □

Since properly guarded clauses are by definition simple, there is a bounded number of them. Indeed,

Lemma 4.23. *The number of simple clauses (featuring no repeated literals) in k variables $x_1, \ldots, x_k$ over a signature Σ with maximum arity k is $2^{2(k+|\Sigma|)^{k^2+k+1}}$*

Proof. There are at most $(|\Sigma| + k)^{k+1}$ terms of depth at most one and hence at most $(|\Sigma| + k) \cdot ((|\Sigma| + k)^{k+1})^k = 2(k + |\Sigma|)^{k^2+k+1}$ simple atoms, and hence at most twice that number of literals based on simple atoms. □

Here now is our strategy for deciding the satisfiability of a given sentence φ (without equality) in the guarded fragment. Assume without loss of generality that φ is in guarded normal form, and convert to a set of guarded clauses Γ over a signature Σ, as guaranteed by Lemma 4.22. Suppose we could show that, for a particular atom ordering $\prec$ and selection function $\mathfrak{s}$, $(\prec, \mathfrak{s})$-resolution and

-factoring never lead outside the set of guarded clauses. Then we could apply these inference rules to Γ to the point of saturation: since the number of such clauses is subject to the bound of Lemma 4.23, we must eventually reach a point where no more clauses derivable. If, by this point, $\perp$ has not been derived, φ is satisfiable; otherwise, it is not. All we require, therefore, is an appropriate choice of $\prec$ and $\mathfrak{s}$.

We have already encountered a suitable ordering $\prec$: we may simply select the depth-ordering $\prec_d$ defined above. Turning now to the selection function, let $\mathfrak{s}$ be *any* (partial) function which maps a clause γ to some negative literal it contains with the following properties: (i) if γ contains any functional negative literals, one of these is selected; (ii) if γ contains functional literals, but they are all positive, no literal is selected; and (iii) otherwise (i.e. γ is non-functional) if γ contains a guard, then a guard is selected. Let us fix these choices of $\prec$ and $\mathfrak{s}$. Observe that, if a literal L is eligible in a clause $L \vee \gamma$, then: (i) L contains all the variables of γ; and (ii) if γ is functional, then so is L.

The key to the decision procedure for $\mathcal{G}^k$ is the following lemma.

Lemma 4.24. *In any application of equality and factoring, with the selection function $\mathfrak{s}$ and literal ordering $\prec$ given above, if the premises are properly guarded clauses, so is the conclusion.*

Proof. We give the proof for resolution only; factoring (easier) is left as an exercise. Consider the inference

$$\frac{A_1 \vee \gamma_1 \qquad\qquad \neg A_2 \vee \gamma_2}{(\gamma_1 \vee \gamma_2)\theta}\ ,$$

where $\theta = \mathrm{mgu}(A_1, A_2)$. Since A_1 and $\neg A_2$ are eligible in their clauses, we have, for $i = 1, 2$: (i) $\mathrm{vars}(A_i) \subseteq \mathrm{vars}(\gamma_i)$, and (ii) if γ_i is functional, then so is A_i. By Lemma 4.21, then, δ is simple and covering.

It remains to show that γ is either ground or contains a guard. Now, if either of $A_1 \vee \gamma_1$ or $\neg A_2 \vee \gamma_2$ is ground, then, since eligible literals contain all the variables of their clause, it follows that δ is ground and there is nothing more to show. Thus, we may suppose that $A_1 \vee \gamma_1$ contains a guard, whence A_1 is functional (since otherwise a guard would be selected and A_1 would not be eligible). Hence θ assigns no functional terms to the variables of $A_1 \vee \gamma_1$, whence, letting $\neg G$ be a guard of $A_1 \vee \gamma_1$, it follows that $\neg G\theta$ is a guard for δ. □

We have proved:

Theorem 4.25. *With the ordering $\prec$ and selection function $\mathfrak{s}$ defined above, resolution theorem-proving and factoring constitute a decision procedure for $\mathcal{G}$ without equality. This procedure runs in time bounded by a singly exponential function of the size of any input in $\mathcal{G}^k$ (for fixed k) and by a doubly exponential function of the size of the input if no bound is placed on k.*

Theorem 4.25 extends to the full fragment $\mathcal{G}$ (i.e. with equality) using the additional rules of *paramodulation*, *equality factoring* and *reflexivity resolution*

The analogues of Lemma 4.24, which state that these rules (under the ordering $\prec$ and selection function $\mathfrak{s}$ used here) preserve membership in the set of guarded clauses are routine, and employ essentially the same style of reasoning. However, the extension of Theorem 4.19 to cover these rules requires more work. As we do not require paramodulation elsewhere in this book, we refer the reader to the Bibliographic Notes.

4.7 Expressive power

In Sec. 3.7, we showed that a first-order sentence is logically equivalent to a sentence of $\mathcal{FO}^k$ if and only if it is preserved under k-bisimulations. In this section, we obtain an analogous semantic characterization of $\mathcal{G}$. Again, readers interested only in (finite) satisfiability can skip this material without loss; the remainder are invited to familiarize themselves with the argument of Sec. 3.7.

We start by adapting the notion of k-bisimulation to the guarded context. Let $\mathfrak{A}$ and $\mathfrak{B}$ be structures interpreting a common relational signature Σ. Call a subset $D \subseteq A$ *guarded* (in $\mathfrak{A}$) if there exists a tuple $\bar{a} \supseteq D$ satisfying some atomic formula $\alpha(\bar{x})$ in $\mathfrak{A}$. A *guarded bisimulation* from $\mathfrak{A}$ to $\mathfrak{B}$ is a set F of finite partial isomorphisms satisfying the following back-and-forth properties:

1. if $f : D \to E \in F$, and $D' \subseteq A$ with D' guarded, then there exist $E' \subseteq B$ and $f' : D' \to E' \in F$ such that $f{\restriction}(D \cap D') = f'{\restriction}(D \cap D')$;

2. if $f : D \to E \in F$, and $E' \subseteq B$ with E' guarded, then there exist $D' \subseteq A$ and $f' : D' \to E' \in F$ such that $f^{-1}{\restriction}(E \cap E') = f'^{-1}{\restriction}(E \cap E')$.

We say that the pointed structures $(\mathfrak{A}, \bar{a})$ and $(\mathfrak{B}, \bar{b})$ are *guarded bisimilar* if there exists a guarded bisimulation F from $\mathfrak{A}$ to $\mathfrak{B}$ and with some $f \in F$ taking $\bar{a}$ to $\bar{b}$.

The critical observation is that, for ω-saturated structures, the relation of satisfying the same formulas of $\mathcal{G}$ *is* a guarded bisimulation. More formally, let $(\mathfrak{A}, \bar{a})$ and $(\mathfrak{B}, \bar{b})$ be pointed structures with $|\bar{a}| = |\bar{b}|$. We write $(\mathfrak{A}, \bar{a}) \equiv_{\mathrm{g}} (\mathfrak{B}, \bar{b})$ if, for every $\mathcal{G}$-formula $\psi(\bar{x})$ of the appropriate arity, $\mathfrak{A} \models \psi[\bar{a}] \Rightarrow \mathfrak{B} \models \psi[\bar{b}]$.

Lemma 4.26. *If $\mathfrak{A}$ and $\mathfrak{B}$ are ω-saturated structures such that $(\mathfrak{A}, \bar{a}) \equiv_g (\mathfrak{B}, \bar{b})$, then $(\mathfrak{A}, \bar{a})$ and $(\mathfrak{B}, \bar{b})$ are guarded bisimilar.*

Proof. Let $\bar{c} = c_1, \ldots c_\ell$ be any tuple of distinct elements of A, and $\bar{d} = d_1, \ldots d_\ell$ any tuple of distinct elements of B. If $(\mathfrak{A}, \bar{c}) \equiv_{\mathrm{g}} (\mathfrak{B}, \bar{d})$, then the mapping $c_i \mapsto d_i$ $(1 \leq i \leq \ell)$ is certainly a partial isomorphism. Let F be the set of all partial isomorphisms from $\mathfrak{A}$ to $\mathfrak{B}$ constructed in this way. Then F is non-empty: in particular, it contains a partial isomorphism taking $\bar{a}$ to $\bar{b}$. We claim that F is a guarded bisimulation, from which the lemma follows.

It is immediate that F is closed under subsets, in the sense that, if $(\mathfrak{A}, \bar{c}) \equiv_{\mathrm{g}} (\mathfrak{B}, \bar{d})$, $\bar{c}_1 \subseteq \bar{c}$, and $\bar{d}_1$ is the corresponding subset of $\bar{d}$, then $(\mathfrak{A}, \bar{c}_1) \equiv_{\mathrm{g}} (\mathfrak{B}, \bar{d}_1)$. Thus, to establish the forth-condition for F, it suffices to show that if $(\mathfrak{A}, \bar{c}_1) \equiv_{\mathrm{g}} (\mathfrak{B}, \bar{d}_1)$, and $\bar{c}_2$ is a tuple from A disjoint from $\bar{c}_1$ such that $\bar{c}_1\bar{c}_2$ is guarded, then there exists a tuple $\bar{d}_2$ from B disjoint from $\bar{d}_1$ such that $(\mathfrak{A}, \bar{c}_1\bar{c}_2) \equiv_{\rm g} (\mathfrak{B}, \bar{d}_1\bar{d}_2)$. Consider then the pointed structure $(\mathfrak{B}, \bar{d}_1)$, and the set of $\mathcal{G}$-formulas (with individual constants from $\bar{d}_1$) given by

$$\Gamma(\bar{x}_2) = \{\psi(\bar{d}_1, \bar{x}_2) : \psi \in \mathcal{G} \text{ and } \mathfrak{A} \models \psi[\bar{c}_1, \bar{c}_2]\}.$$

We first claim that $\Gamma(\bar{x}_2)$ is consistent with $\mathrm{Th}(\mathfrak{B}, \bar{d}_1)$. For suppose not. By compactness, there exists a finite subset of $\Gamma(\bar{x}_2)$ not consistent with $\mathrm{Th}(\mathfrak{B}, \bar{d}_1)$. Denoting the conjunction of this finite subset by $\gamma(\bar{d}_1, \bar{x}_2)$, we have $(\mathfrak{B}, \bar{d}_1) \models \neg\exists\bar{x}_2.\gamma(\bar{d}_1, \bar{x}_2)$. On the other hand, by construction of Γ, we have, for some atomic formula $\alpha(\bar{x}_1, \bar{x}_2, \bar{x}_3)$, and some tuple $\bar{c}_3$, $\mathfrak{A} \models \alpha[\bar{c}_1, \bar{c}_2, \bar{c}_3] \wedge \gamma[\bar{c}_1, \bar{c}_2]$, and thus $(\mathfrak{A}, \bar{c}_1) \models \exists\bar{x}_2\bar{x}_3(\alpha(\bar{c}_1, \bar{x}_2, \bar{x}_3) \wedge \gamma(\bar{c}_1, \bar{x}_2))$. Since the formula

$$\exists\bar{x}_2\bar{x}_3(\alpha(\bar{x}_1, \bar{x}_2, \bar{x}_3) \wedge \gamma(\bar{x}_1, \bar{x}_2))$$

is guarded, $(\mathfrak{A}, \bar{c}_1) \equiv_{\rm g} (\mathfrak{B}, \bar{d}_1)$ implies $(\mathfrak{B}, \bar{d}_1) \models \exists\bar{x}_2\bar{x}_3(\alpha(\bar{d}_1, \bar{x}_2, \bar{x}_3) \wedge \gamma(\bar{d}_1, \bar{x}_2))$, a contradiction. Thus, $\Gamma(\bar{x}_2)$ is consistent with $\mathrm{Th}(\mathfrak{B}, \bar{d}_1)$, as claimed. Since $\mathfrak{B}$ is ω-saturated, there exists a tuple $\bar{d}_2$ such that $\mathfrak{B} \models \Gamma[\bar{d}_2]$. But then $(\mathfrak{A}, \bar{c}_1\bar{c}_2) \equiv_{\rm g} (\mathfrak{B}, \bar{d}_1\bar{d}_2)$ as required. This establishes the forth-condition for F; the back-condition is established symmetrically. □

We are now in a position to give our semantic characterization of the expressive power of $\mathcal{G}$. Its proof is almost word-for-word identical with that of Theorem 3.21.

Theorem 4.27. *A formula of first-order logic is logically equivalent to a formula of $\mathcal{G}$ if and only if it is invariant under guarded bisimulations.*

Proof. The only-if condition is proved by a simple structural induction on formulas, and is left to the reader. For the if-condition, suppose $\varphi(\bar{x})$ is invariant under guarded bisimulations. Let $\Psi = \{\psi \in \mathcal{G} : \models \varphi(\bar{x}) \rightarrow \psi(\bar{x})\}$. It suffices to show that that $\Psi(\bar{x})$ entails $\varphi(\bar{x})$. The theorem then follows by compactness, since some finite subset of $\Psi(\bar{x})$ must entail $\varphi(\bar{x})$, say $\models \psi_1(\bar{x}) \wedge \cdots \wedge \psi_n(\bar{x}) \rightarrow \varphi(\bar{x})$, with the entailment $\models \varphi(\bar{x}) \rightarrow \psi_1(\bar{x}) \wedge \cdots \wedge \psi_n(\bar{x})$ holding by the definition of Ψ.

Suppose then that $\mathfrak{A} \models \Psi[\bar{a}]$ for some tuple $\bar{a} \subseteq A$. Let $\Gamma = \{\gamma(\bar{x}) \in \mathcal{G} : \mathfrak{A} \models \gamma[\bar{a}]\}$. We claim that $\Gamma \cup \{\varphi\}$ is consistent. For if not, by compactness, there exist $\gamma_1, \ldots, \gamma_n \in \Gamma$ such that $\models \gamma_1(\bar{x}) \wedge \cdots \wedge \gamma_n(\bar{x}) \rightarrow \neg\varphi(\bar{x})$. Writing $\gamma(\bar{x})$ for the conjunction $\gamma_1(\bar{x}) \wedge \cdots \wedge \gamma_n(\bar{x})$, we have $\models \varphi(\bar{x}) \rightarrow \neg\gamma(\bar{x})$, whence $\neg\gamma \in \Psi$, whence $\mathfrak{A} \models \neg\gamma[\bar{a}]$, which is impossible, since $\mathfrak{A} \models \Gamma[\bar{a}]$. But if $\Gamma \cup \{\varphi\}$ is consistent, there exists some structure $\mathfrak{B}$ and tuple $\bar{b}$ of elements from B such that $\mathfrak{B} \models \Gamma[\bar{b}]$ (whence $(\mathfrak{A}, \bar{a}) \equiv_{\rm g} (\mathfrak{B}, \bar{b})$) and $\mathfrak{B} \models \varphi[\bar{b}]$. By Lemma 3.19, let $\mathfrak{A}^*$, and $\mathfrak{B}^*$, be ω-saturated elementary extensions of $\mathfrak{A}$ and $\mathfrak{B}$, respectively. Certainly, then $(\mathfrak{A}^*, \bar{a}) \equiv_{\rm g} (\mathfrak{B}^*, \bar{b})$, whence, by Lemma 4.26, $(\mathfrak{A}^*, \bar{a})$ and $(\mathfrak{B}^*, \bar{b})$ are guarded bisimilar. Since φ is, by hypothesis, preserved under guarded bisimulations, $\mathfrak{A}^* \models \varphi[\bar{a}]$, and hence $\mathfrak{A} \models \varphi[\bar{a}]$, which is what we were required to show. □

Concluding remarks

This chapter has presented the same array of basic results regarding guarded fragment as its predecessor did for the two-variable fragment: normal forms, upper and lower complexity bounds, finite model property, characterization of expressive power. Thus, our principal results are Theorems 4.10, 4.12 and 4.27, which together state that the satisfiability (= finite satisfiability) problem is 2-ExpTime-complete for $\mathcal{G}$ and ExpTime-complete for $\mathcal{G}^k$ ($k \geq 2$), and that a formula is logically equivalent to a guarded formula if an only if it is preserved under guarded bisimulations. Not only are these results remarkable from a purely mathematical point of view, but they derive practical importance from the fact that $\mathcal{G}^2$ includes many varieties of so-called *description logic*, which is used to model structured data in computing.

Of all the new techniques and concepts introduced in this chapter, however, the most significant—both from theoretical and a practical point of view—is that of resolution theorem proving, presented in Sec. 4.5. In this book, we shall not discuss the enormous strides made in optimizing the speed and reach of first-order theorem provers, most of which are based on variants of the resolution and factoring rules (see the Bibliographic Notes to this chapter). However, we shall frequently employ resolution theorem proving (often in conjunction with other techniques) to obtain upper complexity bounds for various logical fragments, especially in Parts III and IV of this book.

The material presented here also provides the point of departure for discussions of other decidable fragments, most particularly, the guarded two-variable fragment with *counting quantification*, $\mathcal{GC}^2$, for which the complexity of satisfiability remains in ExpTime. This topic will be taken up in Ch. **??**, and indeed developed further in Ch. **??**. Other generalizations of the guarded fragment will be treated in Ch. **??** and **??**.

Exercises

1. Complete the details of the proof of Lemma **??** in the countable case, and (using the axiom of choice) give a rigorous proof in the general case.

2. Prove Lemma 4.16.

Bibliographic notes

The guarded fragment was first identified by H. Andréka, I. Németi and J. van Benthem in their seminal publication [3], as a generalization of modal logic (see Ch. **??**). The authors are concerned to contrast the model-theoretic properties of $\mathcal{G}$ (good) with those of other decidable fragments such as quantifier prefix fragments and the two-variable fragment (bad). More significantly for our purposes, they establish the upper complexity bounds for Sat($\mathcal{G}$) and Sat($\mathcal{G}^k$) given in Sec. 4.2 (*op. cit.* pp. 256 ff.). We remark that complexity classes are not mentioned there explicitly, but may be easily read off from the proof, which we have followed fairly closely. The material on lower bounds in Sec. 4.3 is due to E. Grädel [30]. The same article contains the proof of the finite model property reproduced in 4.4. Proposition 4.11, on which this result is based, is due to B. Herwig [37], which is a generalization of an earlier result on graphs due to E. Hrushovski [41].

The literature on description logics is enormous. Sensible starting points are the authoritative and compendious [6], or the compact and accessible [7].

The refutational completeness of resolution and factoring for first-order logic (without equality) was originally established by A. Robinson [68]. The use of ordered resolution and factoring to yield decision procedures for first-order fragments was pioneered in paricular S. Maslov [56] and W. Joyner Jr. [44]; see also C. Fermüller *et al.* [22] and, for a more recent survey, C. Fermüller *et al.* [21]. The material of Sec. 4.5, which features in particular the ingenious use of multi-set extensions of literal-orderings to accommodate (arbitrary) selection functions, is taken from L. Bachmair and H. Ganzinger [10]. The presentation has been specialized to the concerns of the present volume; readers wishing for greater generality are referred to the original source. To deal with equality, the so-called *paramodulation rule* is needed, for which refutational completeness (more difficult) was proved by L. Bachmair and H.Ganzinger [9] and also J. Hsiang and M. Rusinovich [42]. The resolution-based decision procedure for $\mathcal{G}$ presented in Sec. 4.6, including the clever choice of selection function, is due to H. Ganzinger and H. de Nivelle [25]. That paper actually deals with the full guarded fragment, using the paramodulation rule to treat equalities. In the interests of brevity, we have presented a decision procedure only for the equality-free sub-fragment of $\mathcal{G}$ here; readers interested in the details of how paramodulation works in this case are again referred to the original source.

The discussion of expressive power in Sec. 4.7 employes the classical proof of [3, p. 245]. For an elementary proof, see [61]. But there is much more to be said about the topic of bisimulations; for a survey, see [32].

Chapter 5

Prefixes

A formula is in *prenex form* if none of its Boolean connectives out-scopes any of its quantifiers. Thus, for example the sentence

$$\forall x(\text{grad-student}(x) \rightarrow (\text{student}(x) \wedge \exists y.\text{supervises}(y, x)))$$

is not in prenex form, while the (logically equivalent) sentence

$$\forall x \exists y(\text{grad-student}(x) \rightarrow (\text{student}(x) \wedge \text{supervises}(y, x))), \tag{5.1}$$

in which the existential quantifier has been fronted, is. It is easily shown that every first-order formula is logically equivalent to one in prenex form; indeed, equivalent prenex forms are trivial to compute. The *quantifier prefix* of a prenex-form sentence is the word over the alphabet $\{\forall, \exists\}$ formed by its quantifiers; thus, for example, the quantifier prefix of (5.1) is $\forall\exists$. Any quantifier prefix—or more generally, any set of quantifer prefixes—defines a fragment of first-order logic, namely, the set of formulas whose prenex forms have one of the quantifier prefixes in question.

It was realized in the early stages of research into the *Entscheidungsproblem* that some fragments defined in this way indeed have decidable satisfiability problems. In 1928, P. Bernays and M. Schönfinkel showed the decidability of satisfiability for (function- and equality-free) sentences with prefixes $\exists \cdots \exists \forall \cdots \forall$ (no universal quantifier precedes any existential quantifier); in 1928 W. Ackermann did the same for the prefixes $\exists \cdots \exists \forall \exists \cdots \exists$ (one universal quantifier); and in 1933–4, K. Gödel [27], K. Schütte [69] and L. Kalmár [45] independently did the same for sentences with prefixes $\exists \cdots \exists \forall\forall \exists \cdots \exists$ (two adjacent universal quantifiers). On the other hand, Turing's proof that the satisfiability problem for first-order logic is not decidable was based on the encoding of runs of Turing machines as first-order formulas that yielded a reduction of the halting problem for Turing machines to the complement of $\text{Sat}(\mathcal{FO})$ [76]. And this approach immediately yields a corresponding undecidability results for the fragment defined by the quantifier prefix of the encoding in question. Indeed, Turing made precisely this observation, observing that the satisfiability problem for formulas with quantifier prefix $\forall\exists\forall\exists\exists\exists\exists\exists$ must also be undecidable. The realization that the satisfiability problem is decidable for some, but not all, quantifier prefixes set in train a programme of determining *which* quantifier prefixes—or which *sets* of quantifier prefixes—define fragments with decidable satisfiability problems. Similar remarks apply in the case of finite satisfiability. The present chapter presents a survey of this work.

Since a quantifier prefix is any string over the alphabet $\{\forall, \exists\}$, there are uncountably many sets of them; hence, we need to restrict the range of sets of quantifier prefixes that we can consider in this context. Say that a *standard prefix specifier* is a finite word over the four-letter alphabet $\{\forall, \exists, \forall^*, \exists^*\}$. We interpret such a string as defining a set of quantifier prefixes, where $\exists^*$ means "any number (including 0) of $\exists$s", and similarly for $\forall^*$. We denote the word $\exists \cdots \exists$ of length n ($n \geq 0$) by $\exists^n$, and similarly for $\forall$. Thus, the set of prefixes considered by Bernays and Schönfinkel is defined by the specifier $\exists^*\forall^*$; the set considered by Gödel, Schütte and Kalmár is defined by the specifier $\exists^*\forall^2\exists^*$; Turing observed the undecidability of satisfiability for the prefix $\forall\exists\forall\exists^5$; and so on. When dealing with prefix specifiers, we will always assume that obvious simplifications have been carried out: thus $\exists\exists^*$ may be replaced by $\exists^*$, and so on. We allow ourselves to write $\exists^\alpha$ as a variable prefix specifier, where α ranges over $\mathbb{N} \cup \{*\}$, (similarly for $\forall$), and take it in this context that $n < *$ for all $n \in \mathbb{N}$.

We need at this point to deal with various complications arising from the non-logical signature. We know from Sec. 3.6 that the fragment $\mathcal{FO}_{\text{Mon}}$, in which predicates of arity at most one are allowed (and no function-symbols), has the finite model property. Thus, restricting the available non-logical signature can, in principle, affect the decidability of the (finite) satisfiability problem for fragments defined using quantifier prefixes. In pratice, however, this is rare: with a few exceptions, limiting attention to monadic predicates is the *only* really non-trivial restriction of this type which restores decidability of satisfiability to such fragments, or even has any effect on complexity. Similar remarks apply to individual constants: allowing these never destroys the decidability or complexity of satisfiability for the fragments we are concerned with here. Therefore, for simplicity, we shall assume for the remainder of this chapter that predicates of all arities and individual constants are available in unlimited quantities. With function-symbols, we have the opposite situation. Function symbols have a quasi-quantificational character which, as one might expect, tends to upset the decidability of first-order fragments. In fact, with rare exceptions, adding function-symbols to the fragments considered here generally renders their satisfiability problems undecidable. Therefore, for simplicity, we shall assume for the remainder of this chapter that no function symbols are available in the non-logical signature. (Remember that we do not count individual constants as function symbols in this book.)

A further complicating issue concerns the presence or absence of equality. The result by W. Ackermann on on the decidability of satisfiability for $\exists^*\forall^2\exists^*$-sentences holds even for sentences with equality; however, the computational complexity is higher in the presence of the equality predicate. Much more dramatically, the result of K. Gödel, K. Schütte and L. Kalmár on the decidability of satisfiability for $\exists^*\forall^2\exists^*$-sentences requires that the equality predicate is not available. Indeed, it was shown by W. Goldfarb in 1984, that adding equality to this fragment yields a logic whose satisfiability and finite satisfiability problems are undecidable [28]. In this chapter, we shall consider fragments both with and without equality: not to do so would mean missing out on two of the greatest results of the whole field of decidable fragments of first-order logic, which we prove in Sec. 5.4.

Accordingly, if w is a standard prefix specifier, we denote the set of function-free, equality-free, first-order, prenex-form formulas with quantifier prefix w by $[w]$, and the the set of function-free, first-order, prenex-form formulas with quantifier prefix w by $[w]_=$. In both cases, predicates of any arity, as well as individual constants, are allowed; in the latter case, equality is allowed; in neither case are function-symbols allowed. We call such a fragment a *quantifier prefix fragment*.

As usual, for any quantifier prefix fragment $\mathcal{F}$, we are interested in the *satisfiability problem*, $\mathrm{Sat}(\mathcal{F})$, and of the finite satisfiability problem, $\mathrm{FinSat}(\mathcal{F})$: given a formula φ of $\mathcal{F}$, is φ (finitely) satisfiable? Note that these problems take as input a *single* formula of $\mathcal{F}$. This may seem surprising, since quantifier prefix fragments are not, formally speaking, closed under conjunction. One might expect an approach similar to that of Chapter 2, where we asked for the satisfiability of a given finite *set* of formulas. In the present case, however, most of the decidable fragments are *effectively closed under conjunctions*, in the sense that, given a set Φ of such sentences of $\mathcal{F}$, we can compute in polynomial time a sentence of $\mathcal{F}$ equivalent to the conjunction of Φ. Thus, we find it more convenient to treat these problems as applying to single formulas.

5.1 Existentials before universals

We begin with the simple case of $[\exists^*\forall^*]_=$, the fragment consisting of all function-free, first-order formulas (with equality) in prenex form, in which no existential quantifiers follow any universal quantifiers. This fragment is sometimes referred to as the *Ramsey fragment* or even the *Bernays-Schönfinkel-Ramsey fragment* (see Bibliographic Notes). Observe that both $[\exists^*\forall^*]_=$ and its equality-free variant $[\exists^*\forall^*]$ are effectively closed under conjunction: if $\exists\bar{x}\forall\bar{y}.\psi_1$ and $\exists\bar{u}\forall\bar{v}.\psi_2(\bar{u},\bar{v})$ are sentences with ψ_1 and ψ_2 quantifier-free, then their conjunction is logically equivalent to $\exists\bar{x}\bar{u}'\forall\bar{y}\bar{v}'(\psi_1 \wedge \psi_2(\bar{u}',\bar{v}'))$, where $\bar{u}'$ and $\bar{v}'$ are fresh tuples of variables.

Theorem 5.1. *The fragment* $[\exists^*\forall^*]_=$ *has the finite model property; the problem* $\mathrm{Sat}([\exists^*\forall^*]_=)$ *is in* NExpTime*; the problem* $\mathrm{Sat}([\forall^*])$ *is* NExpTime*-hard.*

Proof. Let a $[\exists^*\forall^*]_=$-sentence be given. Let ψ be the result of removing all existential quantifiers and replacing each existentially quantified variable by a fresh individual constant (i.e., Skolemizing). Thus, ψ is satisfiable over the same domains as φ. Since ψ is purely universal, its set of models is closed under taking sub-models. Hence, if ψ is satisfiable, it is satisfiable over a model whose domain consists only of the interpretation of the individual constants, and hence has cardinality at most $n = \|\varphi\|$. Thus, $[\exists^*\forall^*]_=$ has the finite model property. Let the number of individual constants and predicates occurring in φ be s and the maximum arity of those predicates be k. The specification of a structure with cardinality n interpreting φ certainly requires no more that $M = s \cdot (k \log n) \cdot n^k$ bits, and satisfaction of φ in such a structure can be determined in time bounded by a polynomial function of $M + n$. Obviously, $s, k \leq n$. This establishes that Sat($[\exists^*\forall^*]_=$) is in NExpTime.|

Where does the complexity of model-checking go?

To show that Sat($[\forall^*]$) is NExpTime-hard, we proceed via reduction from bounded tiling problems with exponential size-bound, as in Theorem 3.13. Fix some bounded tiling problem $P = \mathrm{BTP}\langle C, H, V, 2^x\rangle$, and let $\mathsf{c} = c_0, \ldots, c_{n-1}$ be any instance of T. We construct a formula φ_{c} of $[\forall^*]$, involving individual predicates of arity $2n$ and individual constants, such that φ_{c} is satisfiable if and only if c is a positive instance of P. Again, recall that, if k is an integer ($0 \leq k < 2^n$) with standard n-bit representation $\bar{b} = b_{n-1}, \ldots, b_0$ (b_0 is least significant), then $k' = k + 1 \mod n$ has standard n-bit representation $\bar{b}' = b'_{n-1}, \ldots, b'_0$, where, for all i ($0 \leq i < n$), $b'_i = b_i$ if and only if $b_j = 0$ for some j ($0 \leq j < i$).

Let each $c \in C$ be a $2n$-ary predicate. We imagine c to be interpreted over the domain $\{0, 1\}$, reading $p(\bar{b}, \bar{b}')$ as "the grid position with coordinates encoded by the respective n-bit strings $\bar{b}$ and $\bar{b}'$ has colour c. And let 0 and 1 be individual constants, interpreted, over the same domain, as themselves. We take Φ_{c} to contain the conjunct

$$\forall\bar{x}\forall\bar{y}\left(\bigvee_{c\in C} c(\bar{x},\bar{y}) \wedge \bigwedge_{c,d\in C}^{c\neq d} (c(x) \to \neg d(x))\right) \tag{5.2}$$

asserting that every grid position with coordinates in the range $[0, 2^n - 1]$ has exactly one colour in C. For all i ($1 \leq i < n$), we take Φ_{c} to contain the conjunct

$$\forall x_{n-1}\cdots\forall x_{i+1}\forall\bar{y} \bigwedge_{c,d\in C}^{(c,d)\notin H} (c(x_{n-1}, \ldots, x_{i+1}0, \overbrace{1, \ldots, 1}^{i-1 \text{ times}}, \bar{y}) \to \neg d(x_{n-1}, \ldots, x_{i+1}1, \overbrace{0, \ldots, 0}^{i-1 \text{ times}}, \bar{y}). \tag{5.3}$$

together, these assert that, if $(c, d) \notin H$, no grid position in the range $[0, 2^n - 2]$ coloured with c lies just to the left of a grid position in the range $[1, 2^n - 1]$ coloured with d. We also take Φ_{c} to contain the analogous conjuncts requiring vertical neighbours to conform to V. Finally,| writing $\mathsf{c} = \mathsf{c}^0 \cdots \mathsf{c}^{n-1}$, and writing "$k$" to denote the n-bit representation of the number k ($0 \leq k < 2^n$), we take $\Phi_{\bar{c}}$ be contain the conjuncts

Change super- to sub-scripts?

$$\mathsf{c}^0(\text{“}0\text{”}, \text{“}0\text{”}) \qquad \mathsf{c}^1(\text{“}1\text{”}, \text{“}0\text{”}) \qquad \cdots \qquad \mathsf{c}^{n-1}(\text{“}n-1\text{”}, \text{“}0\text{”}), \tag{5.4}$$

asserting that the n positions in a row at the bottom left corner of the grid are coloured as specified by $\mathtt{c}$.

We claim that $\Phi_{\mathtt{c}}$ is satisfiable if and only if $\mathtt{c}$ is a positive instance of the tiling problem P. For Suppose $\mathfrak{A} \models \Phi_{\mathtt{c}}$. Define $t(i,j) = c$ just in case $\mathfrak{A} \models c[\text{“}i\text{”}, \text{“}j\text{”}]$. By (5.2), t is a well-defined colouring of the grid. By (5.3), t respects the horizontal constraints, H; and similar conjuncts secure the vertical constraints. By (5.4), t respects the initial condition $\mathtt{c}$. Conversely suppose t is a (C, H, V)-tiling with initial condition $\mathtt{c}$, and let $A = \{0, 1\}$. Now define $\mathfrak{A}$ over this domain by taking each constant to interpret itself, and by declaring $\mathfrak{A} \models c[\text{“}i\text{”}, \text{“}j\text{”}]$ just in case $t(i,j) = c$, for all i, j in the range $[0, 2^m - 1]$ and all $c \in C$. It is then obvious that $\mathfrak{A} \models \Phi_{\bar{c}}$.

Now let $\varphi_{\mathtt{c}}$ be the result of moving all conjunctions in the formula $\bigwedge \Phi_{\mathtt{c}}$ inside the universal quantifiers. We see that $\varphi_{\mathtt{c}}$ is an $[\forall^*]$-formula, as required. □

We end this section with an observation on the technique of Skolemization used in Theorem 5.1. Evidently, if w is a standard prefix specifier of the form $\exists^\alpha w'$, then we can can always Skolemize the leading block of existential quantifiers in any $[w]$-sentence so as to obtain a $[w']$-sentence. Thus, given our decision to allow individual constants in the signature, establishing the decidability/complexity of of $\mathrm{Sat}([w'])$ or $\mathrm{FinSat}([w'])$ immediately yields the corresponding results for $\mathrm{Sat}([w])$ or $\mathrm{FinSat}([w])$. Similar remarks apply to $[w]_=$.

5.2 Two undecidable cases

Theorem 5.1 settles the finite model property and the complexity of satisfiability for all fragments defined by a prefix specifier in which no universal quantifier precedes any universal quantifier. In the remainder of this chapter, therefore, we confine attention to prefix specifiers in which at least one existential quantifier follows a universal quantifier. In this section, we consider two such quantifier-prefix fragments without equality, namely, $[\forall\exists\forall]$ and $[\forall\forall\forall\exists]$. We show that their satisfiability and finite satisfiability problems are undecidable.

In the former case, our fox has already been shot! Recall that, in Theorem 3.11, we showed the undecidability of satisfiability and finite satisfiability for the forgment $\mathcal{FO}^3$ by reduction to the constrained infinite, espectively, finite, tiling problems, CIT and CFT , defined in Sec. 3.3.

Theorem 5.2. *The problems* $\mathrm{Sat}([\forall\exists\forall])$ *and* $\mathrm{Sat}([\forall\exists\forall]_=)$ *are co-r.e.-complete; the problems* $\mathrm{FinSat}([\forall\exists\forall])$ *and* $\mathrm{FinSat}([\forall\exists\forall]_=)$ *are r.e.-complete.*

Proof. The formulas φ_T and $\hat{\varphi}_T$ constructed in the proof of Theorem 3.11 are in $[\forall\exists\forall]$. □

In the latter case, we again proceed via reduction from constrained tiling problems, though the reduction in this case is a little different.

Theorem 5.3. *The problems* $\mathrm{Sat}([\forall^3\exists])$ *and* $\mathrm{Sat}([\forall^3\exists])_=$ *are co-r.e.-complete; the problems* $\mathrm{FinSat}([\forall^3\exists])$ *and* $\mathrm{FinSat}([\forall^3\exists])_=$ *are r.e.-complete.*

Proof. Again, we only have to concern ourselves with lower bounds. Considering first the satisfiability problem, let a tiling system (C, V, H) and a colour $c_\alpha \in C$ be given. We compute a $[\forall\exists\forall]$-sentence φ_T, satisfiable if and only if the CIT-instance $T = (C, H, V, c_\alpha)$ is positive. It follows from Lemma 3.7 that $\mathrm{Sat}([\forall^3\exists])$ is co-r.e.-hard. We employ a pair of binary predicates, h and v, to encode grid-like structures. Specifically, we take $\mathfrak{G}$ to be a structure with domain $\mathbb{N}^2$ defined by setting: $\mathfrak{G} \models h[(i,j),(i',j')]$ just in case $i' = i+1$ and $j' = j$, and $\mathfrak{G} \models v[(i,j),(i',j')]$ just in case $i' = i$ and $j' = j+1$. Thus, we may read $h(x,y)$ as "y is immediately to the right of x" and $v(x,y)$ as "y is immediately above x". Evidently, $\mathfrak{G}$ is a model of the following sentences:

$$\forall x \exists u.(\neg h(x,x) \wedge h(x,u)) \tag{5.5}$$

$$\forall x \exists u.(\neg v(x,x) \wedge v(x,u)) \tag{5.6}$$

$$\forall x \forall y \forall z \exists u (h(x,y) \wedge v(x,z) \rightarrow v(y,u) \wedge h(z,u)). \tag{5.7}$$

Now regard the tiles of C as unary predicates, and let o be a new individual constant. If t is a function $t : \mathbb{N}^2 \rightarrow C$, define the expansion $\mathfrak{A}_t$ of $\mathfrak{G}$ by setting $\mathfrak{A}_t \models c[\langle i,j \rangle]$ if and only if $t(i,j) = c$, and setting $o^{\mathfrak{A}_f} = \langle 0,0 \rangle$. It follows that $\mathfrak{A}_t$ verifies the sentence

$$\forall x \left(\bigvee_{c \in C} c(x) \wedge \bigwedge_{c,d \in C}^{c \neq d} \neg(c(x) \wedge d(x)) \right). \tag{5.8}$$

If, in addition, f is a (C, H, V)-tiling, then $\mathfrak{A}_t$ verifies the sentences

$$\bigwedge_{c \in C} \forall x \forall z (h(x,y) \rightarrow \bigvee_{(c,d) \in H} (c(x) \wedge d(y))) \tag{5.9}$$

$$\bigwedge_{c \in C} \forall x \forall z (v(x,y) \rightarrow \bigvee_{(c,d) \in V} (c(x) \wedge d(y))). \tag{5.10}$$

Finally, if t tiles the bottom left-hand square of the grid with c_α, then $\mathfrak{A}_t$ verifies the sentence $c_\alpha(o)$. Denote by φ_T the conjunction of (5.5)—(5.10) together with $c_\alpha(o)$. We have shown that, if T is a positive instance of CIT, then φ_T is satisfiable. Conversely, suppose $\mathfrak{A} \models \varphi_T$. A straightforward induction using (5.5)–(5.7) establishes that A contains a collection of elements $A' = \{a_{i,j} \mid i,j \in \mathbb{N}\}$ such that, $a_{0,0} = o^{\mathfrak{A}}$ and, for all $i,j \in \mathbb{N}$, $\mathfrak{A} \models h[a_{i,j}, a_{i+1,j}]$ and $\mathfrak{A} \models v[a_{i,j}, a_{i,j+1}]$. That is, the elements A' are arranged by $h^{\mathfrak{A}}$ and $v^{\mathfrak{A}}$ in an infinite grid with $o^{\mathfrak{A}}$ in the bottom-left-hand corner. (There is no requirement that these grid elements be distinct.) By (5.8) we may define the function $t_{\mathfrak{A}}(i,j) = c$ where c is the unique element of C such that $\mathfrak{A} \models c[a_{i,j}]$, and by (5.9) and (5.10), $t_{\mathfrak{A}}$ is a (C, H, V)-tiling. Since $\mathfrak{A} \models c_\alpha(o)$, this tiling satisfies the initial condition c_α. We have shown that, if φ_T is satisfiable, then T is a positive instance of CIT.

We are not quite done. Each of the conjuncts of φ_T is a $[\forall^3\exists]$-formula, but φ_T itself is not: indeed, it is not even in prenex form. However, we may easily define a $[\forall^3\exists]$-formula which is logically-equivalent to φ_T. In particular, a quick check reveals that the conjunction of (5.5)—(5.7) is logically equivalent to

$$\begin{aligned}\forall x\forall y\forall z\exists u\,(\neg h(x,x) \wedge \neg v(x,x)\wedge&\\ (\neg h(x,y) \rightarrow h(x,u)) \wedge (\neg v(x,z) \rightarrow v(x,u))\wedge&\\ (h(x,y) \wedge v(x,z) \rightarrow v(y,u) \wedge h(z,u)))\,.&\end{aligned}$$

Turning now to the finite satisfiability problem, let a tiling system (C, V, H) and colours $c_\alpha, c_\omega \in C$ be given. We compute a $[\forall\exists\forall]$-sentence $\hat{\varphi}_T$, satisfiable if and only if the CFT-instance $T = (C, H, V, c_\alpha, c_\omega)$ is positive. In fact $\hat{\varphi}_T$ is a simple modification of φ_T obtained by adding an appropriate 'stop-condition' as in the proof of Theorem 3.11. Again, we have to take steps to ensure that all finite models of $\hat{\varphi}_T$ encounter the stop-predicates at some point, thus defining the margin of the grid; but with three universal quantifiers at our disposal, this is relatively easy. □

By Theorems 5.1, 5.2 and 5.3, we may henceforth confine attention to prefix specifiers of the forms $\exists^\alpha\forall\exists^\beta$ or $\exists^\alpha\forall^2\exists^\beta$, with $\beta \geq 1$. The next two sections consider these two cases in turn.

5.3 One universal quantifier

This section is devoted to consideration of the quantifier prefix fragments $[\exists^\alpha\forall\exists^\beta]$ and $[\exists^\alpha\forall\exists^\beta]_=$, with $\beta \geq 1$. In Sec. 5.3.1, we show that Sat$[\exists^*\forall\exists^*]$ is in ExpTime, and that Sat$[\forall\exists^2]$ is ExpTime-hard. Thus, Sat$([\exists^\alpha\forall\exists^\beta])$ is ExpTime-complete for all α and all $\beta \geq 2$. In Sec. 5.3.2 we show that Sat$[\exists^*\forall\exists^*]_=$ is in NExpTime, and that Sat$[\forall\exists^*]_=$ is NExpTime-hard. Thus, Sat$([\exists^\alpha\forall\exists^*]_=)$ is NExpTime-complete for all α. Finally, we show that Sat$([\exists^\alpha\forall\exists]_=)$ is in PSpace, and that Sat$([\forall\exists])$ is PSpace-hard. Thus, Sat$([\exists^\alpha\forall\exists])$ and Sat$([\exists^\alpha\forall\exists]_=)$ are PSpace-complete for all α. This leaves small gap in the complexity bounds for the fragments Sat$[\forall\exists^\beta]_=$ with $2 \leq \beta < *$. However, our proof that Sat$([\exists^*\forall\exists^*]_=)$ is in NExpTime in fact shows that this fragment—and hence all the fragments considered in this section—have the finite model property. Thus, there is no work to do in respect of finite satisfiability.

5.3.1 One universal quantifier without equality

We employ the technique of ordered resolution and factoring, familiar from Sec. 4.5. More specifically, we define a class of clauses $\mathcal{E}$, and show that, for any $[\exists^*\forall\exists^*]$-sentence φ, we may compute, in exponential time (as a function of $\|\varphi\|$), a family $\mathbf{\Gamma}$ of sets of $\mathcal{E}$-clauses, such that $\varphi \lhd \bigvee\{\forall\Gamma \mid \Gamma \in \mathbf{\Gamma}\}$, and with $\|\Gamma\|$ polynomially bounded for each $\Gamma \in \mathbf{\Gamma}$. We further show that the closure Γ^* of any set Γ of $\mathcal{E}$-clauses under ordered resolution and factoring can be computed in exponentially bounded time (again, as a function of $\|\varphi\|$). Given the soundness and completeness of ordered resolution and factoring, these steps yield an exponential-time algorithm for the problem $\mathrm{Sat}([\exists^*\forall\exists^*])$. We remark that no use is made of selected literals in the ensuing argument; that is to say, we take the selection function $\mathfrak{s}$ to satisfy $\mathfrak{s}(\gamma) = \emptyset$ for every clause γ.

The first step is to Skolemize and clausify φ, as in Lemmas 3.10 and 4.13. We remind ourselves of some definitions from Sec. 4.6: a clause is *simple* if it contains no embedded function symbols, and *covering* if every functional term occurring in it contains all the variables of the clause.

Lemma 5.4. *Let φ be an $[\exists^*\forall\exists^*]$-sentence, and let Δ be the result of Skolemizing φ and converting to clause form. Then every clause in Δ is simple and covering, and features at most one variable.*

Proof. If $\varphi = \exists\bar{x}\forall y\exists\bar{z}.\psi(\bar{x}, y, \bar{z})$ then the result of Skolemizing is $\forall y.\psi(\bar{c}, y, \bar{f}(y))$. The lemma is then immediate given that clausification introduces only non-functional literals with no additional variables. □

The next step is to further process the clauses in Δ so as to separate out the ground and non-ground literals. Define $\mathcal{E}$ to be the class of clauses γ satisfying the following conditions:

(1) all literals of γ are simple and covering;

(2) $\mathrm{vars}(L) = \mathrm{vars}(L')$ for all literals L, L' in γ.

Lemma 5.5. *Let Δ be a set of clauses featuring just one variable, y, and in which all literals are simple and covering. Then we may compute, in time bounded by an exponential function of $\|\Delta\|$, a family $\mathbf{\Gamma}$ of sets of $\mathcal{E}$-clauses, such that $\forall\Delta$ is logically equivalent to $\bigvee\{\forall\Gamma \mid \Gamma \in \mathbf{\Gamma}\}$. Moreover, the set of atoms occurring in Γ is equal to the set of atoms occurring in Δ.*

Proof. Write any $\gamma \in \Delta$ as $\gamma_g \vee \gamma_v$, where γ_g is ground and every literal of γ_v features the variable y. (We allow γ_g and γ_v to be empty.) Thus, $\forall\gamma$ is logically equivalent to $\gamma_g \vee \forall\gamma_v$. Now let $\mathbf{\Gamma}$ be the collection of clause sets Γ obtained by selecting, for each $\gamma \in \Delta$, exactly one of γ_g or γ_v. □

Observe that the following are equivalent: (i) φ is satisfiable; (ii) Δ is universally satisfiable; (iii) some $\Gamma \in \mathbf{\Gamma}$ is universally satisfiable. Thus, to determine the satisfiability of φ, it suffices to determine the universal satisfiability of each $\Gamma \in \mathbf{\Gamma}$.

To do so, we employ resolution and factoring, using the atom-ordering $\prec_d$ defined in Sec. 4.6 together with the empty selection function $\mathfrak{s}(\gamma) = \emptyset$ (no literals are selected). The crucial feature of $\mathcal{E}$-clauses is that they are preserved by $\prec_d$-ordered resolution and factoring.

Lemma 5.6. *If γ_1 and δ are $\mathcal{E}$-clauses, which resolve, under the ordering $\prec_d$ to form a clause ϵ, then ϵ is an $\mathcal{E}$-clause. If γ is an $\mathcal{E}$-clause, which factors, under the ordering $\prec_d$ to form a clause ϵ, then ϵ is an $\mathcal{E}$-clause.*

Proof. We prove the first claim only; the case of factoring follows similarly. It is immediate from the definition of $\mathcal{E}$ that eligible literals contain all the variables of their clauses, and immediate from the choice of ordering that, any eligible literal in a functional clause is functional. That ϵ is simple and covering then follows by Lemma 4.21. That it is either ground or has the same variables in all literals is obvious given that γ and δ satisfy this property. □

Now we have the sought-after result.

Lemma 5.7. *The problem* Sat($[\exists^*\forall\exists^*]$) *is in* ExpTime.

Proof. Skolemize φ and convert to clause form. By Lemma 5.4, every clause in Δ is simple and covering, and features at most one variable. Now convert to a family $\mathbf{\Gamma}$ of sets of $\mathcal{E}$-clauses as in Lemma 5.5. It suffices to show that the universal satisfiability of any $\Gamma \in \mathbf{\Gamma}$ can be tested in time bounded by an exponential function of $\|\varphi\|$.

Fixing some such Γ, let Γ^* be the closure under $\prec_d$-resolution and -factoring. By Theorem 4.19, Γ is universally satisfiable if and only if $\bot \not\in \Gamma^*$. By Lemma 5.6, Γ^* is a set of $\mathcal{E}$-clauses. In particular $d(\gamma) \leq 1$ for all $\gamma \in \Gamma^*$. Equally obviously, any atom occurring in Γ^* is a substitution instance $A\theta$ of an atom A occurring in Γ (and hence in Δ), where $y\theta$ is a term of depth at most 1 in the signature of Γ (possibly involving a different free variable). The number of such atoms $A\theta$ is therefore (up to variable renaming) polynomially bounded as a function of $\|\varphi\|$. Since each $\gamma \in \Gamma^*$ is (after deletion of repeated literals) a set of such atoms (or their negations), it follows that $|\Gamma^*|$ is bounded by an exponential function of $\|\varphi\|$. Thus, Γ^* can be computed in time bounded by an exponential function of $\|\varphi\|$, and we are done. □

Lemma 5.8. *The problem* Sat($[\forall\exists^2]$) *is* ExpTime*-hard.*

Proof. Recall the fragment $\mathcal{L}_{\text{AltVar}}$, defined in Sec. 2.4 to be be the set of first-order formulas of the forms $\exists x.\zeta$ and $\forall x\exists y.\eta$, where ζ is a quantifier-, function- and equality-free formula with free variables $\{x\}$ and η a quantifier-, function- and equality-free formula with free variables $\{x, y\}$. We showed in Lemma 2.28 that the problem of determining the satisfiability of a given *set* of formulas in this language is ExpTime-hard, even when the input contains exactly one formula, ψ, of the form $\exists x.\zeta$, and exactly two formulas, φ_1, φ_2, of the form $\forall x\exists y.\eta$. By skolemizing ψ, renaming the existentially quantified variable in φ_2, and moving quantifiers to the front, we can easily write a $[\forall\exists^2]$-sentence satisfiable over the same domains as $\psi_\wedge\varphi_1 \wedge \varphi_2$. □

Lemmas 5.7 and 5.8 yield:

Theorem 5.9. *The problem* Sat($[\exists^\alpha\forall\exists^\beta]$) *is* ExpTime*-complete for all* $\alpha \geq 0$ *and all* $\beta \geq 2$.

The formula constructed in Lemma 5.8 involves an individual constant. In fact, however, the same lower bound applies even to the sub-fragment of $[\forall\exists^2]$

without individual constants, as can be shown, for example, using an argument parallel to the proof of Lemma 5.18 below (see Exercise 5). By contrast, the assumption in Theorem 5.9 that $\beta \geq 2$ is essential. The case $\beta = 1$ will be covered in Sec. 5.3.3.

5.3.2 One universal quantifier with equality

We now turn our attention to the fragment $[\exists^*\forall\exists^*]_=$, in which equality is available, this time employing model-theoretic, rather than proof-theoretic, techniques to establish an upper complexity bound.

Lemma 5.10. *Any satisfiable $[\exists^*\forall\exists^*]_=$-formula φ has a model of cardinality bounded by a fixed exponential function of $\|\varphi\|$. Thus,* $\mathrm{Sat}([\exists^*\forall\exists^*]_=)$ *is in* NExpTime.

Proof. Let an $[\exists^*\forall\exists^*]_=$-formula φ be given. Skolemizing, we obtain a formula $\forall y.\psi$, satisfiable over the same domains; the signature Σ of ψ is the signature of φ together with additional individual constants and 1-place function-symbols. Let C be the set of individual constants in Σ, and let F be the set of 1-place function-symbols in Σ. Observe that $\|\psi\| \leq \|\varphi\|$, and that ψ is a Boolean combination of atoms whose arguments are of the forms y, c or $f(y)$ for $c \in C$ and $f \in F$. Suppose $\mathfrak{A} \models \forall y.\psi$; we construct a model $\mathfrak{B} \models \forall y.\psi$ such that $|B|$ is bounded by an exponential function of $\|\psi\|$.

Let K be the set of elements of A interpreting the individual constants C; and let K^+ be set of elements $f^{\mathfrak{A}}[c^{\mathfrak{A}}]$ for $c \in C$ and $f \in F$. Call any element of $A \setminus K$ *ordinary*, and any atom *material* if either it occurs in ψ or is of any of the forms $y = c$, $c = d$, $y = f(y)$, $f(y) = c$ or $f(y) = g(y)$, for $c, d \in C$ and $f, g \in F$. (We here identify the atoms $s = t$ and $t = s$.) Call two ordinary elements of A *equivalent* if they satisfy exactly the same material atoms. Thus, the number of equivalence classes in $A \setminus K$ is surely at most $2^{3(\|\psi\|+\|\psi\|^2)}$. From each of these equivalence classes, choose exactly one element. Call an element of $A \setminus K$ *selected* if it is either one of these chosen elements or is one of the elements of $K^+ \setminus K$. Thus, all selected elements are ordinary, and their number is at most $2^{3(\|\psi\|+\|\psi\|^2)} + \|\psi\|^2$. For all $a \in A$, define $F(a) = \{f \in F \mid f^{\mathfrak{A}}(a) \notin K \cup \{a\}\}$, and further define

$$\begin{aligned} H(a) &= \{f^{\mathfrak{A}}(a) \mid f \in F\} \setminus (K \cup \{a\}) \\ &= \{f^{\mathfrak{A}}(a) \mid f \in F(a)\}. \end{aligned}$$

Observe that, if a and a' are equivalent, then $F(a) = F(a')$. For each selected element $a \in A$, and each h $(0 \leq h < 3)$, let D be fresh set of cardinality $|H(a)|$, and $\iota_{a,h} : H(a) \to D$ a bijection. We think of D as a *copy* of $H(a)$. For all h $(0 \leq h < 3)$, let B_h be the union of all these copies $\iota_{a,h}(H(a))$ as a ranges over the *selected* elements of A. Finally, let $B = K^+ \cup B_0 \cup B_1 \cup B_2$. It follows that $|B|$ is bounded by an exponential function of $\|\psi\|$.

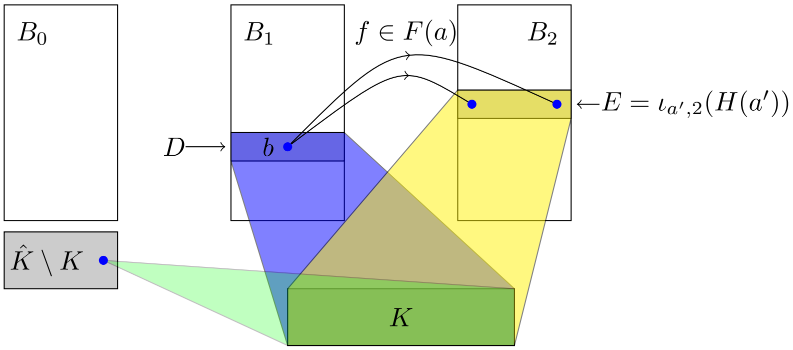


Figure 5.1: Construction of $\mathfrak{B}$ (Lemma 5.10) for $b \in D \subseteq B_1$: a' is the selected element equivalent to the element $a \in A$ corresponding to b; coloured regions indicate sub-structures copied from $\mathfrak{A}$.

We define a structure $\mathfrak{B}$ over B and show that $\mathfrak{B} \models \forall y.\psi$. We write $\lfloor n \rfloor$ to denote n modulo 3. Every element $b \in B$ is either identical to some element $a \in K^+$, or is a copy of some ordinary element a of $\mathfrak{A}$; in either case, we call a the element *corresponding to* b. Set $\mathfrak{B}{\restriction}K := \mathfrak{A}{\restriction}K$. For every $b \in K^+ \setminus K$, Set $\mathfrak{B}{\restriction}(\{b\} \cup K) := \mathfrak{A}{\restriction}(\{b\} \cup K)$. And, for any collection of elements $D = \iota_{a,h}(H(a))$ (with a selected), set $\mathfrak{B} \restriction (D \cup K) := \mathfrak{A} \restriction (H(a) \cup K)$. See Fig. 5.1. These assignments are assumed to partially define function-symbols in the sense that: (i) for all $b \in K$ and all $f \in F$, the function $f^{\mathfrak{B}}(b)$ is defined and equal to $f^{\mathfrak{A}}(b)$; and (ii), for all $b \in B \setminus K$, if a is the element of A corresponding to b, then, for all $f \in F \setminus F(a)$, $f^{\mathfrak{B}}(b)$ is defined and equal to $f^{\mathfrak{A}}(a)$.

Our strategy is to complete the construction of $\mathfrak{B}$ in such a way as to ensure that, for all $b \in B$, we can find some $a' \in A$ such that, for any atom $A(y)$ occurring in ψ, $\mathfrak{B} \models A[b]$ if and only if $\mathfrak{A} \models A[a']$. Suppose first of all that $b \in K$. From the above construction, for any atom $A(y)$ occurring in ψ, we have $\mathfrak{B} \models A[b]$ if and only if $\mathfrak{A} \models A[b]$, so we may simply set $a' = b$. Now suppose $b \in B_h$, for some h ($0 \leq h < 3$). Thus b corresponds to some (ordinary) element $a \in A$, so let a' be the selected element of A equivalent to a. As already observed, $f^{\mathfrak{B}}(b)$ has been defined for all $f \in F \setminus F(a)$. Further, if $A(y)$ is an atom occurring in ψ and all the arguments of $A(y)$ are non-functional (i.e. either y itself or an individual constant), then, from the above construction, $\mathfrak{B} \models A[b]$ just in case $\mathfrak{A} \models A[a]$ and hence (since a and a' are equivalent) just in case $\mathfrak{A} \models A[a']$. To define $f^{\mathfrak{B}}(b)$ for $f \in F(a)$, observe that $B_{\lfloor h+1 \rfloor}$ contains a copy E of $H(a')$, with $\iota_{a',\lfloor h+1 \rfloor} : H(a') \to E$ a bijection. For all $f \in F(a) = F(a')$, we set $f^{\mathfrak{B}}(b) = \iota_{a',\lfloor h+1 \rfloor}(f^{\mathfrak{A}}(a')) \in E$. Intuitively, we take the functional relations between b and E to be the obvious copy of those between a' and $H(a')$. Notice the technique of 'circular witnessing' to prevent clashes: functional values for $b \in B_h$ are found in $B_{\lfloor h+1 \rfloor}$. We claim that, if $A(y)$ is an atom occurring in ψ and y is not itself an argument of $A(y)$, then $\mathfrak{B} \models A[b]$ just in case $\mathfrak{A} \models A[a]$.

For, denoting $\iota_{a',\lfloor h+1 \rfloor}$ by ι, and writing $A(y) = \rho(c_1, \cdots c_\ell, f_1(y), \dots, f_m(y))$, we have

$$\begin{aligned} \mathfrak{B} \models A[b] &\Leftrightarrow \mathfrak{B} \models \rho[c_1^{\mathfrak{B}} \dots c_\ell^{\mathfrak{B}}, f_1^{\mathfrak{B}}(b), \dots, f_m^{\mathfrak{B}}(b)] \\ &\Leftrightarrow \mathfrak{B} \models \rho[c_1^{\mathfrak{A}} \dots c_\ell^{\mathfrak{A}}, \iota(f_1^{\mathfrak{A}}(a')), \dots, \iota(f_m^{\mathfrak{A}}(a'))] \\ &\Leftrightarrow \mathfrak{A} \models \rho[c_1^{\mathfrak{A}} \dots c_\ell^{\mathfrak{A}}, f_1^{\mathfrak{A}}(a'), \dots, f_m^{\mathfrak{A}}(a')] \\ &\Leftrightarrow \mathfrak{A} \models \rho[c_1^{\mathfrak{A}} \dots c_\ell^{\mathfrak{A}}, f_1^{\mathfrak{A}}(a)), \dots, f_m^{\mathfrak{A}}(a)]. \end{aligned}$$

with the final step holding because a and a' are equivalent. Any remaining atoms $A(y)$ occurring in ψ not considered so far have y and at least one $f(y)$ among their arguments, with $f \in F(a') = F(a)$. But $\mathfrak{B}$ has not yet been defined on any tuple involving both b and $f^{\mathfrak{B}}(b) = \iota(f^{\mathfrak{A}}(a'))$, and so we are free to set $\mathfrak{B} \models A[b]$ if and only if $\mathfrak{A} \models A[a']$. Finally, suppose $b \in \hat{K} \setminus K$. Since these elements are all by definition selected, we take $a' = a = b$ and proceed just as we did in the previous case, with h chosen arbitrarily—say, $h = 0$—finding functional values in B_1.

At the end of this process, the (partially defined) structure $\mathfrak{B}$ has the desired property that, for all $b \in B$, we can find some $a' \in A$ such that, for any atom $A(y)$ occurring in ψ, $\mathfrak{B} \models A[b]$ if and only if $\mathfrak{A} \models A[a']$. Since $\mathfrak{A} \models \forall y.\psi$, it follows that, however $\mathfrak{B}$ is completed, $\mathfrak{B} \models \forall y.\psi$, which is what we require. □

The corresponding lower bound again uses the technique of bounded tiling problems with exponential bound.

Lemma 5.11. *The problem* Sat($[\forall\exists^*]_=$) *is* NExpTime*-hard.*

Proof. Fix some bounded tiling problem $P = \text{BTP}\langle C, H, V, 2^x\rangle$, and let $\mathsf{c} = c_0, \dots, c_{n-1}$ be any instance of P. We construct a $[\exists^*\forall\exists^*]_=$-formula φ_{c} such that φ_{c} is satisfiable if and only if there is a $2^n \times 2^n$-sized tiling for (C, H, V) with initial condition c. This proves the theorem. Recall that every integer k in the range $[0, 2^n - 1]$ has a unique bit-string representation $b_{n-1} \cdots b_0$, with b_0 the least significant bit. Recall, as usual, that if k' has bit-string representation $b'_{n-1} \cdots b'_0$, then $k' = k + 1 \mod 2^m$ just in case, for all i $(0 \leq i < m)$ $b_i = b'_i$ if and only the bits $b_0, \dots, b_{i-1}$ are not all 1. To aid readability, we construct a conjunction $\bigwedge \Phi_{\mathsf{c}}$ of universally quantified formulas in a language with unary function-symbols and individual constants; we show how to obtain the desired $\varphi_{\mathsf{c}} \in [\exists^*\forall\exists^*]_=$ at the end.

Let $p_0, \dots, p_{n-1}$, $q_0, \dots, q_{n-1}$ be unary predicates. We imagine these predicates to be interpreted over squares (k, k') of the $2^n \times 2^n$ grid, reading $p_i(y)$ as "the ith bit of the horizontal co-ordinate of y is 1, and $q_i(y)$ as "the ith bit of the vertical co-ordinate of y is 1, for all i $(0 \leq i < n)$. We now duplicate this representation using individual constants 0, 1 and unary function symbols $h_0, \dots, h_{n-1}$ and $v_0, \dots, v_{n-1}$. Specifically, we take Φ_{c} to include, for each i $(0 \leq i < n)$, the two conjuncts

$$\begin{aligned} &\forall y[(p_i(y) \leftrightarrow h_i(y) = 1) \wedge (\neg p_i(y) \leftrightarrow h_i(y) = 0)] \\ &\forall y[(q_i(y) \leftrightarrow v_i(y) = 1) \wedge (\neg q_i(y) \leftrightarrow v_i(y) = 0)]. \end{aligned}$$

It follows from these formulas that h_i and v_i always take values in $\{0, 1\}$. We also employ unary functions H and V. We read $H(y)$ as "the square just to the right of y" (arbitrarily interpreted if the horizontal co-ordinate of y is $2^n - 1$) and $V(y)$ as "the square just above y" (arbitrarily interpreted if the vertical co-ordinate of y is $2^n - 1$). To secure this reading for H, we take Φ_{c} to include, for each i $(0 \leq i < n)$, the two conjuncts

$$\forall y \left[\left(\left(\bigwedge_{j=0}^{i-1} p_j(y) \right) \wedge \neg p_i(y) \right) \rightarrow \right.$$
$$\left. \left(\bigwedge_{j=0}^{i-1} \neg p_j(H(y)) \right) \wedge p_i(y) \wedge \left(\bigwedge_{j=i+1}^{n-1} (p_j(y) \leftrightarrow p_j(H(y))) \right) \right]$$

$$\forall y \, [q_i(y) \leftrightarrow q_i(H(y))] \, .$$

We add similar conjuncts for V. It should be obvious that, in any model of Φ_{c}, there exist elements with all possible horizontal and vertical coordinates in the ranges $[0, 2^n - 1]$. Of course, there is nothing to prevent two different elements from having identical coordinates, and these elements need not be arranged in a grid by the functions H and V.

We take every $c \in C$ to be a unary predicate and also a $2n$-ary predicate. This 'overloading' of symbols need cause no confusion, since formula syntax will always make it clear which predicate we mean. We read the unary atom $c(y)$ as "y is coloured with tile c", and add to Φ_{c} the two conjuncts

$$\forall y \bigvee_{c \in C} c(y) \qquad \qquad \forall y \bigwedge_{c,d \in C}^{c \neq d} \neg(c(y) \wedge d(y))$$

to ensure that each element has a unique colour. What we need is some mechanism ensuring that the colour of an element depends only on its coordinates, and for this we employ the $2n$-ary predicates. For each $c \in C$, we add to Φ_{c} the conjunct

$$\forall y (c(y) \leftrightarrow c(h_{n-1}(y), \ldots, h_0(y), v_{n-1}(y), \ldots, v_0(y)).$$

This clearly has the desired effect.

We then encode the horizontal tiling constraints by adding to Φ_{c} the formula

$$\forall y \bigwedge_{c \in C} \left(\left(c(y) \wedge \bigvee_{j=0}^{n-1} \neg p_j(y) \right) \rightarrow \bigvee_{d:(c,d) \in H} d(H(y)) \right)$$

and similarly for the vertical tiling constraints. Finally, writing $\mathsf{c} = \mathsf{c}^0 \cdots \mathsf{c}^{n-1}$, we encode the condition that elements with coordinate (0,0) are coloured c^0, by adding to Φ_{c} the formula

$$\forall y \left(\bigwedge_{i=0}^{n-1} (\neg p_i(y) \wedge \neg q_i(y)) \rightarrow \mathsf{c}^0(y) \right);$$

similarly, *mutatis mutandis*, for the elements $\mathsf{c}^1, \ldots, \mathsf{c}^{n-1}$.

It is simple to check that Φ_{c} has a model if and only if there is a $2^n \times 2^n$-sized tiling for (C, H, V) with initial condition c. It remains to coerce Φ_{c} into an equisatisfiable Sat($[\exists^*\forall\exists^*]_=$)-formula. But this is easy: let $\forall y.\hat{\psi}_{\mathsf{c}}(y)$ be the result of taking the conjunction $\bigwedge \Phi_{\mathsf{c}}$ and moving the universal quantifier to the front. Renumber the function symbols occurring in ψ_{c} as $f_1, \ldots f_s$. Let ψ_{c} be the result of replacing the individual constants 0 and 1 by x_1 and x_2, respectively, and replacing any occurrence of $f_k(y)$ by z_k $(1 \leq k \leq s)$. Then $\forall x.\hat{\psi}_{\mathsf{c}}$ is the Skolemization of the formula $\varphi_{\mathsf{c}} := \exists x_1 \exists x_2 \forall y \exists z_1, \cdots \exists z_s.\psi_{\mathsf{c}}$. This proves the lemma. □

Lemmas 5.10 and 5.11 yield:

Theorem 5.12. *The fragment* $[\exists^*\forall\exists^*]_=$ *has the finite model property. The problem* Sat($[\exists^\alpha\forall\exists^*]_=$) *is* NExpTime*-complete for all* $\alpha \geq 0$.

The formulas constructed in Lemma 5.11 involve two individual constants and an unbounded number of trailing existential quantifiers. It does not seem possible to eliminate either. Thus, when considering sub-fragments without individual constants, we have NExpTime-hardness only when $\alpha \geq 2$. Moreover, the exact complexity of the problems Sat($[\exists^\alpha\forall\exists^\beta]_=$), for finite $\beta \geq 2$, appears to be open. By Lemma 5.8, these problems are certainly ExpTime-hard, and by Lemma 5.10, they are in NExpTime. The case $\beta = 1$ is dealt with in Sec. 5.3.3.

5.3.3 Just one trailing existential

We round off our discussion of prefix specifiers of the forms $[\exists^\alpha\forall\exists^\beta]$ by considering the only remaining case: $\beta = 1$. We show that the problems Sat($[\exists^\alpha\forall\exists]$) and Sat($[\exists^\alpha\forall\exists]_=$) are both PSpace-complete. The work is admittedly routine, but it does make the point that the lower bounds obtained above require larger values of β: Lemma 5.8 makes essential use of the fact that $\beta \geq 2$, while Lemma 5.11 requires that $\beta = *$: that is, we need two (respectively, unboundedly many) trailing existential quantifiers to provide reductions from arbitrary problems in the relevant complexity classes.

Theorem 5.13. *The problems* Sat($[\exists^\alpha\forall\exists]$) *and* Sat($[\exists^\alpha\forall\exists]_=$) *are* PSpace*-complete.*

Proof. To show membership of Sat($[\exists^\alpha\forall\exists]_=$) in PSpace, we describe a nondeterministic procedure, running in time bounded by a polynomial function of the size of its input, which, when given a $[\exists^*\forall\exists]$-sentence φ, has a successful run if and only if φ is satisfiable. The result then follows from Savitch's Theorem

(Proposition 1.4). We may assume by replacing any individual constants with fresh existentially quantified variables (i.e. de-Skolemizing) that φ does not feature any individual constants. We use the following terminology. Let φ have the form $\exists u_1 \cdots \exists u_m \forall x \exists y . \psi$, where ψ is quantifier- (and equality-) free. Let Ψ be the set of formulas (with free variables from among $\bar{w} = u_1, \ldots, u_m, x, y$) that are either sub-formulas of ψ or the negations of sub-formulas of ψ. If $\bar{v} \subseteq \bar{w}$ is a k-tuple of distinct variables, let $\Psi \restriction \bar{v}$ denote the the set of formulas in Ψ featuring no variables outside $\bar{v}$. We say that an $(m+2)$-*sort* is a maximal consistent subset of Ψ, and, more generally, that a k-*sort* over some k-tuple of distinct variables $\bar{v} \subseteq \bar{w}$, is a maximal consistent subset of $\Psi \restriction \bar{v}$. If $\mathfrak{A}$ is a structure interpreting the signature of φ and $\bar{v} \subseteq \bar{w}$ a k-tuple of distinct variables, then every k-tuple $\bar{a}$ of elements of A satisfies a unique k-sort over $\bar{v}$, which we denote $\text{sort}^{\mathfrak{A}}[\bar{a}]$. Intuitively, we can think of a k-sort as being rather like a k-type, except that we consider only sub-formulas of ψ. In particular, there are at most $N = 2^{\|\psi\|}$ $(m+2)$-sorts. If τ is an $(m+2)$-sort we write $\tau \restriction (\bar{u}, y)$ to denote the $(m+1)$-sort obtained by discarding any formulas in τ involving the variable x. Further, we write $\tau \restriction (\bar{u}, y)[y/x]$ to be the result of substituting x for y in $\tau \restriction (\bar{u}, y)$; notice that this will not in general be an $(m+1)$-sort. The satisfiability-checking procedure is as follows (we write $u_1, \ldots u_m$ as $\bar{u}$):

```
1.  begin [$\exists^*\forall\exists$]-test($\varphi$)
2.     guess $\rho(\bar{u})$, an $m$-sort
3.     for $i = 1 \ldots, m$
4.        guess $\pi(\bar{u}, x)$ an $(m+1)$-sort
5.        if $\rho \wedge \pi(\bar{u}, u_i)$ is not consistent
6.           return No
7.        for $j = 0 \ldots, N$
8.           guess $\tau(\bar{u}, x, y)$, an $(m+2)$-sort
9.           if $\tau \wedge \pi \wedge \psi$ is not consistent
10.             return No
11          let $\pi := \tau\restriction(\bar{u}, y)[y/x]$
12.    return Yes
13. end [$\exists^*\forall\exists$]-test.
```

Bearing in mind that the number N requires only polynomially many bits to store, it is obvious that this procedure runs in polynomial space. Suppose $\mathfrak{A} \models \varphi$: we describe a run of $[\exists^*\forall\exists]$-`test`($\varphi$) which terminates successfully. Let $\bar{a} = a_1, \ldots a_m$ be a tuple witnesses for the leading existential quantifiers in φ. In line 2, guess $\rho = \text{sort}^{\mathfrak{A}}[\bar{a}]$. The algorithm enters the `for`-loop in lines 3–11. For each execution of line 4 (with given value of the parameter i), guess $\pi = \text{sort}^{\mathfrak{A}}[\bar{a}, a_i]$, Thus, the condition in line 5 will not be satisfied, and the algorithm enters the inner `for`-loop in lines 7–11. For the ensuing argument, we define $b_i^0 = a_i$. In each execution of line 8 (with given values of the parameters i and j), and assuming that b_i^j has been defined in such a way that $\mathfrak{A} \models \pi[\bar{a}, b_i^j]$, select some b_i^{j+1} such that $\mathfrak{A} \models \psi[\bar{a}, b_i^j, b_i^{j+1}]$ (always possible since $a_1, \ldots, a_m$ are witnesses for φ), and guess $\tau = \text{sort}^{\mathfrak{A}}[\bar{a}, b_i^j, b_i^{j+1}]$.

Thus, the condition in line 9 will not be satisfied; moreover, line 11 ensures that $\mathfrak{A} \models \pi[\bar{a}, b_i^{j+1}]$. Hence the procedure terminates with success. Conversely, suppose some run of $[\exists^*\forall\exists]$-`test`$(\varphi)$ terminates with success. Let $\bar{a} = a_1, \ldots, a_m$ be distinct individual constants, f a unary function symbol, and A the set of terms $\{f^{(j)}(a_i) \mid 1 \leq i \leq m,\ j \in \mathbb{N}\}$, where $f^{(0)}(a_i)$ denotes a_i, and $f^{(j)}(a_i)$, the j-fold application of f to a_i for $j \geq 1$. We turn A into a model $\mathfrak{A} \models \varphi$ as follows. Fixing some i $(1 \leq i \leq m)$, consider the sequence of values guessed for τ in the various executions of line 8 (one for each iteration of the inner `for`-loop), say $\tau_{i,0}, \tau_{i,1}, \ldots, \tau_{i,N}$. Let j_1 be the smallest value such that $\tau_{i,j_1} = \tau_{i,j_0}$ for some $j_0 < j_1$. (This must happen, since we eventually run out of $(m+2)$-sorts.) Still fixing i, select $\mathrm{tp}^{\mathfrak{A}}[\bar{a}, f^{(j)}(a_i), f^{(j+1)}(a_i)]$ to be consistent with $\tau_{i,j}$ for $0 \leq j < j_1$, and then cycle through the values $\tau_{i,j_0}, \ldots \tau_{i,j_1-1}$ indefinitely. (Formally: for $j \geq j_0$, select $\mathrm{tp}^{\mathfrak{A}}[\bar{a}, f^{(j)}(a_i), f^{(j+1)}(a_i)]$ to be consistent with $\tau_{i,j_0+(j-j_0 \mod j_1-j_0)}$.) Having fixed these types for all i $(1 \leq i \leq m)$, complete $\mathfrak{A}$ arbitrarily. It is immediate from the condition in line 9 that $\mathfrak{A} \models \varphi$.

To show PSpace-hardness of $\mathrm{Sat}([\exists^\alpha\forall\exists])$, we encode runs of polynomial-space bounded deterministic Turing machines directly using the same technique as in Lemma 5.8 (which makes use of Lemma 2.28). Since we are dealing with *deterministic* Turing machines, we do not require to divide set available transitions into sets T_L and T_R as we did in the proof of Lemma 2.28. Therefore, we need only one Skolem function to encode successor configurations. The modifications to the proof are routine and left to the reader. □

5.4 Two universal quantifiers

This section contains two big results. The first is that the fragment $[\exists^*\forall^2\exists^*]$ (the so-called *Gödel fragment*) has the finite model property and that its satisfiability problem is NExpTime-complete. The second is that the satisfiability and finite satisfiability problems for the fragment $[\exists^*\forall^2\exists^*]_=$ are both undecidable.

5.4.1 Two universal quantifiers without equality

We begin with some routine work. The following simple technique will be useful at various points in this book. Let $\mathfrak{B}$ be a structure interpreting a purely relational signature over some domain B, and let n be a positive integer. Then we may form the structure $\mathfrak{B}'$ over the domain $B \times \{0, \ldots, n-1\}$ by declaring, for any k-ary predicate r, $\mathfrak{B}' \models r[\langle b_1, i_1\rangle, \ldots, \langle b_k, i_k\rangle]$ if and only if $\mathfrak{B} \models r[b_1, \ldots, b_k]$, for any elements $b_1, \ldots, b_k \in B$ and integers $i_1, \ldots, i_k \in \{0, \ldots, n-1\}$. We denote $\mathfrak{B}'$ by $\mathfrak{B} \cdot n$, and call it the *n-fold Cartesian product of* $\mathfrak{B}$. The following lemma may be proved by a simple induction on the structure of formulas.

Lemma 5.14. *Let $\psi(\bar{x})$ be an equality-free formula over a purely relational signature Σ. If $\mathfrak{B}$ is a structure interpreting Σ, $\bar{b} = b_1, \ldots, b_k$ a tuple from B and $i_1, \ldots, i_k$ a tuple from $\{0, \ldots, n-1\}$, then $\mathfrak{B} \models \psi[\bar{b}]$ if and only if $\mathfrak{B} \cdot n \models \psi[\langle b_1, i_1\rangle, \ldots, \langle b_k, i_k\rangle]$.*

We observed above that, by Skolemization, we may ignore leading existential quantifiers. That is, we may concern ourselves with the fragment $[\forall\forall\exists^*]$. The following three lemmas show that, even in this restricted case, we may dispense with the individual constants. Somewhat confusingly, the first two lemmas reintroduce equality to the fragment; but the third eliminates it again.

Let $\varphi := \forall x_1 \forall x_2 \exists x_3 \cdots \exists x_n . \psi$ be a formula of $[\forall\forall\exists^*]$, and featuring no individual constants other than $c_1, \ldots, c_m$. Let $\bar{x}$ be the tuple of variables $x_3, \ldots, x_n$. Furthermore, for all h $(1 \leq h \leq m)$, let $\bar{x}'_h = x'_{h,3}, \ldots, x'_{h,n}$ and $\bar{x}''_h = x''_{h,3}, \ldots, x''_{h,n}$ be $(n-2)$-tuples of fresh variables; and for all h, h' $(1 \leq h, h' \leq m)$, let $\bar{x}'''_{h,h'} = x'''_{h,h',3}, \ldots, x'''_{h,h',n}$ be an $(n-2)$-tuple of fresh variables. Denote the concatenation of all the tuples $\bar{x}'_h$ (in some order) by $\bar{x}'$, and similarly for $\bar{x}'''$ and $\bar{x}'''$. Now write

$$\begin{aligned}
\psi' &:= \bigwedge_{h=1}^{m} \psi(c_h, x_2, \bar{x}'_h), \\
\psi'' &:= \bigwedge_{h=1}^{m} \psi(x_1, c_h, \bar{x}''_h), \\
\psi''' &:= \bigwedge_{h=1}^{m} \bigwedge_{h'=1}^{m} \psi(c_h, c_{h'}, \bar{x}'''_{h,h'}),
\end{aligned}$$

and finally define φ' to be

$$\forall x_1 \left(\bigwedge_{h=1}^{m} x_1 \neq c_h \rightarrow \forall x_2 \left(\bigwedge_{h=1}^{m} x_2 \neq c_h \rightarrow \exists \bar{x} \bar{x}' \bar{x}'' \bar{x}''' (\psi \wedge \psi' \wedge \psi'' \wedge \psi''') \right) \right).$$

Lemma 5.15. *Let some $[\forall^2\exists^*]$-sentence φ be given, and construct φ' as just described. Then φ and φ' are logically equivalent.*

Proof. Immediate. □

Lemma 5.16. *Let χ be an equality-free formula, and let the formulas φ' and φ'' be given by*

$$\begin{aligned}
\varphi' &:= \forall x_1 \left(\bigwedge_{h=1}^{m} x_1 \neq c_h \rightarrow \forall x_2 \left(\bigwedge_{h=1}^{m} x_2 \neq c_h \rightarrow \exists \bar{y} . \chi \right) \right) \\
\varphi'' &:= \forall x_1 \left(\bigwedge_{h=1}^{m} x_1 \neq c_h \rightarrow \forall x_2 \left(\bigwedge_{h=1}^{m} x_2 \neq c_h \rightarrow \exists \bar{y} \left(\bigwedge_{y \in \bar{y}} \bigwedge_{h=1}^{m} y \neq c_h \wedge \chi \right) \right) \right).
\end{aligned}$$

Then φ' is (finitely) satisfiable if and only if φ'' is.

Proof. Trivially, $\models \varphi'' \rightarrow \varphi'$. Conversely, suppose $\mathfrak{A} \models \varphi'$. Let the denotations of the constants $c_1, \ldots, c_m$ in $\mathfrak{A}$ be $o_1, \ldots, o_m$, and let $\mathfrak{A}'$ be the reduct of $\mathfrak{A}$ obtained by discarding the individual constants. Now let $\mathfrak{B}'$ be the 2-fold Cartesian product $\mathfrak{A}' \cdot 2$, and $\mathfrak{B}$ the expansion of $\mathfrak{B}'$ obtained by interpreting each individual constant c occurring in φ as $\langle c^{\mathfrak{A}}, 0\rangle$. Writing $\chi'(\bar{z}, \bar{x}, \bar{y})$ for the result of replacing the $c_1, \ldots, c_m$ in χ by fresh variables $z_1, \ldots, z_m$, and taking $\bar{y}$ to be of length ℓ, we see that, for all $a_1, a_2 \in A$, there exist $b_1, \ldots, b_\ell$ (depending on a_1 and a_2) such that $\mathfrak{A}' \models \chi'[o_1, \ldots, o_m, a_1, a_2, b_1, \ldots, b_\ell]$, whence, by Lemma 5.14,

$$\mathfrak{B}' \models \chi'[\langle o_1, 0\rangle, \ldots, \langle o_m, 0\rangle, \langle a_1, i_1\rangle, \langle a_2, i_2\rangle, \langle b_1, 1\rangle, \ldots, \langle b_\ell, 1\rangle],$$

for all values i_1, i_2 chosen from $\{0, 1\}$. That is: $\mathfrak{B} \models \varphi''$. □

Consider any formula φ'' of the form given in Lemma 5.16, featuring no individual constants other than $c_1, \ldots, c_m$. Any atom α of χ is of the form $r(\bar{w}_0, \bar{c}_1, \bar{w}_1, \ldots, \bar{c}_t, \bar{w}_t)$, where r is a predicate, each $\bar{w}_i$ is a k_i-tuple of variables, and each $\bar{c}_i$ is a non-empty tuple of variables. Here, we allow k_0 and k_t to be zero, but insist that $k_1, \ldots k_{t-1}$ all be positive. Let us write $s(\alpha)$ for the sequence $k_0, \bar{c}_1, k_1, \ldots, \bar{c}_t, k_t$ (an alternating integers and words over individual constants), thus obtained. For example, if α is $p(x_1, c_1, x_2, x_1, c_3, c_4, c_1)$, then $s(\alpha) = 1, c_1, 2, c_3, c_4, c_1, 0$. Now define a new predicate $r_{s(\alpha)}$ of arity $(k_0 + \cdots + k_t)$. Note that the arity of $r_{s(\alpha)}$ is given by the number of variables in α (counting repeats); in particular, $r_{s(\alpha)}$ is a proposition letter just in case α is ground. Now let χ^* be the result of replacing each atom $\alpha = r(\bar{w}_0, \bar{c}_1, \bar{w}_1, \ldots, \bar{c}_t, \bar{w}_t)$ by the new atom $r_{s(\alpha)}(\bar{w}_0, \ldots, \bar{w}_t)$, and define

$$\varphi''' := \forall x_1 \forall x_2 \exists x_3 \cdots \exists x_n . \chi^*.$$

Lemma 5.17. *Let φ'' have the form given in Lemma 5.16, and let φ''' be as just defined. Then φ'' is (finitely) satisfiable if and only if φ''' is.*

Proof. Suppose $\mathfrak{A} \models \varphi''$. Let C be the elements interpreting the individual constants in $\mathfrak{A}$, and let $B = A \setminus C$. Define a model $\mathfrak{B}$ interpreting the signature of φ''' over B by setting, for each atom $\alpha = r(\bar{w}_0, \bar{c}_1, \bar{w}_1, \ldots, \bar{c}_t, \bar{w}_t)$ in φ',

$$\mathfrak{B} \models r_{s(\alpha)}[\bar{b}_0, \ldots, \bar{b}_t] \text{ iff } \mathfrak{A} \models r[\bar{b}_0, \bar{c}_1, \bar{b}_1, \ldots, \bar{c}_t, \bar{b}_t],$$

completing the model arbitrarily. It is then simple to verify that $\mathfrak{B} \models \varphi'''$.

Suppose conversely $\mathfrak{B} \models \varphi'''$. Let C be the set of individual constants occurring in φ'', and let $A = B \cup C$. Define a model $\mathfrak{A}$ interpreting the signature of φ'' over A as follows: (i) for each constant in $c \in C$, set $c^{\mathfrak{A}} = c$; (ii) for each atom $\alpha = r(\bar{w}_0, \bar{c}_1, \bar{w}_1, \ldots, \bar{c}_t, \bar{w}_t)$ in φ', and each sequence of tuples $\bar{b}_0, \ldots, \bar{b}_t$ from B, where $\bar{b}_k$ has the same length as $\bar{w}_k$), set

$$\mathfrak{A} \models r[\bar{b}_0, \bar{c}_1, \bar{b}_1, \ldots, \bar{c}_t, \bar{b}_t] \text{ iff } \mathfrak{B} \models r_{s(\alpha)}[\bar{b}_0, \ldots, \bar{b}_t];$$

(iii) complete the extensions of the predicates in regard to any other tuples arbitrarily. Note that there can be no double definition here. It is then simple to verify that $\mathfrak{A} \models \varphi''$. □

It is clear that, given any sentence φ in the fragment $[\forall\exists^*]$, we may compute the sentences φ', φ'' and φ''' given in Lemmas 5.15– 5.17 in polynomial time. Thus, we have established:

Lemma 5.18. *The (finite) satisfiability problem for* $[\exists^*\forall^2\exists^*]$ *is reducible in polynomial time to the (finite) satisfiability problem for the sub-fragment of* $[\forall^2\exists^*]$ *without individual constants.*

With these details behind us, we turn to the key ideas of our proof. Recall the notion of a n-type τ (over a relational signature Σ) for $n \geq 1$: a maximal, consistent set of non-equality Σ-literals in the variables $x_1, \ldots, x_n$. In Sec. 4.2, we learned to think of τ as a *structure* over the domain $\{x_1, \ldots, x_n\}$: for any word w over the alphabet $\{x_1, \ldots, x_n\}$ and any predicate r a of Σ with arity $|w|$, we write $\tau \models r[w]$ just in case the positive literal $r(w)$ is in τ. It will be useful to keep this view of n-types in mind throughout the remainder of Sec. 5.4.1.

Also useful for representing and reasoning about types is the following notation. Fix some n-type τ over a relational signature Σ, and again let w be a word over the alphabet $\{x_1, \ldots, x_n\}$, of length $m \geq 1$. Overloading notation in the obvious way, we may write $\bar{w} = x_{w(1)}, \ldots, x_{w(m)}$. We now proceed to define an m-type, which we shall denote $\tau\langle w\rangle$. If r is a k-ary predicate of Σ and $w' = x_{w'(1)}, \ldots, x_{w'(k)}$ a word of length k over the alphabet $\{x_1, \ldots, x_m\}$, we take it that

$$r(\bar{w}') \in \tau\langle w\rangle \text{ iff } r(x_{w(w'(1))}, \ldots, x_{w(w'(k))}) \in \tau.$$

This rather opaque definition becomes much more transparent if we think of τ as a structure over the domain $\{x_1, \ldots, x_n\}$. Letting $\bar{w}_0$ be the tuple obtained by removing any repeats from $\bar{w}$, we consider $\tau_0 = \mathrm{tp}^{\tau}[\bar{w}_0]$, the type of $\bar{w}_0$ in the structure τ. The type τ_0 can also be regarded as a structure, over some domain, say, $\{x_1, \ldots, x_\ell\}$, where $\ell = |\bar{w}_0|$. Now form the m-ary Cartesian product $\tau_0 \cdot m$. Then $\tau\langle w\rangle$ is simply the type of $\langle w_1, 0\rangle, \ldots, \langle w_m, m-1\rangle$ in $\tau_0 \cdot m$. Still more informally, it is the type obtained from τ_0 by pretending that repeated elements in $\bar{w}$ as different. As an illustration, suppose τ is a k-type ($k \geq 1$). Then: (i) $\tau\langle x_1\rangle$ is the 1-type obtained by taking those literals of τ featuring only x_1; (ii) $\tau\langle x_2\rangle$ is the 1-type obtained by of taking those literals of τ featuring only x_2 *and then renaming* x_2 *as* x_1; (iii) $\tau\langle x_1, x_3, x_4, \ldots, x_k\rangle$ is the $(k-1)$-type obtained by removing from τ all literals featuring x_2 and re-naming the variable x_i as x_{i-1} for all i ($3 \leq i \leq k$); (iv) $\tau\langle x_1, x_1, x_3, x_4, \ldots, x_k\rangle$ is the k-type obtained from the $(k-1)$-type $\tau\langle x_1, x_3, x_4, \ldots, x_k\rangle$ by restoring the original variable names x_i as $3 \leq i \leq k$ and 'cloning' the variable x_1 as x_2. Remember: $\tau\langle\bar{w}\rangle$ is always a k-type, where $k = |\bar{w}|$, and k-types are always over the variables $x_1, \ldots, x_k$. (You have been warned.)

Keeping Σ to be a finite, purely relational signature, suppose that the predicates of Σ all have arity at most n, for some integer n. Without loss of generality, we may assume $n \geq 3$. In the sequel, all types, of whatever arity, are assumed to be over Σ. Further, let $\psi = \psi(x_1, \ldots, x_n)$ be a quantifier- and equality-free formula over Σ, and let $\varphi := \forall x_1 \forall x_2 \exists x_3 \cdots \exists x_n.\psi$. A *certificate* for φ is a triple $\mathfrak{C} = (R, S, T)$, where R is a set of 1-types, S a set of 2-types, and T a set of n-types consistent with ψ, subject to the following properties:

C1 for every $\rho_1, \rho_2 \in R$, there exists $\sigma \in S$ such that $\rho_1 = \sigma\langle x_1\rangle$ and $\rho_2 = \sigma\langle x_2\rangle$;

C2 for every $\sigma \in S$ there exists $\tau \in T$ such that $\sigma = \tau\langle x_1, x_2\rangle$;

C3 for every $\rho \in R$, there exists $\tau \in T$ such that $\rho = \tau\langle x_1\rangle$ and $\tau = \tau\langle x_1, x_1, x_3, \ldots, x_n\rangle$;

C4 for every $\tau \in T$ and every $(1 \leq i \leq n)$, $\tau\langle x_i\rangle \in R$;

C5 for every $\tau \in T$ and every $(1 \leq i, j \leq n)$, $\tau\langle x_i, x_j\rangle \in S$.

Lemma 5.19. *A formula φ of the fragment $[\forall\forall\exists^*]$ with no individual constants is satisfiable if and only if it has a certificate.*

Proof. Let us write $\varphi := \forall x_1 \forall x_2 \exists x_3 \cdots \exists x_n.\psi$. The only-if direction is (almost) immediate. Suppose that $\mathfrak{A} \models \varphi$, and let $\mathfrak{B}$ be the Cartesian product $\mathfrak{B} = \mathfrak{A} \cdot n$. Since φ is equality-free, $\mathfrak{B} \models \varphi$. Let R and S be, respectively, the sets of 1-types and 2-types realized in $\mathfrak{B}$, and let T be the set of n-types τ realized in $\mathfrak{B}$ and satisfying ψ (i.e. such that $\models \tau \rightarrow \psi$). It is routine to check that (R, S, T) is a certificate for φ. We illustrate with (**C3**). If $\rho \in R$, let $b_1 = \langle a_1, 0\rangle$ be such that $\mathrm{tp}^{\mathfrak{B}}[b] = \rho$. Now let $b_2 = \langle a_1, 1\rangle$, and pick $b_3, \ldots, b_n$ in any way such that the elements $\bar{b} = b_1, \ldots, b_n$ are distinct. Letting $\tau = \mathrm{tp}^{\mathfrak{B}}[\bar{b}]$, it is obvious that $\rho = \tau\langle x_1\rangle$ and $\tau = \tau\langle x_1, x_1, x_3, \ldots, x_n\rangle$. The other conditions are established similarly.

For the converse, we employ a *pairing function*, that is, a bijective function $\pi : {\mathbb{N}^+}^2 \rightarrow \mathbb{N}^+$ (where $\mathbb{N}^+$ denotes the positive integers). For definiteness, we may take π to be

$$\pi(x, y) = \frac{1}{2}(x + y - 2)(x + y - 1) + y.$$

Since π is bijective, for any natural number k, the equation $\pi(x, y) = k$ has a unique solution over $\mathbb{N}$; we write $\pi_1(k) = x$ and $\pi_2(k) = y$ to denote the components of this solution, and refer to π_i as the ith *projection function* ($i = 1, 2$). We define a sequence $\{\zeta_k\}_{k \geq 0}$, where ζ_k is a table of arity

$$a(k) = 1 + k(n - 2),$$

for all $k \geq 0$. (Thus, ζ_0 is a 1-table.) At every point k in the construction, we ensure that

$$\begin{array}{lll}
\zeta_k\langle x_1, \ldots, x_{a(k-1)}\rangle = \zeta_{k-1} & \text{if } k > 0; & (5.11)\\
\models \zeta_k \rightarrow \psi(x_{\pi_1(k)}, x_{\pi_2(k)}, x_{a(k-1)+1}, \ldots, x_{a(k)}) & \text{if } k > 0; & (5.12)\\
\zeta_k\langle x_i\rangle \in R & \text{for } 1 \leq i \leq a(k); & (5.13)\\
\zeta_k\langle x_i, x_j\rangle \in S & \text{for all } 1 \leq i, j \leq a(k). & (5.14)
\end{array}$$

The construction begins by choosing ζ_0 to be any element $\rho \in R$. Conditions (5.11)–(5.13) are then trivial, while (5.14) is secured by (**C3**) and (**C5**). Assuming that ζ_k has been defined satisfying (5.11)–(5.14), we define ζ_{k+1} also satisfying (5.11)–(5.14) (but with k replaced by $k + 1$). It greatly helps to think

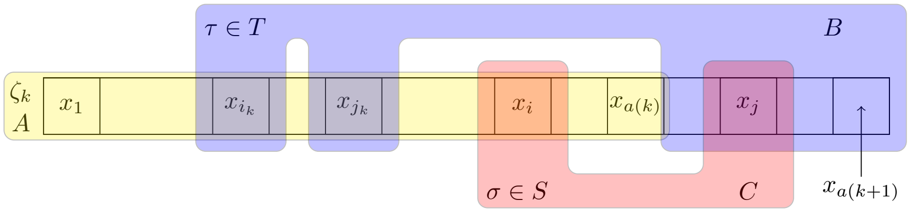


Figure 5.2: Construction of ζ_{k+1} from ζ_k (proof of Lemma 5.19).

of the ζ_k as structures, rather than as sets of formulas. The construction of ζ_{k+1} is illustrated in Fig. 5.2: the elements of the domain $\{x_1, \ldots, x_{a(k+1)}\}$ of ζ_{k+1} are depicted by the squares in the long rectangle, while various structures defined on (subsets of) this domain are indicated by the coloured regions with rounded corners. We first set $\zeta_{k+1} \restriction \{x_1, \ldots, x_{a(k)}\} := \zeta_k$ (see region A in Fig. 5.2), thus securing Condition (5.11) for $k + 1$. Recall now that, for $k \geq 0$, $i_k = \pi_1(k + 1)$ and $j_k = \pi_2(k + 1)$ are the components of the ordered pair assigned to $k + 1$ by the pairing function π, and observe that $1 \leq i_k, j_k \leq k+1 \leq a(k)$. Let $\rho_1 = \zeta_k\langle i_k\rangle$ and $\rho_2 = \zeta_k\langle j_k\rangle$; by (5.13), $\rho_1, \rho_2 \in R$. Assume first that $i_k \neq j_k$. By (**C1**) and (**C2**), there exists $\tau \in T$ such that $\tau\langle x_1\rangle = \rho_1 = \zeta_k\langle i_k\rangle$ and $\tau\langle x_2\rangle = \rho_2 = \zeta_k\langle j_k\rangle$, and of course τ satisfies ψ. We may then set $\zeta_{k+1} \restriction (x_{i_k}, x_{j_k}, x_{a(k)+1}, \ldots x_{a(k+1)}) = \tau$. (See region B in Fig. 5.2.) If, on the other hand, $i_k = j_k$, then, by (**C3**), there exists $\tau \in T$ such that $\rho_1 = \zeta_k\langle i_k\rangle = \tau\langle x_1\rangle$ and $\tau = \tau\langle x_1, x_1, x_3, \ldots, x_n\rangle$. Hence, there exists an $(n-1)$-type $\mu = \tau\langle x_1, x_3, \ldots, x_n\rangle$ such that $\mu\langle x_1\rangle = \zeta_k\langle i_k\rangle$ and $\models \mu(x_1, x_3, \ldots, x_n) \rightarrow \psi(x_1, x_1, x_3, \ldots, x_n)$. We may then set $\zeta_{k+1} \restriction (x_{i_k}, x_{a(k)+1}, \ldots x_{a(k+1)}) = \mu$. Either way, (5.12) holds for $k + 1$. That (5.13) also holds for $k + 1$ is guaranteed by inductive hypothesis and (**C4**).

It remains to secure (5.14). For each i ($1 \leq i \leq a(k)$), distinct from both i_k and j_k, we know from (5.13) that $\zeta_{k+1}\langle i\rangle = \zeta_k\langle i\rangle \in R$; and for each j ($a(k) + 1 \leq i \leq a(k + 1)$), we know from **(C4)** that $\zeta_{k+1}\langle j\rangle = \tau\langle j - a(k) + 1\rangle \in R$, whence from **(C1)** we can choose a 2-table $\sigma \in S$ and consistently set $\zeta_{k+1} \restriction \{x_i, x_j\}$ to be given by σ. (See region C in Fig. 5.2.) Finally, set any tuples in ζ_{k+1} that have not yet been defined arbitrarily. This completes the definition of ζ_k. To show that (5.14) holds for $k+1$, we consider four cases. (i) If $1 \leq i, j \leq a(k)$, the result is immediate by (5.11) and inductive hypothesis. (ii) If $a(k) + 1 \leq i, j \leq a(k + 1)$, then the result follows from (**C5**), bearing in mind that $\zeta_{k+1}\langle x_{a(k)+1}, \ldots, x_{a(k+1)}\rangle$ is $\tau\langle x_3, \ldots, x_n\rangle$ for some $\tau \in T$. (iii) If $1 \leq i \leq a(k) < j \leq a(k + 1)$, we have two sub-cases. If $i = i_k$ or $i = j_k$, then the result again follows from (**C5**), bearing in mind that $\zeta_{k+1}\langle x_{x_i, a(k)+1}, \ldots, x_{a(k+1)}\rangle$ is either $\tau\langle x_1, x_3, \ldots, x_n\rangle$ or $\tau\langle x_2, x_3, \ldots, x_n\rangle$ for some $\tau \in T$. If $i \neq i_k$ and $i \neq j_k$, then we have just defined ζ_{k+1} so that $\zeta\langle x_i, x_j\rangle \in S$. This completes the third case. (iv) If $1 \leq j \leq a(k) < i \leq a(k + 1)$, we simply proceed as in Case (iii), but with i and j transposed. This completes the inductive step and thus the construction of the sequence $\{\zeta_k\}_{k\geq 0}$.

From (5.11), bearing in mind that each $a(k)$-table ζ_k may be regarded as a structure over $\{x_1, \ldots, a_{a(k)}\}$, we may define the structure $\mathfrak{X}$, over the domain $X = \{x_1, x_2, x_3, \ldots\}$, to be the union of all the ζ_k. From (5.12), taking any objects x_i, and x_j in X, and writing $k = \pi(i, j)$, we see that $\mathfrak{X} \models \psi[x_i, x_j, x_{a(k-1)+1}, \ldots, x_{a(k)}]$. That is to say, $\mathfrak{X} \models \varphi$, as required. □

Lemma 5.19 shows that the satisfiability problem for constant-free formulas of $[\forall^2\exists^*]$ is decidable, since a certificate over a finite signature Σ is clearly finite. Note however that the number of atoms featuring the variables $x_1, \ldots, x_k$ over Σ is $o(|\Sigma| \cdot k^r)$, where r is the maximum arity of predicates in Σ, whence the number of k-types is $o(2^{|\Sigma| \cdot k^r})$, which, for $k > 1$, grows doubly exponentially if r is unbounded, thus yielding an upper complexity bound of at best 2-NExpTime. The following simple idea shows how this may be improved to obtain a tight bound. Let φ and Σ be as above. Say that an atom β over Σ is *active for* φ if either: (i) β is an atom over Σ featuring just one variable; (ii) β can be written as $\beta'(x_i, x_j/x_{i'}, x_{j'})$, where β' occurs in φ and features the two distinct variables x_i and x_j; or (iii) β occurs in φ. (Here, $\beta'(x_i, x_j/x_{i'}, x_{j'})$ denotes the result of simultaneously substituting the variables $x_{i'}$ and $x_{j'}$ for x_i and x_j in β'.) The number of atoms falling under condition (i) is $|\Sigma|$, the number of atoms falling under condition (ii) is bounded above by $(\|\varphi\| \cdot n^2) \leq \|\varphi\|^3$, where n is the number of variables in φ; and the number of atoms falling under condition (iii) is bounded above by $\|\varphi\|$. If ζ is a k-type, then define ζ^* to be the result of making all non-active atoms occurring in ζ false. Thus, if ρ is a 1-type, then $\rho^* = \rho$, and, moreover, for any $k \geq 1$, the number of k-types of the form ζ^* is $O(2^{\|\varphi\|^3})$. Finally, if Z is a set of k-types, write $Z^* = \{\zeta^* \mid \zeta \in Z\}$. It is then a simple matter to check that, if (R, S, T) is a certificate for φ, then so is (R, S^*, T^*).

Theorem 5.20. *The satisfiability problem for* $[\exists^*\forall^2\exists^*]$ *is* NExpTime*-complete.*

Proof. For the upper bound, let $\varphi \in [\exists^*\forall^2\exists^*]$ be given. By Lemma 5.18, we may assume that φ is of the form $\forall x_1 \forall x_2 \exists x_3 \cdots \exists x_n.\psi$, over a purely relational signature Σ. To test the satisfiability of φ, guess a set of 1-types R, a set of 2-types S^* (over Σ) and a set of n-types T^* (over Σ) such that all atoms that are non-active for φ are assigned to be false in S^* and T^*; and check that (R, S^*, T^*) is a certificate for φ.

The lower bound is established by the proof of Lemma 3.13, since the formulas constructed there are in $[\forall^2\exists^*]$. □

We are not quite finished with the fragment $[\exists^*\forall^2\exists^*]$. The structures built in the proof of Lemma 5.19 are infinite, and thus do not straightforwardly yield any information about the finite satisfiability problem. The following beautiful proof rectifies this matter. It constitutes an alternative proof of (the hard direction of) Lemma 5.19.

Theorem 5.21. *The fragment* $[\exists^*\forall^2\exists^*]$ *has the finite model property.*

Proof. By Lemma 5.18, we need consider only the sub-fragment of $[\forall^2\exists^*]$ in which individual constants do not appear.

Let $\varphi := \forall x_1 \forall x_2 \exists x_3 \cdots \exists x_n . \psi$ be a satisfiable formula of $[\forall^2\exists^*]$ over a purely relational signature. From Lemma 5.19 (easy direction), let (R, S, T) be a certificate for φ. We proceed to build a finite model of φ. Fixing $\ell > 0$, and, letting $m = |R|$, define A_ℓ to be the union of ℓ pairwise disjoint sets $B_1, \ldots, B_\ell$, each of cardinality m. Write $R = \{\rho_1, \ldots, \rho_m\}$ and $B_i = \{a_{i,1}, \ldots, a_{i,m}\}$ for each i $(1 \leq i \leq \ell)$, and define a structure $\mathfrak{A}_\ell$ on A_ℓ in three steps as follows:

1. for each i $(1 \leq i \leq \ell)$ and each j $(1 \leq j \leq m)$, set $\text{tp}^{\mathfrak{A}_\ell}[a_{i,j}] = \rho_j$;

2. for each pair of distinct elements $\langle a, b \rangle \in A^{(n)}$, set $\text{tp}^{\mathfrak{A}_\ell}[a, b] = \sigma$, where σ is chosen at random (with a flat distribution) from those elements of S such that $\sigma\langle x_1 \rangle = \text{tp}^{\mathfrak{A}_\ell}[a]$ and $\sigma\langle x_2 \rangle = \text{tp}^{\mathfrak{A}_\ell}[b]$;

3. for each k-tuple $\bar{a}$ of elements $(3 \leq k \leq r)$ containing at least three distinct elements, and every predicate p of arity k, set either $\mathfrak{A}_\ell \models p[\bar{a}]$ or $\mathfrak{A}_\ell \models \neg p[\bar{a}]$ at random (each with probability 1/2).

Note that the selection of σ is always possible in Step 2, by virtue of (**C1**). Thus, $\mathfrak{A}_\ell$ is actually a random variable: in particular, there exists, for each $\ell > 0$, a particular probability that $\mathfrak{A}_\ell \models \varphi$. It suffices to show that, for ℓ sufficiently large, this probability is non-zero, for in that case, φ certainly has at least one finite model.

Let $a_1, \ldots, a_n$ be an n-tuple in A_ℓ. If $a_1 \neq a_2$, then, since $\text{tp}^{\mathfrak{A}_\ell}[a_1, a_2] \in S$, by (**C2**), there exists a $\tau \in T$ such that $\tau\langle x_1, x_2 \rangle = \sigma$. Letting $d = \binom{n-2}{2}$ be the number of ways of choosing 2 objects from $n - 2$, and e the number of atoms featuring a predicate from Σ in which at least three distinct variables occur, we see that the probability that $\mathfrak{A}_\ell \models \psi[a_1, \ldots, a_n]$ for a particular choice of $a_3, \ldots, a_n$ is at least $\epsilon = (1/|S|)^{d+2(n-2)} \cdot (1/2)^e$. If $a_1 = a_2$, we proceed similarly, using (**C3**) and the fact that $\text{tp}^{\mathfrak{A}_\ell}[a_1] \in R$, obtaining the (larger) lower bound of $(1/|S|)^{d+n-2} \cdot (1/2)^e$ on the probability that $\mathfrak{A}_\ell \models \psi[a_1, \ldots, a_n]$ for a particular choice of $a_3, \ldots, a_n$.

Now fix $a_1, a_2 \in A_\ell$. We bound from below the probability that there exist $a_3, \ldots, a_n \in A_\ell$ such that $\mathfrak{A}_\ell \models \psi[a_1, \ldots, a_n]$. Let $m = \lfloor (\ell - 2)/(n - 2) \rfloor$, so that $m(n - 2) \leq \ell - 2$. Thus, we may choose τ for a_1, a_2 as in the previous paragraph, select m different $(n - 2)$-tuples $\bar{b}_i = b_{i,3}, \ldots b_{i,n}$, with all elements distinct and distinct from a_1, a_2, and such that $\text{tp}^{\mathfrak{A}_\ell}[b_{i,j}] = \tau\langle x_j \rangle$. The chance that none of these is a witness for ψ with respect to a_1, a_2 is at most $(1 - \epsilon)^m \leq (1 - \epsilon)^{(\ell-2)/(n-2)}$.

Thus the probability that $\mathfrak{A}_\ell \not\models \varphi$ is at most the sum of the probability for each of the $(|R| \cdot \ell)^2$ pairs a_1, a_2, i.e.

$$(|R| \cdot \ell)^2 \cdot \left(1 - \left(\frac{1}{|S|} \right)^{\binom{n-2}{2} + 2(n-2)} \cdot \left(\frac{1}{2} \right)^e \right)^{\frac{\ell - 2}{n-2}}.$$

Now just take ℓ large enough that this quantity is less than unity. □

5.4.2 Two universal quantifiers with equality

This section is devoted to a proof of the following theorem.

Theorem 5.22. *The problem* Sat($[\forall\forall\exists^*]_=$) *is undecidable.*

The proof exploits the ability of $[\forall\forall\exists^*]_=$ to re-create the structure of the natural numbers, as expressed in the following lemma. Once again, we employ the Cantor pairing function $\pi : \mathbb{N}^2 \rightarrow \mathbb{N}$, with projection functions π_1 and π_2 (see Lemma 5.19).

Lemma 5.23. *There exists a $[\forall\forall\exists^*]_=$-sentence $\varphi_{\mathbb{N}}$ over a relational signature containing* (*among other predicates*) *the unary predicate Z and binary predicates S, P_1, P_2, and having the following properties.* (i) *The sentence $\varphi_{\mathbb{N}}$ has a model over the domain $\mathbb{N}$ in which Z is interpreted as the singleton $\{0\}$, S as the successor relation $\{\langle n, n+1 \rangle \mid n \in \mathbb{N}\}$ and P_i as the projection relation $\{\langle n, \pi_i(n) \rangle \mid n \in \mathbb{N}\}$ for $i = 1, 2$.* (ii) *For any model $\mathfrak{A} \models \varphi_{\mathbb{N}}$, the domain A contains an infinite subset, denoted $\{\mathbf{0}, \mathbf{1}, \mathbf{2}, \ldots\}$ such that:*

(*A*) *$\mathbf{0}$ is the unique element c such that $\mathfrak{A} \models Z[c]$, and for all $a \in A$, $\mathfrak{A} \not\models S[a, \mathbf{0}]$;*

(*B*) *for all $n > 0$, $\mathbf{n-1}$ is the unique element a of A such that $\mathfrak{A} \models S[a, \mathbf{n}]$;*

(*C*) *for all $n > 0$, $\mathbf{n}$ is the unique element a of A such that $\mathfrak{A} \models S[\mathbf{n-1}, a]$;*

(*D*) *for $i = 1, 2$ all $n \geq 0$ and all $a, b \in A$, if $\mathfrak{A} \models P_i[a, \mathbf{n}]$ and $\mathfrak{A} \models P_i[a, b]$, then $b = \mathbf{n}$.*

We first define the $[\forall\forall\exists^*]_=$-sentence $\varphi_{\mathbb{N}}$, together with a model $\mathfrak{N}$ over the domain of natural numbers, $\mathbb{N}$. Statement (i) of the lemma requires that we read $Z(x)$ as "x is zero", read $S(x, y)$ as "y is the successor of x", and read $P_i(x, y)$ as "y is the ith component of the pair encoded by x ($i = 1, 2$). The sentence features the additional binary predicates N, Q, R_1 and R_2. As an aide to intuition, read $N(x, y)$ as "y encodes the pair obtained by incrementing the first component of the pair encoded by x, read $Q(x, y)$ as "y is the successor of the second component of the pair encoded by x", and read $R_i(x, y)$ as "y encodes *any* pair whose ith component is the successor of the second component of the pair encoded by x ($i = 1, 2$)". For ease of reading, we do not write $\varphi_{\mathbb{N}}$ in prenex form; however, it will be obvious that $\varphi_{\mathbb{N}}$ is logically equivalent to a $[\forall\forall\exists^*]_=$-sentence.

Let $\mathfrak{N}$ be the interpretation of the signature of $\varphi_{\mathbb{N}}$ over $\mathbb{N}$ obtained by setting

$$\begin{aligned}
&Z^{\mathfrak{N}} = \{0\}\\
&S^{\mathfrak{N}} = \{\langle n, n+1\rangle \mid n \in \mathbb{N}\}\\
&P_i^{\mathfrak{N}} = \{\langle n, \pi_i(n)\rangle \mid n \in \mathbb{N}\} \qquad i = 1, 2\\
&N^{\mathfrak{N}} = \{\langle \pi(m,n), \pi(m+1,n)\rangle \mid m, n \in \mathbb{N}\}\\
&Q^{\mathfrak{N}} = \{\langle n, \pi_2(n)+1\rangle \mid n \in \mathbb{N}\}\\
&R_i^{\mathfrak{N}} = \{\langle m, n\rangle \mid m, n, \in \mathbb{N}, \pi_2(n) = \pi_i(m)+1\} \qquad i = 1, 2
\end{aligned}$$

as just suggested. And let $\varphi_{\mathbb{N}} = \forall x \forall y \exists z_0 . \psi_{\mathbb{N}}$, where $\psi_{\mathbb{N}}$ is the conjunction of the formulas:

$$Z(x) \wedge Z(y) \to x = y \tag{5.15}$$

$$Z(z_0) \wedge \neg S(x, z_0) \wedge \bigwedge_{i=1}^{2}((P_i(x, z_0) \wedge P_i(x, y)) \to y = z_0) \tag{5.16}$$

$$\exists z . S(x, z) \tag{5.17}$$

$$\neg Z(x) \wedge x \neq y \to \exists w(S(w, x) \wedge \neg S(w, y)) \tag{5.18}$$

$$S(x, y) \to \exists z(Q(z, y) \wedge P_2(z, x) \wedge P_1(z, z_0)) \tag{5.19}$$

$$\exists z(N(x, z) \wedge (Q(x, y) \to Q(z, y)) \wedge \bigwedge_{i=1}^{2}(R_i(x, y) \to R_i(z, y))) \tag{5.20}$$

$$N(x, y) \to \exists z(P_2(x, z) \wedge P_2(y, z)) \tag{5.21}$$

$$N(x, y) \to \exists w \exists u(P_1(x, w) \wedge S(w, u) \wedge P_1(y, u)) \tag{5.22}$$

$$Q(x, y) \to \exists z(P_1(x, z) \wedge (S(z, y) \to P_2(x, z))) \tag{5.23}$$

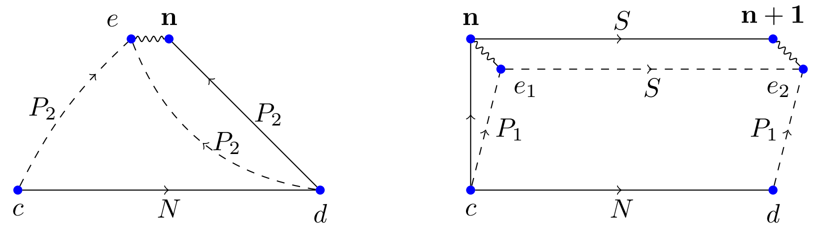
$$\begin{aligned}
&\bigwedge_{i=1}^{2}[P_i(x, y) \wedge \neg Z(y) \to\\
&\qquad \exists z \exists w(R_i(z, x) \wedge P_2(z, w) \wedge P_1(z, z_0)) \wedge S(w, y)]
\end{aligned} \tag{5.24}$$

$$\bigwedge_{i=1}^{2}[R_i(x, y) \to \exists z \exists w(P_1(x, z) \wedge S(z, w) \wedge (P_i(y, w)) \to P_2(x, z))]. \tag{5.25}$$

A simple check shows:

Lemma 5.24. $\mathfrak{N} \models \varphi_{\mathbb{N}}$.

Lemma 5.24 establishes statement (i) of Lemma 5.23. To establish statement (ii), suppose $\mathfrak{A} \models \varphi_{\mathbb{N}}$. Since $\mathfrak{A}$ will be fixed for the present, we omit the symbols "$\mathfrak{A} \models$" when making statements about satisfaction of formulas by elements of $\mathfrak{A}$, writing, for example $N[c, d]$ instead of $\mathfrak{A} \models N[c, d]$. We define the set $\{\mathbf{n} \mid n \in \mathbb{N}\} \subseteq A$ by induction on n, establishing properties (A)–(D) as we go. For the base case, the conjuncts (5.15) and (5.16) of ψ ensure that there is a unique element satisfying Z; define $\mathbf{0}$ to be this element. Property (A) is then



(a) Backward.

(b) Forward.

Figure 5.3: Steps in backward and forward N-chain propagation.

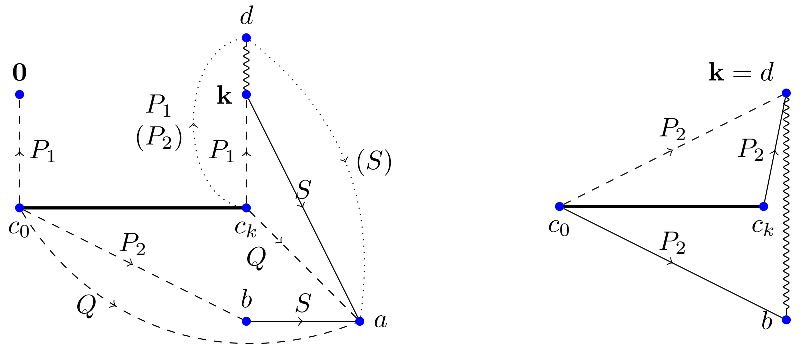
immediate. For $n = 0$, properties (B) and (C) are vacuous, while property (D) follows from (5.16).

Let us assume, then, that $\mathbf{0}, \dots \mathbf{k}$ have been defined ($k \geq 0$), with properties (A)–(D) holding for all n in the range $[0, k]$. A key element in the inductive step is a pair of observations concerning the predicate N. Suppose c, d are elements of A such that $N[c, d]$ and $0 \leq n \leq k$ such that $P_2[d, \mathbf{n}]$. We claim that $P_2[c, \mathbf{n}]$. Indeed, by conjunct (5.21), there exists e such that $N[c, e]$ and $P_2[d, e]$; hence by property (D), $d = \mathbf{n}$, establishing the claim. The situation is illustrated in Fig. 5.3(a): dashed lines indicate the relations arising from (5.21), and the zig-zag line the identification forced by property (D). Likewise, suppose c, d are elements of A and $0 \leq n < k$ such that $N[c, d]$ and $P_1[c, \mathbf{n}]$. We claim that $P_1[d, \mathbf{n+1}]$. Indeed, by conjunct (5.22), there exist e_1, e_2 such that $P_1[c, e_1]$, $S[e_1, e_2]$ and $P_1[d, e_2]$; hence by property (D), $e_1 = \mathbf{n}$, whence by property (C), $e_2 = \mathbf{n+1}$, establishing the claim. The situation is illustrated in Fig. 5.3(b): again, dashed lines indicate the relations arising from (5.22), and the zig-zag line the identifications (first e_1 with $\mathbf{n}$, and then e_2 with $\mathbf{n+1}$) forced by properties (D) and (C). Call an *N-chain* any sequence $c_0, \dots c_n$ ($n \leq k$) such that $N(c_i, c_{i+1})$ for all i ($0 \leq i < n$). Applying the above claims repeatedly, we see that, for any such N-chain, and any n ($0 \leq n \leq k$), $P_2(c_m, \mathbf{n})$ entails $P_2(c_0, \mathbf{n})$, and $P_1(c_0, \mathbf{0})$ entails $P_1(c_m, \mathbf{m})$. We refer to these entailments in the sequel as *backward chain propagation* and *forward chain propagation*, respectively.

Armed with backward and forward N-chain propagation, we can complete the inductive step with a series of simple lemmas.

Lemma 5.25. *For all $a, b \in A$, if $S[\mathbf{k}, a]$ and $S[b, a]$, then $b = \mathbf{k}$.*

Proof. The construction is illustrated in Fig. 5.4, where two separate drawings have been given to reduce visual clutter. By conjunct (5.19), there exists $c_0 \in A$ such that $Q[c_0, a]$, $P_2[c_0, b]$ and $P_1[c_0, \mathbf{0}]$. By conjunct (5.20), we may construct an N-chain $c_0, \dots, c_k$ all elements of which are related to a by Q. By forward N-chain propagation, $P_1[c, \mathbf{k}]$. By (5.23), there exists $d \in A$ such that $P_1[c_k, d]$ and either $\neg S[d, a]$ or $P_2[c_k, d]$. These steps are illustrated in Fig. 5.4(a), where solid arrows indicate already-established relations, dashed/dotted arrows indicate newly-inferred relations, zigzag lines indicate forced identities, and labels



(a) First stage.

(b) Second stage.

Figure 5.4: Proof of Lemma 5.25.

on arrows in parentheses indicate relations that hold conditionally. By property (D), $d = \mathbf{k}$. But since $S[\mathbf{k}, a]$, we have $P_2[c_k, \mathbf{k}]$. The final step is illustrated in Fig. 5.4(b) (under the same diagrammatic semantics). By backward N-chain propagation, $P_2[c_0, \mathbf{k}]$. But since $P_2[c_0, b]$, property (D) entails $b = \mathbf{k}$, as required. □

Lemma 5.26. *There is a unique $a \in A$ such that $S[\mathbf{k}, a]$.*

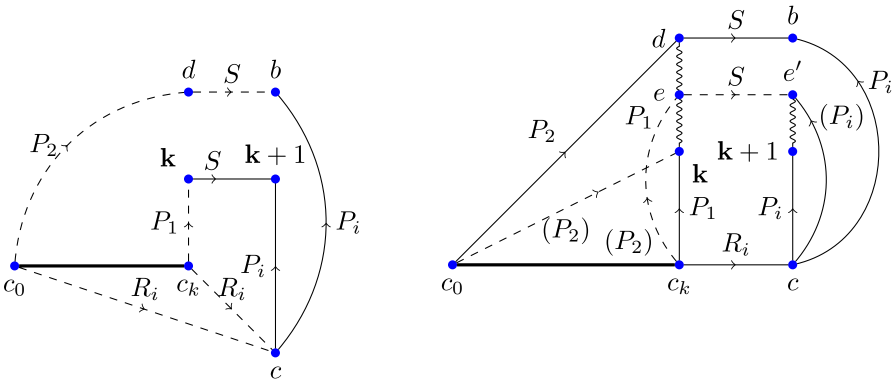
Proof. There exists $a \in A$ such that $S[\mathbf{k}, a]$ by conjunct (5.17). To show that a is unique, suppose $b \in A \setminus \{a\}$. We have already established (Property A) that $\mathbf{0}$ is the unique element satisfying Z and that $\mathbf{0}$ is not the successor of anything. Thus, $\neg Z[a]$. By conjunct (5.18), there exists c such that $S[c, a]$ and $\neg S[c, b]$. By Lemma 5.25, $c = \mathbf{k}$. Thus, $\neg S[b, \mathbf{k}]$. □

Define $\mathbf{n+1}$ to be the unique a guaranteed by Lemma 5.26. It follows from Lemmas 5.25 and 5.26 that properties (B) and (C) are maintained for the value $k + 1$. We need only check property (D).

Lemma 5.27. *For all $b, c \in A$ and all i $(1 \leq i \leq 2)$, if $P_i[c, \mathbf{k+1}]$ and $P_i[c, b]$, then $b = \mathbf{k+1}$.*

Proof. We first claim that $\neg Z[b]$. For by (A), $Z[b]$ implies $b = \mathbf{0}$, and by (D), $\mathbf{k+1} = \mathbf{0}$. But this is impossible since $S[\mathbf{k}, \mathbf{k+1}]$. This having been established, the construction is illustrated in Fig. 5.5, where, again, two separate drawings have been given to reduce visual clutter, using the same diagrammatic semantics as in Fig. 5.4.

By conjunct (5.24), there exist c_0, d such that $R_i[c_0, c]$, $P_2[c_0, d]$, $P_1[c_0, \mathbf{0}]$ and $S[d, b]$. By conjunct (5.20), we may construct an N-chain $c_0, \ldots, c_k$ all elements of which are related to c by R_i. Since $P_1[c_0, \mathbf{0}]$, we have by forward N-chain



(a) First stage.

(b) Second stage.

Figure 5.5: Proof of Lemma 5.27.

propagation $P_1[c_k, \mathbf{k}]$. Recall that $\mathbf{k+1}$ has already been defined, and that $S[\mathbf{k}, \mathbf{k+1}]$. The situation is illustrated in Fig. 5.5(a).

By conjunct (5.25), there exist e, e' such that $P_1[c_k, e]$, $S[e, e']$ and either $\neg P_i[c, e']$ or $P_2[c_k, e]$. By property (D), $e = \mathbf{k}$, whence $e' = \mathbf{k+1}$, by definition of $\mathbf{k+1}$. But by assumption, $P_i[c, \mathbf{k+1}]$, i.e., $P_i[c, e']$, whence $P_2[c_2, \mathbf{k}]$. Using backward N-chain propagation, $P_2[c_0, \mathbf{k}]$. By property (D), $d = \mathbf{k}$, and hence $S[\mathbf{k}, b]$. By Lemma 5.26, $b = \mathbf{k+1}$. See Fig. 5.5(b). □

This completes the induction, and hence the proof of Lemma 5.23. Theorem 5.22 is now easy.

Proof of Theorem 5.22. We reduce the problem $\text{Sat}([\forall\exists\forall])$ to $\text{Sat}([\forall\forall\exists^*]_=)$. The result then follows from Theorem 5.2.

Let $\varphi = \forall x \exists y \forall z . \psi(x, y, z)$ be a given formula of the quantifier prefix fragment $[\forall\exists\forall]$, where ψ is quantifier-free. Let $\varphi_{\mathbb{N}}$ be the formula of Lemma 5.23, and $\mathfrak{N}$ the standard model of $\varphi_{\mathbb{N}}$ mentioned in Lemma 5.24. We may assume without loss of generality that the signatures of φ and $\varphi_{\mathbb{N}}$ are disjoint. Now let φ' be the formula $\varphi_{\mathbb{N}} \wedge \forall x \forall z \exists y (S(x, y) \wedge \psi(x, y, z)))$. (Note the changed order of quantifiers.) By renaming existentially quantified variables and performing routine logical manipulations, we can easily obtain an $[\forall\forall\exists^*]_=$-sentence logically equivalent to φ'. It therefore suffices to show that φ is satisfiable if and only if φ' is.

Suppose, then, φ is satisfiable. Then the Skolemized form $\forall x \forall z . \psi(x, s(x), z)$ of φ is satisfiable. Adding an individual constant 0 to the signature, it evident that φ is satisfied in some structure $\mathfrak{A}$ with domain $\{s^{(n)}(0) \mid n \in \mathbb{N}\}$, where $s^{(n)}$ denotes the ground term formed by applying s to 0 n-times. Identifying $s^{(n)}$ with the integer n, and interpreting the signature of $\varphi_{\mathbb{N}}$ according to the structure $\mathfrak{N}$, we see that $\mathfrak{A} \models \varphi'$. Conversely, suppose $\mathfrak{A} \models \varphi'$. Then $\mathfrak{A} \models \varphi_{\mathbb{N}}$, so let $N = \{\mathbf{0}, \mathbf{1}, \ldots\}$ be the infinite subset of A guaranteed by Lemma 5.23. Let $\mathfrak{B} = \mathfrak{A} \restriction N$. Thus, $\mathfrak{B} \models \forall x \forall z \exists y (S(x, y) \wedge \psi(x, y, z)))$, and, moreover, for every pair of elements a, b from B, there exists at most one element c such that $\mathfrak{B} \models S[a, c]$, and this depends only on a. Therefore, $\mathfrak{B} \models \forall x \exists y \forall z . \psi(x, y, z)$ as required. □

The proof of Theorem 5.22 uses a specific number of trailing existential quantifiers (actually, thirteen). Can we do any better in this regard? The answer is yes, though we do not give the proof here, which is rather technical. (See the Bibliographic Notes for details.) We simply state without proof:

Proposition 5.28. *The problem* Sat$([\forall\forall\exists]_=)$ *is undecidable.*

Concluding remarks

In this chapter, we have considered quantifier prefix fragments defined by means of standard prefix specifiers over signatures containing any number of relational symbols and individual constants. In all cases, we have determined the decidability of the satisfiability problem, and, where that problem is decidable, tight complexity bounds (modulo a small complexity gap relating to the fragments $[\exists^*\forall\exists^\alpha]_=$, for finite $\alpha \geq 2$). Two of the results reported in this chapter tower above the others: Theorem 5.20, which establishes the decidability of the satisfiability problem for the fragment $[\exists^*\forall\forall\exists^*]$, and Theorem 5.22, which establishes the undecidability of the corresponding problem for $[\exists^*\forall\forall\exists^*]_=$. But even the lesser results mentioned here are important because of the techniques that they employ—techniques which recur throughout this book.

We remark that all the fragments considered in this chapter are either undecidable for both satisfiability and finite satisfiability (the fragments $[\forall\forall\forall\exists]$, $[\forall\exists\forall]$, $[\forall\forall\exists]_=$ and their super-fragments) or have the finite model property (the fragments $[\exists^*\forall^*]_=$, $[\exists^*\forall\exists^*]_=$, $[\exists^*\forall\forall\exists^*]$ and their sub-fragments). Bearing in mind the results of Ch. 3 and Ch. 4, the reader might be forgiven for conjecturing that *all* naturally defined fragments of first-order logic having decidable satisfiability problems enjoy the finite model property. In Part II of this book, we shall see that this is not the case.

Exercises

1. Which quantifier prefix fragments $\mathcal{F}$ are closed under conjunction, in the sense that, if $\varphi_1, \varphi_2 \in \mathcal{F}$, there exists a $\varphi \in \mathcal{F}$ such that φ is logically equivalent to $\varphi_1 \wedge \varphi_2$?

2. Complete the details in the proof of the undecidability of FinSat$([\forall^3\exists])$ sketched in Theorem 5.3.

3. Show that Sat($[\exists\forall^*]$) is NPTime-complete if no individual constants (or function symbols) are allowed.

4. Let C, D be clauses in the uniform 1-variable clausal fragment, such that C and D d-resolve to form a clause E. It is tempting to suppose that $d(E) \leq d(C) + d(D)$. Show that this is not necessarily so.

5. Strengthen the proof of Lemma 5.8 so as to obviate the need for an individual constant.

Bibliographic notes

The quest for an algorithm to decide validity (satisfiability) of formulas in first-order logic dates back to D. Hilbert and W. Ackermann in their seminal book *Grundlagen der mathematischen Logic* [38, 39], where it was referred to as the *Entscheidungsproblem* (Decision Problem). Although the semantics of first-order logic was not at that time formalized in the style now familiar in logic textbooks, it is nevertheless clear from texts written at the time that the *Entscheidungsproblem* was understood, in all essentials, in the form it is today. When the first edition of the *Grundlagen* (1928) was written, no clear definition of the notion of an algorithmic procedure was available, and Hilbert and Ackermann mention simply mention some known decidable fragments—in particular the monadic fragment and the fragment $[\exists^*\forall^*]$—as special, solved cases. By contrast, the second (1938) edition was able to mention not only the positive solution of the *Entscheidungsproblem* for the fragments $[\exists^*\forall\exists^*]$ and $[\exists^*\forall^2\exists^*]$, but also the negative result of A. Church [17] concerning the general case. Hilbert seems to concede that there might be no general solution to the problem as he (presumably) originally envisaged. The paper by A. Turing [76] reconstructing the notion of algorithmic solvability in terms of (what we now call) Turing machines is not mentioned in this edition. We remarked above that Turing himself observed (*op. cit.*, p. 263) that the satisfiability problem for the quantifier prefix fragment $[\forall\exists\forall\exists^5]$ is undecidable.

A complete classification of the quantifier prefixes for which the satisfiability problem is decidable can be found in E. Börger, E. Grädel and Y. Gurevich [15], which gives the decidability and computational complexity of all fragments defined by (i) the allowed quantifier prefixes, (ii) the maximal numbers of predicates of all arities and function-symbols (including individual constants) of all arities, and (iii) the presence or absence of the equality predicate. We owe our brisk, one-chapter treatment here to the decision not to consider restrictions on the numbers of available predicates (of various arities), or signatures with function symbols. Earlier surveys of the state-of-the-art on the satisfiability problem for quantifier-prefix fragments—now essentially only of historical interest—are those of W. Ackermann [2] and B. Dreben and W. Goldfarb [20].

The decidability of the fragment $[\exists^*\forall^*]_=$ was first observed by P. Bernays and M. Schönfinkel [14], generalizing earlier results of H. Behmann [11]. This fragment was also studied by F.P. Ramsey [67]; however, the object of Ramsey's interest was not the (essentially trivial) satisfiability problem, but rather the *spectrum* problem for this fragment. The *spectrum* of a formula is the set of natural numbers which are the cardinalities of its finite models. Ramsey showed that the spectra of formulas of the fragment $[\exists^*\forall^*]_=$ are exactly the finite and cofinite subsets of the positive integers; he proved his celebrated combinatorial theorem on classification of sequences of k-tuples taken from infinite sets in the course of this argument. The decidability of $\mathrm{Sat}([\exists^*\forall\exists^*])$ was also established early on, by W. Ackermann [1]. We gave a very different proof to Ackermann's in our Lemma 5.7, using a resolution-based approach originally due to W. Joyner; for details, see the Bibliographic Notes to Ch. 3. The complexity upper-bound of ExpTime which we thereby obtained in fact extends to the fragment in which arbitrary function-symbols are permitted, as shown by E. Grädel [29]; but this latter result requires a much more difficult argument. The complexity of the problem $\mathrm{Sat}([\exists^*\forall\exists^*]_=)$ (i.e. the same fragment, but with equality) is due to P. Kolaitis and M. Vardi [49]. The decidability of the satisfiability for the fragment $\mathrm{Sat}([\exists^*\forall^2\exists^*])$ was established independently by K. Gödel [27], L. Kalmár [45] and K. Schütte [69]. Gödel obtains an exponential-size bound models, using a difficult argument; the much more perspicuous proof in our Lemma 5.19 is taken from Kalmár *op. cit.* The probabilistic proof of our Lemma 5.21 is copied (more or less directly) from Y. Gurevich and S. Shelah [36]. That proof actually yields a doubly-exponential bound on model-sizes; however, this does not affect the complexity, which is already established by Theorem 5.20, whose proof we have simply taken from [15, Ch. 6]. In the final sentence of his paper, Gödel remarked that his decidability result holds without essential change if equality is admitted, a remark eventually shown to be false by W. Goldfarb [28], who proved the undecidability of $\mathrm{Sat}[\forall^2\exists^*]_=$. Our proof of Theorem 5.22 closely follows Goldfarb's presentation. A detailed proof of Proposition 5.28, which strengthens this result to show the undecidability of $\mathrm{Sat}[\forall^2\exists]_=$ can again be found in [15, Ch. 6].

Undecidability results based on tilings, of the kind encountered Section 5.2, can be routinely developed from the original results on the undecidability of $\mathrm{Sat}(\mathcal{FO})$ and $\mathrm{FinSat}(\mathcal{FO})$ by Turing [76] and Trakhtenbrot [75]. The undecidability of $\mathrm{Sat}([\forall^3\exists])$ and $\mathrm{Sat}([\exists\forall\exists])$ reported here may be attributed to J. Surányi [74] and H. Wang [78], respectively. For more information on this technique, see the Bibliographic Notes to Ch. 3.

Bibliography

[1] W. Ackermann. Über die Erfüllbarkeit gewisser zählaudrücke. *Mathematische Annalen*, 100:638–649, 1928.

[2] W. Ackermann. *Solvable Cases of the Decision Problem.* Noth-Holland, 1954.

[3] H. Andréka, J. van Benthem, and I. Németi. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27:217–274, 1998.

[4] Aristotle. *Prior Analytics.* Hackett, Indianapolis, IN, 1989. (R. Smith, Tr.).

[5] A. Arnauld. *Logic, or, the Art of Thinking ("The Port-Royal Logic").* Bobbs-Merrill, 1964. tr. J. Dickoff and P. James (first published 1662).

[6] F. Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider. *The Description Logic Handbook: theory, implementation, and applications.* Cambridge University Press, 2nd edition, 2010.

[7] F. Baader, I. Horrocks, C. Lutz, and U Sattler. *An Introduction to Description Logic.* Cambridge University Press, 2017.

[8] F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.

[9] L. Bachmair and H. Ganzinger. On restrictions of ordered paramodulation with simplification. In *Proceedings. 10th International Conference on Automated Deduction*, volume 449 of *Lecture Notes in Artificial Intelligence*, pages 427–441, Berlin, 1990. Springer.

[10] L. Bachmair and H. Ganzinger. Resolution theorem proving. In *Handbook of Automated Reasoning*, volume 1, chapter 2, pages 19–99. North Holland, 2001.

[11] H. Behmann. Beiträge zur Algebra der Logik, insbesondere zum Entscheidungsproblem,. *Mathematische Annalen*, 86:163–229, 1922.

[12] R. Berger. The undecidability of the domino problem. *Memoirs of the American Mathematical Society*, 66:??, 1966.

[13] R. Berger. Complexity results for classes of quantificational formulas. *Journal of Computer and System Sciences*, 21(3):317–353, 1980.

[14] P. Bernays and M. Schönfinkel. Zum Entscheidungsproblem der mathematischen Logik. *Mathematische Annalen*, 99:342–372, 1928.

[15] E. Börger, E. Grädel, and Y. Gurevich. *The classical decision problem.* Springer, Berlin, 1997.

[16] C. Chang and H. Keisler. *Model Theory.* North-Holland, Amsterdam, 3rd edition, 1990.

[17] Alonzo Church. A note on the entscheidungsproblem. *J. Symb. Log.*, 1(1):40–41, 1936.

[18] J. Corcoran. Completeness of an ancient logic. *Journal of Symbolic Logic*, 37(4):696–702, 1972.

[19] N. Cutland. *Computability.* Cambridge University Press, Cambridge.

[20] B. Dreben and W. Goldfarb. *Solvable Classes of Quantificational Formulas.* Addison-Wesley, 1979.

[21] C. Fermüller, U. Hustadt, A. Leitsch, and T. Tammet. Resolution decision procedures. In *Handbook of Automated Reasoning*, volume 2, chapter 25, pages 1792–1849. North Holland, 2001.

[22] C. Fermüller, A. Leitsch, T. Tammet, and N. Zamov. *Resolution Methods for the Decision Problem*, volume 679 of *Lecture Notes in Artificial Intelligence*. Springer, Berlin, 1993.

[23] Robert J. Fogelin. Hamilton's Quantification of the Predicate. *The Philosophical Quarterly*, 26(104):217–228, 1976.

[24] M. Fürer. The computational complexity of the unconstrained limited domino problem (with implications for logical decision problems). In E. Börger, G. Hasenjaeger, and D. Rödding, editors, *Logic and Machines: Decision Problems and Complexity*, volume 171 of *LNCS*, pages 312–319. Springer, 1984.

[25] H. Ganzinger and H. de Nivelle. A superposition decision procedure for the guarded fragment with equality. In *Proceedings. 14th Symposium on Logic in Computer Science*, pages 295–303. IEEE Xplore, 1999.

[26] K. Gödel. Die vollständigkeit der axiome des logischen Funktionenkalküls. *Monatshefte für Mathematik und Physik*, 37(1):349–360, 1930.

[27] K. Gödel. Zum Entscheidungsproblem des logischen Funktionenkalküls. *Monatshefte für Mathematik und Physik*, 40(1):433–443, 1933.

[28] W. Goldfarb. The unsolvability of the gödel class with identity. *Journal of Symbolic Logic*, 49:1237–1252, 1984.

[29] E. Grädel. Satisfiability of formulae with one ∀ is decidable in exponential time. *Archiv für mathematische Logik und Grundlagenforschung*, 29(?):265–276, 1990.

[30] E. Grädel. On the restraining power of guards. *Journal of Symbolic Logic*, 64:1719–1742, 1999.

[31] E. Grädel, P. Kolaitis, and M. Vardi. On the decision problem for two-variable first-order logic. *Bulletin of Symbolic Logic*, 3(1):53–69, 1997.

[32] E. Grädel and M. Otto. The freedoms of (guarded) bisimulation. In editor, editor, *Johan van Benthem on Logic and Information Dynamics*, number 5 in Outstanding Contributions to Logic 5, pages 3–31. Springer International, 2014.

[33] E. Grädel, M. Otto, and E. Rosen. Two-variable logic with counting is decidable. In *Logic in Computer Science*, pages 306–317. IEEE, 1997.

[34] Sir William Hamilton. *Discussions on Philosophy and Literature, Education and University Reform.* William Blackwood and Sons, Edinburgh and London, 1853.

[35] Sir William Hamilton. *Lectures on Logic*, volume II. William Blackwood and Sons, Edinburgh and London, 1860.

[36] B. Herwig. Random models and the Gödel case of the decision problem. *Journal of Symbolic Logic*, 48(4):1120–1124, 1983.

[37] B. Herwig. Extending partial isomorphisms on finite structures. *Combinatorica*, 15(3):365–371, 1993.

[38] D. Hilbert and W. Ackermann. *Grundzüge der mathematischen Logik.* Check, 1st edition, 1928.

[39] D. Hilbert and W. Ackermann. *Grundzüge der mathematischen Logik.* Check, 2nd edition, 1938.

[40] W. Hodges. *Model Theory.* Encyclopedia of Mathematics and its applications. Cambridge University Press, Cambridge, 1997.

[41] E. Hrushovski. Extending partial isomorphisms of graphs. *Combinatorica*, 12(4):411–416, 1991.

[42] J Hsiang and M Rusinovich. Proving refutational completeness of theorem-proving strategies: the transfinite semantic tree method. *Journal of the ACM*, 38(3):559–587, 1991.

[43] W. Jevons. *Pure Logic and Other Minor Works.* Macmillan, London, 1890.

[44] W. Joyner. Resolution strategies as decision procedures. *Journal of the ACM*, 23(3):398–417, 1976.

[45] L. Kalmá. Über die Erfüllbarkeit derjenigen zählaudrücke, welche in der normalform zwei benachbarte allzeichen enthalten. *Mathematische Annalen*, 108:466–484, 1933.

[46] E. Kieroński, J. Michalyszyn, I. Pratt-Hartmann, and L. Tendera. Two-variable first-order logic with equivalence closure. *SIAM Journal on Computing*, 43(3):1012–1063, 2014.

[47] E. Kieroński and M. Otto. Small substructures and decidability issues for first-order logic with two variables. *Journal of Symbolic Logic*, 77:729–765, 2012.

[48] W. Kneale and M. Kneale. *The Development of Logic.* Clarendon Press, 1962.

[49] P. Kolaitis and M. Vardi. 0-1 laws and decision problems for fragments of second-order logic. *Information and Computation*, 87(?):302–338, 1990.

[50] D. Kozen. *Theory of Computation.* Texts in Computer Science. Springer, Berlin, 2006.

[51] S. Kripke. Semantical analysis of modal logic I: normal modal propositional calculi. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 9:67–96, 1963.

[52] R. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM Journal on Computing*, 6:467–480, 1980.

[53] L. Löwenheim. Über Möglichkeiten im Relativkalkül. *Math. Annalen*, 76:447–470, 1915.

[54] J. Łukasiewicz. *Aristotle's Syllogistic.* Clarendon Press, Oxford, 2nd edition, 1957.

[55] J. Martin. Aristotle's natural deduction revisited. *History and Philosophy of Logic*, 18(1):1–15, 1997.

[56] S. Maslov. The inverse method for establishing deducibility for logical calculi. *Proceedings, Steklov Institute of Mathematics*, 98:25–96, 1968.

[57] A. De Morgan. *Formal Logic: or, the calculus of inference, necessary and probable.* Taylor and Walton, London, 1847.

[58] A. De Morgan. On the syllogism, Part IV. *Transactions of the Cambridge Philosophical Society*, 10:331–357, 1860.

[59] M. Mortimer. On languages with two variables. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 21:135–140, 1975.

[60] N. Nishihara, K. Morita, and S. Iwata. An extended syllogistic system with verbs and proper nouns, and its completeness proof. *Systems and Computers in Japan*, 21(1):760–771, 1990.

[61] M. Otto. *Bisimulation invariance and finite models*, pages 276–298. Lecture Notes in Logic. Cambridge University Press, 2006.

[62] L. Pacholski, W. Szwast, and L. Tendera. Complexity of two-variable logic with counting. In *Logic in Computer Science*, pages 318–327. IEEE, 1997.

[63] C. Papadimitriou. *Computational Complexity.* Addison Wesley Longman. Springer, Reading, Mass., 2006.

[64] B. Poizat. *A Course in Model Theory.* Springer, New York, 2000.

[65] Ian Pratt-Hartmann. The hamiltonian syllogistic. *Journal of Logic, Language and Information*, 20(4):445–474, 2011.

[66] Ian Pratt-Hartmann and Lawrence S. Moss. Logics for the relational syllogistic. *Review of Symbolic Logic*, 2(4):647–683, 2009.

[67] F. P. Ramsay. On a problem of formal logic. *Proc. London Mathematical Society*, 30:264–286, 1930.

[68] J. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, 1965.

[69] K. Schütte. Untersuchungen zum entscheidungsproblem der mathematischen logik. *Mathematische Annalen*, 109:572–603, 1934.

[70] D. Scott. A decision method for validity of sentences in two variables. *Journal Symbolic Logic*, 27:477, 1962.

[71] J. Shepherdson. On the interpretation of Aristotelian syllogistic. *Journal of Symbolic Logic*, 21:137–147, 1956.

[72] T. Smiley. What is a syllogism? *Journal of Philosophical Logic*, 2:135–154, 1973.

[73] P. Spade. *Thoughts, Words and Things: An Introduction to Late Mediaeval Logic and Semantic Theory.* 2007-12-27. accessed 14.10.19.

[74] J. Surańyi. *Reduktionstheorie des Entscheidungsproblems im Prädikatenkalkül der ersten Stufe.* Ungarische Akademie der Wissenschaften, 1959.

[75] B. Trakhtenbrot. The impossibility of an algorithm for the decision problem for finite models. *Doklady Akademii Nauk*, 70:572–596, 1950. English translation: AMS Translations Series 2, vol. 33 (1963), pp. 1–6.

[76] Alan M Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London mathematical society*, 42(2):230–265, 1936.

[77] H. Wang. Proving theorems by pattern recognition ii. *The Bell System Technical Journal*, XL:1–41, 1961.

[78] Hao Wang. Dominoes and the ∀∃∀-case of the decision problem. In *Proceedings of Symposium on the Mathematical Theory of Automata*, pages 23–55. Brooklyn Polytechnic Institute, 1962.

[79] Dag Westerståhl. Aristotelian syllogisms and generalized quantifiers. *Studia Logica*, XLVIII(4):577–585, 1989.