BRICS

Basic Research in Computer Science

BRICS LS-95-5 D. P. Dubhashi: Complexity of Logical Theories

Complexity of Logical Theories

Devdatt P. Dubhashi

BRICS Lecture Series LS-95-5

ISSN 1395-2048 September 1995

See back inner page for a list of recent publications in the BRICS Lecture Series. Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK - 8000 Aarhus C
Denmark

Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@daimi.aau.dk

BRICS publications are in general accessible through WWW and anonymous FTP:

`http://www.brics.aau.dk/BRICS/`
`ftp ftp.brics.aau.dk (cd pub/BRICS)`

Complexity of Logical Theories

Devdatt P. Dubhashi

Devdatt P. Dubhashi
BRICS[1]
Department of Computer Science
University of Aarhus
Ny Munkegade
DK-8000 Aarhus C, Denmark

[1]**B**asic **R**esearch **I**n **C**omputer **S**cience, Centre of the Danish National Research Foundation.

Preface

These notes formed the basis for lectures I gave as part of the course "Fundamental results in Complexity Theory", taught by Erik Meineche Schmidt in the winter of 1994 at Århus University.

The idea of preparing such a collection was Erik's. His inspiration was two–fold. The first was Dexter Kozen's collection of lectures [16] from the courses he has taught at Cornell. These are distinguished from your typical textbook presentation by their informality, which preserves the chatty discursive style of the original lectures. As a result, they are highly readable and largely self–contained. This last feature even extends to various sub–modules which the discriminating reader can simply pick out and read without bothersome references to previous chapters, as in a typical textbook.

The second was the possibility of using World Wide Web, which renders our world increasingly smaller, to make this material widely available to instructors planning similar courses.

Dexter's notes were originally scribed by students attending the lectures, to form an outline which he then embellished. Each lecture in the book therefore, reflects fairly faithfully, the original lecture in the class. I had no such luck! Since I had to prepare the notes myself, I have chosen finally, to group them by subject rather than by individual lecture. This has resulted in five chapters that are well demarcated from each other.

In Chapters 1–3, I introduce the basic methodology for the study of the complexity of logical theories. The approach I follow is heavily influenced by the monograph of Ferrante and Rackoff [9], augmented by the ideas of Berman [2] and Kozen [15], as the reader will discern. The major components of the approach can be summarised by the contents of these chapters:

- Chapter 1 introduces the notion of (h, H)–bounded structures to reduce the decision problem for a class of structures to a finite version involving so–called "limited" quantifiers. The ubiquitous *method of quantifier elimination* is also introduced. (Since the students were familiar with the basic notions of model theory, I was able to give short shrift to laying out the Tarskian definitions of truth etc. Should the students lack this background, an instructor must, of course, develop these concepts carefully.)

- Chapter 2 introduces the alternating Turing machine model as a useful tool in analysing the complexity of logical theories. This model and its associated complexity classes are also related to the traditional complexity measures.

- Chapter 3 introduces the method of Ehrenfeucht–Fraïssé games to show that classes of structures are (h, H)–bounded as defined in Chapter 1.

The general techniques developed in Chapters 1–3 are then applied to two examples that form the last two chapters. The two examples are rather attractive choices in my opinion, because they are two of the most fundamental mathematical structures of special relevance to Computer Science, and because they illustrate the techniques very well, while avoiding unnecessary technical complications.

- Chapter 4 considers the theory of real addition. Matching upper and lower bounds are described in terms of alternating classes. The material is based on Ferrante and Rackoff [8] and Berman [2]. In the exercises (also see below), the theory of integer addition – so–called Presburger's Arithmetic – is developed in parallel. This material is based on Oppen [18].

- Chapter 5 treats the theory of Boolean algebras. The technique is inspired by Kozen [15], but I give a simpler account of the theory restricted to atomic algebras. This enables one to highlight the basic ideas while avoiding the technical complications resulting from the existence of atomless algebras. Rather than plunging into the well–developed and rather elaborate structure theory of Boolean algebras, I give a self–contained account, developing only the simple version of the representation theorem describing atomic and complete Boolean algebras as algebras of sets.

Exercises are distributed throughout the notes. They range from the routine verifications of definitions through simple extensions of the text material, to finally rather lengthy project assignments (for instance, Presburger's Arithmetic is developed in the exercises in Chapter 4). Exercises in the last category are marked with a $*$. Some exercises are full–fledged research problems to which I hope to attract the efforts of zealous readers.

Most of the material is not new, so originality is claimed almost solely for the presentation. For instance, the material in Chapters 4 and 5 appear for the first time couched in the unifying framework of Ehrenfeucht–Fraïssé games. I conceive of these notes as supplementing the monograph of Ferrante and Rackoff by updated material and points of view, and providing entry points into the literature. The serious student of this subject must pursue these sources.

I have tried to imitate Dexter's style of presentation in these notes. Those who have been fortunate enough to have attended Dexter's lectures could hardly disagree that this was always going to be a rather formidable undertaking! I would consider myself amply rewarded if, however, a few kind readers would opine that I have not utterly failed in this task!

I would like to express my heartfelt thanks: to Erik, for having given me the opportunity to give these lectures and for his active encouragement in publishing these notes. To Dany Breslauer, Gerth Stølting Brodal and Thore Husfeld for many suggestions; especially Thore for his valuable TeXpertise. And last but not he least, the students of

the course for having suffered through my lectures and having dutifully relayed back to me numerous corrections as they alertly caught them.

Contents

1 Introduction **1**
- 1.1 Models, Languages and Theories 1
- 1.2 Decision Problems and their Complexity 2
- 1.3 Bibliographical Notes 5

2 Alternation **7**
- 2.1 Alternating Turing Machines 7
- 2.2 Alternating Complexity Classes 9
- 2.3 Logical Theories in Alternating Classes 12
- 2.4 Bibliographical Notes 12

3 Ehrenfeucht–Fraïssé Games **13**
- 3.1 Ehrenfeucht–Fraïssé Games and Elementary Equivalence 13
- 3.2 Ehrenfeucht–Fraïssé Games and Bounded Structures 15
- 3.3 Bibliographical Notes 17

4 Real Addition **19**
- 4.1 Quantifier–Elimination 19
  - 4.1.1 A procedure 19
  - 4.1.2 Analysis 21
- 4.2 An EF Game 23
- 4.3 Lower Bound 26
- 4.4 A Research Problem 31
- 4.5 Bibliographical Notes 31

5 Boolean Algebras **33**
- 5.1 The Structure of Boolean Algebras 33
- 5.2 Decision Procedures 36
- 5.3 Some Lower Bounds 38
- 5.4 Extensions 41
- 5.5 A Research Problem 41
- 5.6 Bibliographical Notes 43

Chapter 1

Introduction

In this chapter, we quickly review some basic notions from logic and model theory and then introduce the fundamental problem that will concern us in these lectures.

1.1 Models, Languages and Theories

We quickly review some fundamental concepts from Universal Algebra and Model Theory.

A *signature* Σ, is a collection of *relation* and *function* symbols. Each relation symbol R or function symbol f, comes associated with an *arity*, $ar(R)$ and $ar(f)$, respectively which are non–negative integers. A function symbol with arity 0 is also called a *constant*.

An *algebra* or *structure* of signature Σ is a tuple $\mathsf{A} := (A, \{R\}, \{f\})$ comprising of an underlying *carrier* set A, and for each relation (function) symbol of Σ of arity k, a relation $R \subseteq A^k$ (a function $f : A^k \rightarrow A$). For $k \geq 0$, a k*–pointed* algebra is a tuple $(\mathsf{A}, \mathbf{a})$ where $\mathbf{a} \in A^k$. A *first–order* language for the signature Σ contains a symbol corresponding to every relation and function symbol in Σ, and in addition contains certain *logical* symbols $(\wedge, \vee, \neg, \exists, \forall, =)$ and a set of *variable* symbols $x_1, x_2, \ldots$.

Terms and *formulas* of the language are defined in the usual inductive fashion, see for example [11], Ch. II or [19], Ch. 5. An important notion for us will be the notion of the *quantifier–rank* of a formula which is defined by induction on the structure of the formula to be the maximum number of nested quantifiers in it. For example, the formula $\neg\exists x(\forall y \quad Rxy \wedge Qy)$ has quantifier rank 2. Following Tarski, one then defines, again inductively the relation of *satisfaction*,

$$(\mathsf{A}, \mathbf{a}_k) \models \varphi(\mathbf{x}_k)$$

between a k–pointed algebra and a formula with k free variables, [11], Ch. III or [19], Ch. 5. We shall consistently use the notational convention that $\mathbf{a}_k$ denotes an element of A^k and that $\varphi(\mathbf{x}_k)$ denotes a formula with free variables $\{x_1, \ldots, x_k\}$. An important special case of this definition is the case $k = 0$; a formula with no free variables is also called a *sentence* and we have the relation

$$\mathsf{A} \models \varphi$$

between an algebra and a sentence.

For an algebra A, define the *theory of* A by

$$\mathrm{Th}(\mathsf{A}) := \{\varphi \mid \mathsf{A} \models \varphi\}.$$

For a class of structures $\mathcal{C}$,define the theory of $\mathcal{C}$ by

$$\mathrm{Th}(\mathcal{C}) := \bigcap_{\mathsf{A} \in \mathcal{C}} \mathrm{Th}(\mathsf{A}).$$

Sometimes we will consider also

$$\mathrm{Sat}(\mathcal{C}) := \{\varphi \mid \neg\varphi \notin \mathrm{Th}(\mathcal{C})\},$$

the set of formulas *satisfiable* in some algebra in $\mathcal{C}$.

1.2 Decision Problems and their Complexity

The decision problem for a class of structures $\mathcal{C}$ is: given a (first–order) sentence φ, decide if $\varphi \in \mathrm{Th}(\mathcal{C})$. Equivalently, since $\varphi \in \mathrm{Th}(\mathcal{C})$ iff $\neg\varphi \notin \mathrm{Sat}(\mathcal{C})$, we can also phrase the decision problem as requiring us to decide if $\varphi \in \mathrm{Sat}(\mathcal{C})$.

At first sight, this might appear a formidable undertaking, given that the class $\mathcal{C}$ can be potentially infinite, so, the decision problem might seem to necessitate an infinite number of verifications: decide if $\varphi \in \mathrm{Th}(\mathsf{A})$ for each $\mathsf{A} \in \mathcal{C}$! Our task is much simplified if the class $\mathcal{C}$ satisfies the following property: $\varphi \in \mathrm{Th}(\mathcal{C})$ iff $\varphi \in \mathrm{Th}(\mathsf{A})$ for some "small" algebra $\mathsf{A} \in \mathcal{C}$. In order to formalise this notion, let us assume that we have a notion of a *norm* on the algebras in the class $\mathcal{C}$ at hand; for instance, this could be the size of the underlying carrier set for the finite algebras in $\mathcal{C}$. Let us denote this norm by $\|.\|$.

Definition 1 *Let $h : \mathsf{N} \to \mathsf{N}$. A class $\mathcal{C}$ is h–***bounded** *if for every sentence φ of quantifier rank at most n, $\varphi \in \mathrm{Sat}(\mathcal{C})$ iff $\varphi \in \mathrm{Th}(\mathsf{A})$ for an algebra $\mathsf{A} \in \mathcal{C}$ with norm $\|\mathsf{A}\| \leq h(n)$.*

So, for a h–bounded class of structures, we obtain the first reduction of our problem: to check if $\varphi \in \mathrm{Sat}(\mathcal{C})$, we may restrict attention to only those algebras A of norm at most $h(n)$, where n is the quantifier rank of φ.

Now, let us suppose that we are trying to decide if $(\mathsf{A}, \mathbf{a}) \models \varphi(\mathbf{x})$. If φ is quantifier free, this is quite easy from the definitions. Suppose not; we may assume without loss of generality that φ is of the form $(\exists y)\phi(y, \mathbf{x})$. Once again, we are involved with a potentially formidable search for this elusive witness y. The problem is simplified if we are assured that if there is such a y at all, then there is a "small" one. To formalise this, once again, we assume there is a notion of norm on the elements of an algebra, and use the same notation $\|.\|$ for it, (allowing the context to resolve any possible confusion).

Definition 2 *Let $H : \mathsf{N}^3 \to \mathsf{N}$, and let $n, k, m \geq 0$ be arbitrary integers. Let $\mathbf{a} \in A^k$ be such that $\|a_i\| \leq m$ for each $i \in [k]$. Let $\phi(x_1, \ldots, x_k, y)$ be a formula of quantifier rank at most n involving the free variables $x_1, \ldots, x_k$ and y, and let $\varphi(\mathbf{x}) := (\exists y)\phi(\mathbf{x}, y)$. An algebra A is H–**bounded** if $(\mathsf{A}, \mathbf{a}) \models \varphi(\mathbf{x})$ iff there exists a $a_{k+1} \in A$, with $\|a_{k+1}\| \leq H(n, k, m)$ such that $(\mathsf{A}, \mathbf{a}, a_{k+1}) \models \phi(\mathbf{x}, y)$.*

EXERCISE **3** *Let A be a H–bounded algebra, for some $H : \mathsf{N}^3 \to \mathsf{N}$. Show that*

$$(\mathsf{A}, \mathbf{a}_k) \models \forall y \quad \varphi(\mathbf{x}_k, y),$$

for elements $a_i \in A$ with $\|a_i\| \leq m$ for each $i \in [k]$ iff

$$(\mathsf{A}, \mathbf{a}_k, a_{k+1}) \models \varphi(\mathbf{x}_k, y),$$

for all $a_{k+1} \in \{a \in A \mid \|a\| \leq H(n, k, m)\}$.

If we know that a structure is H–bounded for some appropriate H, then one can restrict the search for witnesses to elements of bounded norm. We shall use the so–called *limited* quantifiers $\exists x \leq m\phi(x)$ and $\forall x \leq m\phi(x)$ to abbreviate $\bigvee_{\|x\| \leq m} \phi(x)$ and $\bigwedge_{\|x\| \leq m} \phi(x)$ respectively. The following proposition yields directly a decision procedure for the theory of a bounded structure.

Proposition 4 *Let A be a H–bounded algebra. Let $n, k \in \mathsf{N}$ and let $Q_1 x_1 Q_2 x_2 \cdots Q_k x_k \phi(x_1, x_2, \ldots, x_k)$ be a sentence with quantifier rank at most $n + k$, (i.e. ϕ has quantifier rank at most n). Let $m_0, m_1, \ldots, m_k$ be a sequence of integers such that $m_0 \leq m_1 \leq \cdots \leq m_k$ and $H(n + k - i, i - 1, m_{i-1}) \leq m_i$ for each $i \in [k]$. Then*

$$\mathsf{A} \models Q_1 x_1 Q_2 x_2 \cdots Q_k x_k \phi(x_1, x_2, \ldots, x_k)$$

iff

$$\mathsf{A} \models (Q_1 x_1 \leq m_1)(Q_2 x_2 \leq m_2) \cdots (Q_k x_k \leq m_k)\phi(x_1, x_2, \ldots, x_k).$$

Proof. We prove by induction on i, that for $0 < i \leq k + 1$,

$$\mathsf{A} \models Q_1 x_1 Q_2 x_2 \cdots Q_k x_k \phi(x_1, x_2, \ldots, x_k).$$

iff

$$\mathsf{A} \models (Q_1 x_1 \leq m_1)(Q_2 x_2 \leq m_2) \cdots (Q_{i-1} x_{i-1} \leq m_{i-1}) Q_i x_i \ldots Q_k x_k \phi(x_1, x_2, \ldots, x_k)$$

For $i = 1$ this is tautologically true. Assume it for some $i > 0$ and prove the statement for $i + 1$. Assume without loss of generality that the quantifier Q_i is $\exists$. Let $\mathbf{a}_{i-1} \in A^{i-1}$ be such that $\|a_j\| \leq m_j$ for all $j \in [i - 1]$. Then, since A is H–bounded, we have

$$(\mathsf{A}, \mathbf{a}_{i-1}) \models \exists x_i [Q_{i+1} x_{i+1} \ldots Q_k x_k \phi(\mathbf{x}_{i-1}, x_i, x_{i+1}, \ldots, x_k)]$$

iff

$$(\mathsf{A}, \mathbf{a}_{i-1}) \models \exists x_i \leq H(n + k - i, i - 1, m_{i-1})[Q_{i+1} x_{i+1} \ldots Q_k x_k \phi(\mathbf{x}_{i-1}, x_i, x_{i+1}, \ldots, x_k)]$$

Since $H(n+k-i, i-1, m_{i-1}) \leq m_i$, we have that for all $\mathbf{a}_{i-1} \in A^{i-1}$, such that $a_j \leq m_j$ for all $j \in [i-1]$,

$$(\mathsf{A}, \mathbf{a}_{i-1}) \models \exists x_i [Q_{i+1} x_{i+1} \ldots Q_k x_k \phi(\mathbf{x}_{i-1}, x_i, x_{i+1}, \ldots, x_k)] \leftrightarrow$$
$$(\mathsf{A}, \mathbf{a}_{i-1}) \models (\exists x_i \leq m_i)[Q_{i+1} x_{i+1} \ldots Q_k x_k \phi(\mathbf{x}_{i-1}, x_i, x_{i+1}, \ldots, x_k)] \tag{1.1}$$

Hence,

$$\mathsf{A} \models Q_1 x_1 Q_2 x_2 \cdots Q_k x_k \phi(x_1, x_2, \ldots, x_k)$$

iff (by the induction hypothesis)

$$\mathsf{A} \models (Q_1 x_1 \leq m_1)(Q_2 x_2 \leq m_2) \cdots (Q_{i-1} x_{i-1} \leq m_{i-1}) Q_i x_i \ldots Q_k x_k \phi(x_1, x_2, \ldots, x_k)$$

iff (by the equivalence (1.1))

$$\mathsf{A} \models (Q_1 x_1 \leq m_1)(Q_2 x_2 \leq m_2) \cdots (Q_i x_i \leq m_i) Q_{i+1} x_{i+1} \ldots Q_k x_k \phi(x_1, x_2, \ldots, x_k)$$

which completes the induction. ∎

Assume that there are only a finite number of elements of a given (finite) norm, and that furthermore, we can effectively enumerate these elements. Then we may in a straightforward fashion decide the sentence $(Q_1 x_1 \leq m_1)(Q_2 x_2 \leq m_2) \cdots (Q_k x_k \leq m_k)\phi(x_1, x_2, \ldots, x_k)$ with limited quantifiers, and thereby (via the equivalence of the preceding proposition) also decide the sentence $Q_1 x_1 Q_2 x_2 \cdots Q_k x_k \phi(x_1, x_2, \ldots, x_k)$.

Definition 5 *A class of structures $\mathcal{C}$ is said to be (h, H)–***bounded** *if it is h–bounded and each algebra in $\mathcal{C}$ is H–bounded.*

Using the reductions above, we may decide the theory of classes of structures assuming that they are (h, H)–bounded for suitable functions. In the next lecture, we shall introduce a model of computation, the *alternating Turing machine* which enables one to easily decide a sentence of the above form. In the following lecture, we will introduce the method of Ehrenfeucht–Fraïssé games to show that various structures are suitably bounded.

Another related useful technique is the *method of quantifier elimination.* Given a formula $\varphi(\mathbf{x}) := \exists y \phi(y, \mathbf{x})$, one can sometimes find a formula $\varphi'(\mathbf{x})$ from which the quantifier $\exists y$ has been eliminated, but which is nevertheless *equivalent* to φ in the sense that

$$(\mathsf{A}, \mathbf{a}) \models \varphi(\mathbf{x}) \quad \text{iff} \quad (\mathsf{A}, \mathbf{a}) \models \varphi'(\mathbf{x})$$

So with respect to Th(A), the formula φ may be replaced by the formula φ' of strictly lower quantifier rank.

EXAMPLE **6** *Consider the structure $(\mathsf{R}, \leq, 0, +, \cdot)$. We have the following simple example of quantifier elimination:*

$$(\mathsf{R}, a) \models \exists y \quad y \cdot y = x \qquad \textit{iff} \qquad (\mathsf{R}, a) \models 0 \leq x.$$

∎

EXERCISE **7** *For the structure* $(\mathsf{R}, \leq, 0, +, \cdot)$*, write quantifier–free equivalent formulas for the following formulas:*

1. $\varphi(a, b, c) := \exists x \quad a \cdot x \cdot x + b \cdot x + c = 0.$

2. $\phi(a, b, c) := \forall x \quad a \cdot x \cdot x + b \cdot x + c \neq 0.$

(Note that they express respectively, the existence and non–existence of roots to a certain quadratic equation.)

Assuming we can always find such quantifier–free equivalent formulas at every step, we can iterate this procedure to arrive at a *quantifier–free* formula that is equivalent to the original formula. Since this final formula is simply a finite boolean combination of atomic formulas, we can easily decide such a formula. In this way, we also obtain a decision procedure. This procedure is intuitive and explicit in producing a *witness* for the asserted decision, but can be expensive in the space requirement to write down the quantifier–free equivalent formula.

1.3 Bibliographical Notes

A compact but very careful introduction to logic and model theory is the text [11]. Our exposition of the complexity of logical theories follows the approach of Ferrante and Rackoff [9], but the formulation in alternating complexity classes is inspired by [2, 15, 1].

Chapter 2

Alternation

We introduce the concept of *alternation* in Turing machines, a generalisation of the more familiar notion of non–determinism. We introduce correspondingly, *alternating complexity classes*. These alternating classes will prove to be exactly the right classes to capture the complexity of logical theories. We will also give general results that enable one to relate the alternating classes to the more familiar time and space classes.

2.1 Alternating Turing Machines

An *alternating Turing machine* (ATM) is a tuple $M := (Q, \Sigma, \Gamma, \delta, q_0, g)$ where Q is a finite set of *states*, Σ is the *input alphabet*, $\Gamma \supseteq \Sigma$ is the *tape alphabet*, $\delta \subseteq Q \times \Gamma \times Q \times \Gamma \times \{left, right\}$ is the *next move* relation, $q_0 \in Q$ is the *initial state* and $g : Q \rightarrow \{\wedge, \vee, \neg, \mathbf{accept}, \mathbf{reject}\}$ is a labelling function on the states. If $g(q) = \wedge$, (respectively, $\vee, \neg$, **accept**,**reject**), q is said to be a *universal* (respectively *existential*, *negating*, *accepting* or *rejecting*) state.

On input x, the ATM starts in state q_0 with the head scanning the leftmost symbol of the input x written on the tape. A *step* of M currently in state q consists of reading the tape symbol under the head, writing a new symbol , moving the head left or right one square, and entering a new state in accordance with the transition relation δ. A *configuration* of ATM M is an element of

$$C_M := \Gamma^* \times Q \times \Gamma^*,$$

intending that the machine is in state q, the tape contents are the two strings concatenated and the head is reading the first symbol of the right string. We put the *space* of a configuration $\alpha := (x, q, y)$ to be $\text{space}(\alpha) = |x| + |y|$. If q is the state associated with a configuration α, then α is said to be universal etc. accordingly as q is. The *initial configuration* of M on input x is $\sigma_M(x) := (\epsilon, q, x)$ (where ϵ is the *empty* string).

Given M, and two configurations α, β, we write $\alpha \vdash \beta$ and say β is a *successor* of α if β results from α in one step by an application of the transition relation δ. (The relation $\vdash$ is not necessarily single–valued as δ is not.) We require that δ is such that universal and existential configurations have at least one successor and negating configurations have exactly one. Inductively, we define $\vdash^0 := id$, and for a non–negative integer n, $\vdash^{n+1} := \vdash \circ \vdash^n$. Denote $\vdash^{\leq n} := \bigcup_{0 \leq i \leq n} \vdash^i$. The reflexive and transitive closure of $\vdash$, $\bigcup_{i \geq 0} \vdash^i$ is denoted $\vdash^*$. A computation path of M on input x is a sequence $\sigma_M(x) =: \alpha_0 \vdash \cdots \vdash \alpha_n$ for some $n \geq 0$. The *full computation tree* of M on input x, $T_M(x)$, is the tree of configurations rooted at $\sigma_M(x)$ and such that if $\alpha \in T_M(x)$ and $\alpha \vdash \beta$, then also $\beta \in T_M(x)$.

Acceptance for ATMs is defined to capture the following intuition of *parallel* Turing machine computation. A single process is started in the initial configuration $\sigma_M(x)$. Subsequently, if a process is in an existential configuration α, and $\beta_1, \cdots, \beta_m$ are the configurations following from α in one step, then the process corresponding to α spawns m distinct new offspring processes which in parallel try to determine if any of the β_i lead to acceptance. Some of the offspring computations may be infinite, but if at least one is accepting, this is reported back to the parent process, and thereby α also leads to acceptance. If the initial process corresponded to a universal configuration α, then it leads to acceptance iff *all* the offspring processes β_i report acceptance. A process in a negating configuration leads to acceptance iff the successor configuration leads to rejection. A process in an accepting (rejecting) configuration merely reports that to its parent.

To formalise this notion of acceptance[1], we need to avoid the technical difficulty that there may be potentially infinite paths in the computation tree, preventing a naive labelling procedure to determine acceptance. To this end, define a *permissible* subtree T of the full computation tree $T_M(x)$ on input x as follows:

- T is *finite*.
- The root of T is $\sigma_M(x)$.
- If $v \in T$ is an existential configuration, then at least one of the children configurations is in T. If $v \in T$ is universal, then all children of v are in T.
- A leaf is either an accepting or a rejecting configuration.

Given a permissible subtree T, of $T_M(x)$, we define a 0–1 labelling l of its nodes as follows. An accepting (rejecting) configuration is labelled 1(0). For a non–leaf node v, the label associated with v is determined by applying the boolean operation given by $g(v)$ to the labels of its children. For example, if v is existential ($g(v) = \vee$) and has children $w_1, \ldots, w_k$, then $l(v) := l(w_1) \vee \cdots \vee l(w_m)$ The tree T is an *accepting* subtree of $T_M(x)$ if the root of T gets label 1 in this fashion. It is a *rejecting* subtree if the root gets labelled 0.

An ATM M *accepts* input x if there is an accepting permissible subtree of $T_M(x)$.An ATM M *rejects* input x if there is an rejecting permissible subtree of $T_M(x)$. Define:

$$L(M) := \{x \in \Sigma^* \mid M \text{ accepts } x\}.$$

Proposition 8 *The family of sets recognised by ATMs is exactly the family of recursively enumerable sets.*

[1] Our definition differs from that in [1] which proceeds via least fixed points of monotone operators. Our definition, while less elegant, is equivalent to theirs and conceptually simpler.

Proof. One direction follows from the fact that deterministic TMs are just a special case of ATMs. For the reverse containment, observe that we may enumerate all permissible trees one–by–one (say, by increasing depth) and check if any of them are accepting. ■

For quantitative complexity, we introduce the natural time and space measures. An ATM M is said to *accept x in time t* if there is a permissible accepting subtree T of $T_M(x)$ of depth at most t i.e. $T \subseteq C$ where

$$C = C(\sigma_M(x), t) := \{\beta \mid \sigma_M(x) \vdash^{\leq t} \beta\}.$$

M accepts x in space s, if there is a permissible accepting subtree T of $T_M(x)$ each of whose nodes correspond to configurations of size at most s, i.e. $T \subseteq D$ where

$$D = \{\beta \mid \text{space}(\beta) \leq s\}.$$

Let $T, S : \mathsf{N} \to \mathsf{N}$. We say that M *accepts in time $T(n)$* (respectively *space $S(n)$*) if for each $x \in L(M)$, M accepts x in time at most $T(|x|)$ (respectively space at most $S(|x|)$.

Exercise **9** *Argue that these measures correspond to the usual time and space measures for non–deterministic and deterministic TMs.*

The next exercise justifies the name *alternating* TM:

Exercise **10** *Show that for every ATM M, there is an ATM N such that N has no negating states, $L(M) = L(N)$ and for all $x \in L(M)$ and all $s, t \geq 0$, if M accepts x in time t (respectively space s) then N accepts x in time t (respectively space s).*

2.2 Alternating Complexity Classes

We introduce alternating complexity classes which capture the notion of *simultaneous* resource bounds.

An alternating TM, M (with no negating states, cf. Ex. 10) is said to be $A(n)$–*alternation bounded* on input x if whenever

$$\sigma_M(x) =: \alpha_0 \vdash^* \alpha_1 \vdash^* \cdots \vdash^* \alpha_m$$

with α_i a universal (existential) configuration whenever α_{i+1} is an existential (universal) configuration, for $0 \leq i < m$, then $m < A(|x|)$. In other words, any $\vdash$ derivation starting at $\sigma_M(x)$ alternates universal and existential configurations at most $A(|x|) - 1$ times. Hence, an ATM whose start state is existential and which is 1 alternation bounded is the same as a non–deterministic TM. By convention we agree that a 0 alternations bounded ATM is one each of whose states is existential and for which each configuration has a unique successor, namely a plain old deterministic TM.

We define $\Sigma STA(S(n), T(n), A(n))$ to be the class of languages accepted by alternating TMs that are simultaneously $S(n)$ space bounded, $T(n)$ time bounded and $A(n)$ alternation bounded, and whose start states are existential. Dually we define the $\Pi STA(S(n), T(n), A(n))$ classes. We denote by $STA(S(n), T(n), A(n))$ the union of the corresponding Σ and Π classes. Also an asterisk such as in $\Sigma STA(*, g(n), n)$ indicates no limit on the corresponding resource.

In analogy with the familiar classes, we introduce:

$$ATIME(T(n)) := STA(*, T(n), *)$$

and

$$ASPACE(S(n)) := STA(S(n), *, *).$$

Using these, one can define, again by analogy,

$$ALOGSPACE := ASPACE(\log n) = STA(\log n, *, *)$$

$$APTIME := \bigcup_{k \geq 0} ATIME(n^k) = \bigcup_{k \geq 0} STA(*, n^k, *)$$

$$APSPACE := \bigcup_{k \geq 0} ASPACE(n^k) = \bigcup_{k \geq 0} STA(n^k, *, *)$$

$$AEXPTIME := \bigcup_{c \geq 0} STA(*, 2^{n^c}, *)$$

In order to compare these classes against the usual time and space classes, we give the following general theorem relating alternating, deterministic and non–deterministic classes.

Proposition 11 *Let $T, S : \mathsf{N} \to \mathsf{N}$ be constructible.*

1. *If $S(n) \geq n$, then*

$$NSPACE(S(n)) \subseteq \bigcup_{c>0} ATIME(c \cdot S(n)^2).$$

2. *If $T(n) \geq n$, then*

$$ATIME(T(n)) \subseteq DSPACE(T(n)).$$

3. *If $S(n) \geq \log n$, then*

$$ASPACE(S(n)) \subseteq \bigcup_{c>0} DTIME(c^{S(n)}).$$

4. *If $T(n) \geq n$ and $c > 0$, then*

$$DTIME(T(n)) \subseteq ASPACE(c \cdot \log T(n)).$$

5. *If $S(n) \geq \log n$, then*

$$ASPACE(S(n)) = \bigcup_{c>0} DTIME(c^{S(n)}).$$

Proof. We shall prove (2) which we will have occasion to use most extensively, leaving the rest as exercises. We will show how an alternating TM M which accepts in time $T(n)$ can be simulated by a deterministic TM M' which accepts in space $T(n)$. The simulation strategy is to start with the initial configuration $\sigma_M(x)$, write down the time bound $T(n)$ and then traverses the computation tree with configurations from the set

$$C := \{\beta \mid \sigma_M(x) \vdash^{\leq T(n)} \beta\}$$

calculating the 0–1 labelling *in postorder*. M' accepts iff the root gets labelled 1 in this fashion. Let us check the space requirement for this strategy. At any point of the computation, if M' is visiting a node labelled α, then only α and the position of α in the tree need appear on the tape. The string may be a d–ary numeral, where d is the maximum fan–out of any configuration of M, denoting the unique path from the root to that vertex. Such a string need be of length at most $T(n)$. Hence M' accepts in space at most $T(n)$. ∎

EXERCISE **12** *Prove the remaining parts of Proposition 11.*

These general results characterising the power of alternation directly yield some striking relationships between alternating classes and the usual classes, partly summarised in the motto accompanying the following corollary:

Corollary 13 ("Alternating Time = Deterministic Space")

$$AEXPTIME = EXPSPACE$$

$$APSPACE = EXPTIME$$

$$APTIME = PSPACE$$

$$ALOGSPACE = PTIME$$

EXERCISE **14** *Apply Proposition 11 to prove Corollary 13.*

In addition, one can define the following classes for each non–negative integer k:

$$A\Sigma_k^p := \bigcup_{i \geq 0} \Sigma STA(*, n^i, k)$$

and dually,

$$A\Pi_k^p := \bigcup_{i \geq 0} \Pi STA(*, n^i, k).$$

Proposition 15 *For each $k \geq 0$, we have*

$$A\Sigma_k^p = \Sigma_k^p \quad and \quad A\Pi_k^p = \Pi_k^p$$

where Σ_k^p, Π_k^p are the classes of the **Polynomial Time Hierarchy***.*

EXERCISE **16** *Prove the above Proposition.(*HINT*: Consider the language QBF of* **quantified boolean formulas***, see [19, p. 428]. For $k \geq 1$, let ΣQBF_k denote the set of quantified boolean formulas whose first quantifier is existential and in which there are at most $k-1$ alternations of quantifiers. Dually define ΠQBF_k.*

1. *Show that $\Sigma QBF_k \in A\Sigma_k^p$ and $\Pi QBF_k \in A\Pi_k^p$.*
2. *Adapt the proof of Cook's Theorem (see for example, [19, pp. 171–172]) to show that ΣQBF_k is complete for $A\Sigma_k^p$ under log–space reductions, and ΠQBF_k similarly for $A\Pi_k^p$.)*

This last proposition makes the ATM model an attractive one in order to study the polynomial time hierarchy.

2.3 Logical Theories in Alternating Classes

Alternating TM classes enable one to precisely capture the complexity of logical theories in the sense that the corresponding decision problems are complete for some particular $STA(\cdot,\cdot,\cdot)$ classes under logspace reductions. We shall see concrete examples later on. For the moment we observe that ATMs enable us to easily derive upper bounds on the complexity of the decision problems of (h, H)–bounded structures.

Proposition 17 *Let $h, H : \mathsf{N} \to \mathsf{N}$.*

- *Let A be a H–bounded structure. Then*

$$\mathrm{Th}(\mathsf{A}) \in \bigcup_{c>0} STA(*, c \cdot \log(\hat{H}(n,n)), n),$$

  where $\hat{H}$ is defined recursively by $\hat{H}(n,0) := 0$, and for $i > 0$, $\hat{H}(n,i) := H(n-i, i-1, \hat{H}(n, i-1))$.

- *Let $\mathcal{C}$ be a class of (h, H)–bounded structures. Then*

$$\mathrm{Sat}(\mathcal{C}) \in \bigcup_{c>0} \Sigma STA(*, c \log h(n) + c \log \hat{H}(n,n), n).$$

EXERCISE **18** *Prove Proposition 17 by giving an alternating TM algorithm that decides* $\mathrm{Th}(\mathsf{A}), \mathrm{Sat}(\mathcal{C})$ *in the stated resource bounds. Use Proposition 4 from Chapter 1.*

2.4 Bibliographical Notes

Alternating Turing machines were introduced in [1]. The relevance of alternating classes to the classification of the complexity of logical theories was pointed out by [2].

Chapter 3

Ehrenfeucht–Fraïssé Games

We shall introduce the method of Ehrenfeucht–Fraïssé Games in order to show that certain (classes of) structures are suitably bounded.

3.1 Ehrenfeucht–Fraïssé Games and Elementary Equivalence

Given two k–pointed algebras $(\mathsf{A}, \mathbf{a}_k)$ and $(\mathsf{B}, \mathbf{b}_k)$, we shall say they are (n, k)–elementarily equivalent, and write $(\mathsf{A}, \mathbf{a}_k) \equiv_{n,k} (\mathsf{B}, \mathbf{b}_k)$ if for every formula $\varphi(\mathbf{x}_k)$ with quantifier rank at most n, we have $(\mathsf{A}, \mathbf{a}_k) \models \varphi(\mathbf{x}_k)$ iff $(\mathsf{B}, \mathbf{b}_k) \models \varphi(\mathbf{x}_k)$. That is, the two algebras satisfy exactly the same formulas of quantifier rank at most n or, intuitively, the two algebras cannot be told apart by formulas of low rank. Two algebras A and B are said to be *elementarily equivalent* if they are $(n, 0)$–equivalent for all $n \geq 0$.

How can we determine if two algebras are (n, k) equivalent (short of checking them on all possible formulas!)? Ehrenfeucht, building on work of Fraïssé gave the following appealing game–theoretical characterisation. The *Ehrenfeucht–Fraïssé (EF) Game* for k–pointed algebras $(\mathsf{A}, \mathbf{a}_k)$ and $(\mathsf{B}, \mathbf{b}_k)$ is played between two players Shiva [1] and Vishnu [2] on two "boards", which are the two structures. For every $r \geq 1$, there is a red pebble and a blue pebble numbered r. The players take turns to place these pebbles on the boards. Initially the red pebbles numbered 1 through k are on the elements $a_1, \ldots, a_k$ and the corresponding blue pebbles are on the elements $b_1, \ldots, b_k$. The game now proceeds in a series of rounds. In round $r \geq k$ first Shiva picks up the red pebble numbered r and places it on an element on one of the two boards. Vishnu responds by placing the corresponding blue pebble on an element on the *other* board. In the n–round game, play stops after round $n + k$ and Vishnu wins iff the mapping $a_i \mapsto b_i$ for $i \in [n + k]$ is a *partial isomorphism* between the two structures; otherwise Shiva wins.

A mapping f between two k–pointed algebras $(\mathsf{A}, \mathbf{a}_k)$ and $(\mathsf{B}, \mathbf{b}_k)$ is a *homomor-*

[1] The "Destroyer" in Hindu mythology..

[2] and The "Preserver".

phism if[3]

- For $i \in [k]$, $f(a_i) = b_i$.
- For every relation symbol R of arity r, and all $c_1, \ldots, c_r \in A$ we have

$$R^A(c_1, \cdots, c_r) \quad \text{iff} \quad R^B(f(c_1), \cdots, f(c_r)).$$

A homomorphism which is also injective and surjective is called an *isomorphism*. A map f is a *partial* isomorphism if it is an isomorphism on the substructures obtained by restriction to the domain and range of f.

Thus Shiva's goal is to try to show that the two structures are different i.e. *destroy* the isomorphism while Vishnu's aim is to show they are the same by *preserving* it. A *strategy* for either player is a rule that tells the player, given the current state of the board and the past history of all moves, how to make the next move. A player is said to have a *winning* strategy for, or simply, to *win* a particular game, if He has a strategy that always guarantees a win for Him, no matter what strategy the opposing player adopts.

Theorem 19 *We have* $(\mathsf{A}, \mathbf{a}_k) \equiv_{n,k} (\mathsf{B}, \mathbf{b}_k)$ *iff Vishnu wins the n–round Ehrenfeucht–Fraïssé Game on the two structures.*

Proof. Let us first prove the "if" part by induction on $n \geq 0$. Assume that Vishnu has a winning strategy for the n–round EF–game on $(\mathsf{A}, \mathbf{a}_k)$ and $(\mathsf{B}, \mathbf{b}_k)$. We shall show that this implies $(\mathsf{A}, \mathbf{a}_k) \equiv_{n,k} (\mathsf{B}, \mathbf{b}_k)$. The $n = 0$ case holds by definition. For $n > 0$, let us take a formula (without loss of generality) $\varphi(\mathbf{x}_k) := \exists y \phi(\mathbf{x}_k, y)$ with ϕ a formula of quantifier rank less than n. Suppose $(\mathsf{A}, \mathbf{a}_k) \models \varphi(\mathbf{x})$. Then, by definition, there exists a $a_{k+1} \in A$ such that $(\mathsf{A}, \mathbf{a}_k, a_{k+1}) \models \phi(\mathbf{x}, y)$. Since Vishnu has a winning strategy for the n–round game on $(\mathsf{A}, \mathbf{a}_k)$ and $(\mathsf{B}, \mathbf{b}_k)$, He has a counter–move $b_{k+1} \in B$, to Shiva's move to pick a_{k+1} which ensures that He will also win the $(n-1$–round game on the structures $(\mathsf{A}, \mathbf{a}_k, a_{k+1})$ and $(\mathsf{B}, \mathbf{b}_k, b_{k+1})$. By induction, this implies that $(\mathsf{A}, \mathbf{a}_k, a_{k+1}) \equiv_{n-1,k+1} (\mathsf{B}, \mathbf{b}_k, b_{k+1})$. In particular, $(\mathsf{B}, \mathbf{b}_k, b_{k+1}) \models \phi(\mathbf{x}_k, y)$. Hence $(\mathsf{B}, \mathbf{b}_k) \models \varphi(\mathbf{x}_k)$. The roles of A, B can be reversed to complete the proof of the "if" implication.

We prove the "only–if" implication also by induction on $n \geq 0$. Assuming that $(\mathsf{A}, \mathbf{a}_k) \equiv_{n,k} (\mathsf{B}, \mathbf{b}_k)$, we shall show that Vishnu has a winning strategy for the $(n+k)$–round game on the two structures. For $n = 0$, the null strategy is winning for Vishnu. For $n > 0$, assume by induction the statement for all values upto $n - 1$. Consider now, the n–round game on $(\mathsf{A}, \mathbf{a}_k)$ and $(\mathsf{B}, \mathbf{b}_k)$. Suppose Shiva picks the element $a_{k+1} \in A$. Let us now consider all possible *non–equivalent* formulas of quantifier–rank less than n involving $k+1$ free variables; it can be shown that there are only finitely many of these, [11, p. 189]. Let $\psi(\mathbf{x}_{k+1}$ be the conjunction of these (finitely many) formulas or their negations (accordingly as they are satisfied in $(\mathsf{A}, \mathbf{a}_k, a_{k+1})$ or not). By the assumption

[3]To avoid technical complications, we restrict ourselves to signatures involving only relational symbols, see [11, Chap. XI, p. 181,186].

that $(\mathsf{A}, \mathbf{a}_k) \equiv_{n,k} (\mathsf{B}, \mathbf{b}_k)$, there is a $b_{k+1} \in B$ such that $(\mathsf{B}, \mathbf{b}_k, b_{k+1}) \models \psi(\mathbf{x}_{k+1})$. Moreover, $(\mathsf{A}, \mathbf{a}_k, a_{k+1}) \equiv_{n-1,k+1} (\mathsf{B}, \mathbf{b}_k, b_{k+1})$. By induction, Vishnu has a winning strategy in the $(n-1)$–round game on the structures $(\mathsf{A}, \mathbf{a}_k, a_{k+1})$ and $(\mathsf{B}, \mathbf{b}_k, b_{k+1})$. Vishnu's winning strategy for the $(n+k)$–round is to combine this strategy together with the choice of the particular b_{k+1} described above, for the first move. ∎

Exercise **20** *In this exercise, you are asked to apply the characterisation of Theorem 19 to study questions of elementary equivalence of structures over the reals and rationals.*

1. *Consider the two structures* $(\mathsf{R}, <)$ *and* $(\mathsf{Q}, <)$. *Show that there is no first-order sentence in the signature* $\{<\}$ *that can distinguish between the two structures i.e. the two structures are elementarily equivalent..*

2. *A linearly ordered structure* $(\mathsf{A}, <)$ *is said to be* dense *if for any two elements* $a_1 < a_2 \in A$, *there is an element* $a_3 \in A$ *such that* $a_1 < a_3 < a_2$. *Formulate and prove an appropriate generalisation of the first part.*

3. *Show that the structures* $(\mathsf{R}, 0, 1, +, <)$ *and* $(\mathsf{Q}, 0, 1, +, <)$ *are elementarily–equivalent.*

4. *Are the structures* $(\mathsf{R}, 0, 1, +, \cdot, <)$ *and* $(\mathsf{Q}, 0, 1, +, \cdot <)$ *elementarily equivalent?*

The following alternative inductive characterisation of the $\equiv_{n,k}$ relations will be our starting point in the next section.

Proposition 21 *Let* $(\mathsf{A}, \mathbf{a})$, *and* $(\mathsf{B}, \mathbf{b})$ *be two* k*–pointed algebras.*

1. $(\mathsf{A}, \mathbf{a}) \equiv_{0,k} (\mathsf{B}, \mathbf{b})$ *iff the map* $a_i \mapsto b_i$ *for* $i \in [k]$ *is a partial isomorphism between the two structures.*

2. *For* $n > 0$, $(\mathsf{A}, \mathbf{a}) \equiv_{n,k} (\mathsf{B}, \mathbf{b})$ *iff for every* $a_{k+1} \in A$ *(every* $b_{k+1} \in B$*), there exists a* $b_{k+1} \in B$ *(*$a_{k+1} \in A$*) such that* $(\mathsf{A}, \mathbf{a}, a_{k+1}) \equiv_{n-1,k+1} (\mathsf{B}, \mathbf{b}, b_{k+1})$.

Exercise **22** *Prove this proposition by induction. Prove it directly, or via the game–theoretical characterisation of Theorem 19.*

3.2 Ehrenfeucht–Fraïssé Games and Bounded Structures

Let $\mathcal{C}$ be a class of structures. For $n, k \in \mathsf{N}$, we define equivalence relations $\sim_{n,k}$ on k–pointed algebras which will be refinements of the $\equiv_{n,k}$ relations from the previous section. The following proposition will then be used to show that algebras in $\mathcal{C}$ are H–bounded.

Proposition 23 *Let* $H : \mathsf{N}^3 \to \mathsf{N}$. *Suppose for* $n, k \in \mathsf{N}$ *we have defined relations* $\sim_{n,k}$ *on* k*–pointed algebras in* $\mathcal{C}$ *such that for every two* k*–pointed algebras* $(\mathsf{A}, \mathbf{a})$ *and* $(\mathsf{B}, \mathbf{b})$ *in* $\mathcal{C}$, *and for every* $m \in \mathsf{N}$, *the following conditions obtain:*

1. $(\mathsf{A},\mathbf{a}) \sim_{0,k} (\mathsf{B},\mathbf{b})$ *implies* $(\mathsf{A},\mathbf{a}) \equiv_{0,k} (\mathsf{B},\mathbf{b})$.

2. *If* $(\mathsf{A},\mathbf{a}) \sim_{n+1,k} (\mathsf{B},\mathbf{b})$ *and* $\|b_i\| \leq m$ *for each* $i \in [k]$*, then for all* $a_{k+1} \in A$*, there exists a* $b_{k+1} \in B$ *with* $\|b_{k+1}\| \leq H(n,k,m)$ *and* $(\mathsf{A},\mathbf{a},a_{k+1}) \sim_{n,k+1} (\mathsf{B},\mathbf{b},b_{k+1})$.

Then

1. *For every* $n, k \in \mathsf{N}$ *and every two* k*–pointed algebras* $(\mathsf{A},\mathbf{a})$ *and* $(\mathsf{B},\mathbf{b})$ *in* $\mathcal{C}$*, we have* $(\mathsf{A},\mathbf{a}) \sim_{n,k} (\mathsf{B},\mathbf{b})$ *implies* $(\mathsf{A},\mathbf{a}) \equiv_{n,k} (\mathsf{B},\mathbf{b})$.

2. *Every structure in* $\mathcal{C}$ *is* H*–bounded.*

Proof. First we prove (1) by induction on n. By the definition of the relations $\sim$, the statement holds for $n = 0$ and all $k \in \mathsf{N}$. For the induction step, assume the statement holds for some $n \in \mathsf{N}$ and for all $k \in \mathsf{N}$. We prove the statement now for $n+1$. So, suppose that $(\mathsf{A},\mathbf{a}_k) \sim_{n+1,k} (\mathsf{B},\mathbf{b}_k)$, and let $\varphi(\mathbf{x}_k)$ be a formula of quantifier depth at most $n+1$. We will show that $(\mathsf{A},\mathbf{a}_k) \models \varphi(\mathbf{x}_k)$ iff $(\mathsf{B},\mathbf{b}_k) \models \varphi(\mathbf{x}_k)$. We may assume, without loss of generality, that $\varphi(\mathbf{x}_k) = \exists x_{k+1}\phi(\mathbf{x}_k, x_{k+1})$. If $(\mathsf{A},\mathbf{a}_k) \models \varphi(\mathbf{x}_k)$, then $(\mathsf{A},\mathbf{a}_k,a_{k+1}) \models \phi(\mathbf{x}_k,x_{k+1})$ for some $a_{k+1} \in A$. Since $(\mathsf{A},\mathbf{a}_k) \sim_{n+1,k} (\mathsf{B},\mathbf{b}_k)$, we have that for some $b_{k+1} \in B$, $(\mathsf{A},\mathbf{a}_k,a_{k+1}) \sim_{n,k+1} (\mathsf{B},\mathbf{b}_k,b_{k+1})$. By the induction hypothesis, this implies that $(\mathsf{A},\mathbf{a}_k,a_{k+1}) \equiv_{n,k+1} (\mathsf{B},\mathbf{b}_k,b_{k+1})$, hence, in particular that $(\mathsf{B},\mathbf{b}_k,b_{k+1}) \models \phi(\mathbf{x}_k,x_{k+1})$. So we have that $(\mathsf{B},\mathbf{b}_k) \models \varphi(\mathbf{x}_k)$. Symmetrically interchanging the roles of A and B,we also have the other implication.

To prove (2), let $n, k \in \mathsf{N}$, and let $\mathsf{A} \in \mathcal{C}$. Let $\phi(\mathbf{x}_{k+1})$ have quantifier rank at most n, and suppose $(\mathsf{A},\mathbf{a}_k) \models \exists x_{k+1}\phi(\mathbf{x}_k,x_{k+1})$, with $\|a_i\| \leq m$ for each $i \in [k]$. Then, for some $a_{k+1} \in A$, we have $(\mathsf{A},\mathbf{a}_k,a_{k+1}) \models \phi(\mathbf{x}_{k+1})$. Since $(\mathsf{A},\mathbf{a}_k) \sim_{n+1,k} (\mathsf{A},\mathbf{a}_k)$, we have, by the second condition in the hypothesis, that there is a $a'_{k+1} \in A$ such that $(\mathsf{A},\mathbf{a}_k,a_{k+1}) \sim_{n,k+1} (\mathsf{A},\mathbf{a}_k,a'_{k+1})$ and $\|a'_{k+1}\| \leq H(n,k,m)$. From the first part of the conclusion above, we have that this implies $(\mathsf{A},\mathbf{a}_k,a_{k+1}) \equiv_{n,k+1} (\mathsf{A},\mathbf{a}_k,a'_{k+1})$, in particular that $(\mathsf{A},\mathbf{a}_k,a'_{k+1}) \models \phi(\mathbf{x}_k,x_{k+1})$. So A is H–bounded. ∎

To show that condition (1) of Proposition 23 holds, we shall appeal to the following simple

Lemma 24 *Suppose,* $(\mathsf{A},\mathbf{a}_k) \sim_{0,k} (\mathsf{B},\mathbf{b}_k)$ *implies that* $(\mathsf{A},\mathbf{a}_k) \models \varphi(\mathbf{x}_k)$ *iff* $(\mathsf{B},\mathbf{b}_k) \models \varphi(\mathbf{x}_k)$ *for every atomic formula* $\varphi(\mathbf{x}_k)$*. Then* $(\mathsf{A},\mathbf{a}_k) \equiv_{0,k} (\mathsf{B},\mathbf{b}_k)$.

EXERCISE **25** *Prove Lemma 24 by induction on atomic formulas.*

Proposition 23 will be used in conjunction with the following one which applies to show that certain classes are h–bounded.

Proposition 26 *Let the relations* $\{\sim_{n,k}\}$ *satisfy the conditions of Proposition 23. Let* $h : \mathsf{N} \to \mathsf{N}$ *be such that for every* $n \in \mathsf{N}$*, and for every* $\mathsf{A} \in \mathcal{C}$*, there exists an algebra* $\mathsf{A}' \in \mathcal{C}$ *such that* $\mathsf{A} \sim_{n,0} \mathsf{A}'$ *and* $\|\mathsf{A}'\| \leq h(n)$*. Then* $\mathcal{C}$ *is* (h,H)*–bounded.*

Proof. By Proposition 23, it follows that every structure in $\mathcal{C}$ is H–bounded. It remains to show that for every formula φ with no free variables and quantifier rank at most n,

$$\varphi \in \mathrm{Sat}(\mathcal{C}) \quad \text{iff} \quad \mathsf{A} \models \varphi$$

for some algebra A with $\|\mathsf{A}\| \leq h(n)$. One implication is trivial. For the other, assume that for some $\mathsf{A}' \in \mathcal{C}$, we have $\mathsf{A}' \models \varphi$. By hypothesis, there is a $\mathsf{A} \in \mathcal{C}$ such that $\mathsf{A} \sim_{n,0} \mathsf{A}'$ with $\|\mathsf{A}\| \leq h(n)$. By the properties of the relation $\sim_{n,0}$, we also have $\mathsf{A} \equiv_{n,0} \mathsf{A}'$. Hence, in particular, $\mathsf{A} \models \varphi$. ∎

3.3 Bibliographical Notes

The model–theoretic notion of elementary equivalence is treated in many texts including [11, Chap. VI,§ 4]. A careful discussion of Ehrenfeucht–Fraïssé Games is in [11, Chap. XI]. Our use of them to establish boundedness of structures follows Ferrante and Rackoff [9].

Chapter 4

Real Addition

In this and the following chapter, we shall apply the general techniques developed in the previous chapters to solving the decision problem for two particular (classes of) structures.

In this chapter, we focus on *the theory of real addition*, namely the theory of the structure $(\mathsf{R}, +, <, 0, 1)$. Conspicuous by its absence is the multiplication operator. The full theory of the reals, $\mathrm{Th}(\mathsf{R}, +, \cdot, <, 0, 1)$ is considerably more complicated, see § 4.4 where we remark on it briefly..

4.1 Quantifier–Elimination

In addition to the usual purely logical machinery, our language will naturally have symbols to denote the operations of our algebra, namely $+, <$ and the constants $0, 1$. In addition, we shall augment our language to include symbols to represent and be interpreted by rational numbers; so for every two integers a, b, there will be the symbol $\frac{a}{b}$ to denote the corresponding rational number. The reason to include these symbols will be to facilitate the description of the quantifier elimination procedure. A term in the language is now of the form

$$\sum_{1 \le i \le n} \frac{a_i}{b_i} y_i := \frac{a_1}{b_1} y_1 + \cdots + \frac{a_n}{b_n} y_n,$$

Hence an atomic formula has the form $t_1 = t_2$ or $t_1 < t_2$, where t_1, t_2 are terms. The formulas are now built up as usual, and are interpreted over the reals in the standard Tarskian fashion.

4.1.1 A procedure

We shall prove:

Theorem 27 *For every formula $\varphi(\mathbf{x_k})$, there exists an equivalent quantifier–free formula $\varphi'(\mathbf{x}_k)$, and moreover, this can be obtained effectively.*

The proof will be by structural induction over formulas. We shall begin by dispensing with the easy cases involving only the atomic formulas and boolean connectives and no quantifiers. An atomic formula is already quantifier–free. If $\varphi := \phi \wedge \psi$, then recursively, $\varphi' := \phi' \wedge \psi'$ is the required quantifier–free equivalent. The other boolean connectives are similar.

For the non–trivial case, we give the following lemma to eliminate *one* quantifier:

Lemma 28 *Let $\varphi(x, \mathbf{x}_k)$ be quantifier–free. Then there is an effective procedure to obtain a quantifier–free formula $\varphi'(\mathbf{x}_k)$ which is equivalent to $\exists x \varphi(x, \mathbf{x}_k)$.*

Proof. First we describe the construction, and then we argue the correctness. The construction proceeds in three steps:

1. "Solve for x" in each atomic formula in φ to bring it into the form $\exists x \psi(\mathbf{x}_k, x)$ such that in ψ, every atomic formula either does not involve x, or has the form $t < x$, $t = x$ or $x < t$ where t is a term not involving x.

2. Define $\psi_\infty(\mathbf{x}_k)$ ($\psi_{-\infty}(\mathbf{x}_k)$) by replacing in ψ,

$$\begin{array}{rcl} x < t & \text{by} & 0 = 1 (0 = 0) \\ x = t & \text{by} & 0 = 1 (0 = 1) \\ t < x & \text{by} & 0 = 0 (0 = 1) \end{array}$$

   Observe that for every $\mathbf{a}_k \in \mathsf{R}^k$, if a is a sufficiently large real then $(\mathsf{R}, \mathbf{a}_k, a) \models \psi(\mathbf{x}_k, x)$ iff $(\mathsf{R}, \mathbf{a}_k) \models \psi_\infty(\mathbf{x}_k)$ and similarly, if $a \in \mathsf{R}$ is sufficiently small, $(\mathsf{R}, \mathbf{a}_k, a) \models \psi(\mathbf{x}_k, x)$ iff $(\mathsf{R}, \mathbf{a}_k) \models \psi_{-\infty}(\mathbf{x}_k)$. This explains the choice of the notation.

3. Let U denote the set of all terms, t (not involving x), such that the atomic formula $t < x$ or $t = x$ or $x < t$ appears in ψ. Then for $u, v \in U$, let $\psi_{u,v}(\mathbf{x}_k)$ be the result of substituting $x := (u + v)/2$ in $\psi(\mathbf{x}_k)$. Finally, we define

$$\varphi'(\mathbf{x}_k) := \psi_{-\infty}(\mathbf{x}_k) \vee \psi_\infty(\mathbf{x}_k) \vee \bigvee_{u,v \in U} \psi_{u,v}(\mathbf{x}_k).$$

   (Note that in the big disjunction, we include the terms $\psi_{u,u}$ for every $u \in U$.)

Let us now argue that for any $\mathbf{a}_k \in A^k$,

$$(\mathsf{R}, \mathbf{a}_k) \models \exists x \varphi(\mathbf{x}_k, x) \quad \text{iff} \quad (\mathsf{R}, \mathbf{a}_k) \models \varphi'(\mathbf{x}_k).$$

For the implication from right to left, if $(\mathsf{R}, \mathbf{a}_k) \models \psi_{u,v}(\mathbf{x}_k)$ for some $u, v \in U$, then in fact $(\mathsf{R}, \mathbf{a}_k, (u(\mathbf{a}_k) + v(\mathbf{a}_k))/2) \models \varphi(\mathbf{x}, x)$ [1]and so $(\mathsf{R}, \mathbf{a}_k) \models \exists x \varphi(\mathbf{x}, x)$. If $(\mathsf{R}, \mathbf{a}_k) \models \psi_\infty(\mathbf{x}_k)$, then, for sufficiently large[2] $a \in \mathsf{R}$ we have $(\mathsf{R}, \mathbf{a}_k, a) \models \varphi(\mathbf{x}, x)$, hence again, $(\mathsf{R}, \mathbf{a}_k) \models \exists x \varphi(\mathbf{x}, x)$. the argument for $\psi_{-\infty}$ is similar to that for ψ_∞.

[1] By $u(\mathbf{a}_k)$ we mean the standard interpretation of the term $u(\mathbf{x}_k)$ in the structure $(\mathsf{R}, \mathbf{a}_k)$

[2] A closer look shows that we only need take $a > (u(\mathbf{a}_k) + v(\mathbf{a}_k))/2$ for all $u, v \in U$.

For the reverse implication, suppose that $(\mathsf{R}, \mathbf{a}_k) \models \exists x \varphi(\mathbf{x}, x)$. Then $(\mathsf{R}, \mathbf{a}_k, a_{k+1}) \models \varphi(\mathbf{x}, x)$ for some $a_{k+1} \in \mathsf{R}$. Number the terms in U as $t_1, t_2, \ldots, t_r$ so that

$$t_1(\mathbf{a}_k) \leq t_2(\mathbf{a}_k) \leq \cdots \leq t_r(\mathbf{a}_k).$$

Let us also use $t_1, \ldots, t_r$ to denote these interpretations of the terms in U. Exactly one of the following conditions must obtain with respect to the relation between a_{k+1} and $t_1, \ldots, t_r$.

1. $a_{k+1} < t_1$.
2. $t_m < a_{k+1}$.
3. For some $i \in [r]$, $t_i = a_{k+1}$.
4. For some $1 \leq i < r$, $t_i < a_{k+1} < t_{i+1}$.

Then, we have respectively,

1. $(\mathsf{R}, \mathbf{a}_k) \models \psi_{-\infty}(\mathbf{x}_k)$.
2. $(\mathsf{R}, \mathbf{a}_k) \models \psi_{\infty}(\mathbf{x}_k)$.
3. $(\mathsf{R}, \mathbf{a}_k) \models \psi_{t_i,t_i}(\mathbf{x}_k)$.
4. $(\mathsf{R}, \mathbf{a}_k) \models \psi_{t_i,t_{i+1}}(\mathbf{x}_k)$.

Hence also $(\mathsf{R}, \mathbf{a}_k) \models \varphi'(\mathbf{x}_k)$. ∎

4.1.2 Analysis

To analyse the complexity of the quantifier–elimination procedure, we shall try to find how the length of the formulas and the size of the constants in them grow, as the procedure continues. For this purpose, we now introduce the *norm* that we will use to characterise the size of the constants. We define, for $x \in \mathsf{R}$,

$$\|x\| := \begin{cases} \infty, & \text{if } x \text{ is not rational;} \\ k, & \text{where } k := \min\{j \in \mathsf{N} \mid x = n/m, n, m \in \mathsf{N}, |n|, |m| \leq j\}. \end{cases}$$

It is easy to check that in fact this yields a norm on the reals. We will use this norm also in the analysis of the EF–game of the next section. The following easy lemma will be used repeatedly to estimate the norms of various constants appearing in our formulae:

Lemma 29 *Let $r_1, r_2 \ldots, r_k \in \mathsf{R}$ be such that $\|r_i\| \leq m$ for each $i \in [k]$. Then*

1. $\|r_1 \cdot r_2 \cdots r_k\| \leq m^k$.
2. $\|r_1 + r_2 + \cdots + r_k\| \leq k \cdot m^k$.

EXERCISE **30** *Supply a proof.*

We will use the notation $s(\varphi)$ to denote the largest norm of a constant appearing in the formula φ and $l(\varphi)$ to denote the length of φ. Let s_0, l_0 be these parameters for the original formula. First, let us analyse the growth of the coefficients in the three steps of the procedure for eliminating a single quantifier. Suppose we start with a formula φ with $s(\varphi) =: s$.

1. "Solve for x" involves subtracting one rational number from another and dividing by a third. Applying Lemma 29 twice, we conclude that $s(\psi) \leq 4s^4$.

2. No new constant is created by writing down ψ_∞ and $\psi_{-\infty}$.

3. In writing down $\psi_{u,v}$, we have to make the substitution $x := \frac{u+v}{2}$. Applying Lemma 29, we conclude that $s(\psi_{u,v}) \leq 4 \cdot (s(\psi))^2 \leq 64 \cdot s^8 \leq s^{14}$, provided $s \geq 2$.

Hence, overall, $s(\varphi') \leq s^{14}$.

Similarly, we compute how the length of the formula grows. Suppose $l(\varphi) =: l$. Then

1. First, we observe that trivially, $l(\psi_\infty), l(\psi_{-\infty}) \leq l$.

2. There are at most l^2 formulas of the form $\psi_{u,v}$. In forming $\psi_{u,v}$, note that in each of the at most l atomic formulas we may have to write two terms, each containing k rational coefficients bounded in norm by s^14, hence in length by $14 \cdot \lg s$. Hence $l(\psi_{u,v}) \leq l \cdot 2 \cdot k \cdot 2 \cdot 14 \lg s \leq 60l^3$.

Hence, finally, $l(\varphi') \leq 2l + l^2 \cdot 60l^3 \leq l^14$, provided $l \geq 2$,

We now compute an upper bound on the space and time required to carry out the full quantifier elimination procedure. Suppose we start with a formula (without loss of generality already in prenex normal form) of length l_0, with $n_0 \leq l_0$ quantifiers and with all coefficients in it bounded in norm by $s_0 \leq 2^{l_0}$. Then after eliminating all the quantifiers, the above analysis shows that the length of the largest formula obtained during the procedure is at most

$$l_0^{14n_0} \leq l_0^{14l_0} \leq 2^{2^{c_0 \cdot l_0}},$$

for some constant c_0. (This is of course a bit pessimistic.) And the largest norm of a constant appearing in the procedure is bounded by $s_0^{14n_0}$. Notice that if φ is a sentence, then the total space used in eliminating quantifiers is at most $2^{2^{c_0 \cdot l(\varphi)}}$. Since the total number of steps involved in each quantifier elimination and also in evaluating the final formula involving boolean relations between rational constants is some fixed polynomial, the procedure also operates in time $2^{2^{c_1 \cdot l(\varphi)}}$, for some constant c_1.

We can summarise with

Theorem 31 (Quantifier Elimination for the Reals) *There is a quantifier elimination procedure for the theory of real addition that runs in double exponential time and space.*

EXERCISE **32** **(Gerth Stølting Brodal)** *Consider the theory of reals with multiplication,* $\mathrm{Th}(\mathsf{R}, \cdot, <, 0, 1)$*. Give a quantifier–elimination procedure and analyse its complexity. (*HINT*: Augment the language to have terms of the form* $\prod_i x_i^{\frac{a_i}{b_i}} := x_1^{\frac{a_1}{b_1}} \cdots \cdot x_n^{\frac{a_n}{b_n}}$*. Then proceed similarly as above, replacing arithmetic means by geometric means.)*

*EXERCISE **33** *Consider the theory of* integer addition, $\mathrm{Th}(\mathsf{N}, +, <, 0, 1)$*, also called* **Presburger's Arithmetic***. Give a quantifier–elimination procedure for the theory and analyse its complexity. (*HINT*: Augment the language with symbols for each positive integer. Proceed as above. Care is required in "solving for" x: one must avoid divisions. With this in view, introduce appropriate least common multiples instead. Also care is needed in defining the formulas* $\psi_{u,v}$*. See [18].)*

4.2 An EF Game

In this section, we shall define an EF–game and employ it to show that the structure $(\mathsf{R}, +, <, 0, 1)$ is H–bounded, for suitable H. In this fashion, we shall obtain a decision procedure which will enable us to get a rather precise upper bound on $\mathrm{Th}((\mathsf{R}, +, <, 0, 1)$. While roughly of the same time complexity as the procedure of the previous section, it will be more efficient in the space requirement.

Define the relations $\sim_{n,k}$ between the k–pointed algebras $(\mathsf{R}, \mathbf{a}_k)$ and $(\mathsf{R}, \mathbf{b}_k)$ as follows. (We shall abbreviate $(\mathsf{R}, \mathbf{a}_k) \sim_{n,k} (\mathsf{R}, \mathbf{b}_k)$ by $\mathbf{a}_k \sim_n \mathbf{b}_k$ and similarly for $\equiv_{n,k}$.)

$$\mathbf{a}_k \sim_n \mathbf{b}_k \quad \text{iff} \quad v + \sum_{1 \le i \le k} v_i \cdot a_i \quad R \quad 0 \leftrightarrow v + \sum_{1 \le i \le k} v_i \cdot b_i \quad R \quad 0,$$

where R is $=$ or $<$ and where $v, v_1, \cdots, v_k \in V_n$ with the sets V_n defined inductively via:

$$V_0 := \{-1, 0, 1\}$$

and for $i \ge 0$,

$$V_i' := \{v/v' \mid v, v' \in V_i, v' \neq 0\}$$

$$V_{i+1} := \{u + v \mid u, v \in V_i'\}.$$

It is easy to check that $\mathbf{a}_k \sim_0 \mathbf{b}_k$ implies that $\mathbf{a}_k \equiv_0 \mathbf{b}_k$. Suppose for some $n \ge 0$ that $\mathbf{a}_k \sim_{n+1} \mathbf{b}_k$, with $\|b_i\| \le m$ for each $i \in [k]$.and let $a_{k+1} \in \mathsf{R}$ be arbitrary. We shall find a suitable b_{k+1} of "small" norm such that $\mathbf{a}_{k+1} \sim_n \mathbf{b}_{k+1}$. Consider all terms

$$t(\mathbf{x}_k) := v + \sum_{1 \le i \le k} v_i x_i,$$

such that $v, v_1, \ldots, v_k \in V_n'$.

Lemma 34 *We have for any two terms* t_1, t_2,

$$t_1(\mathbf{a}_k) \quad R \quad t_2(\mathbf{a}_k) \qquad \textit{iff} \qquad t_1(\mathbf{b}_k) \quad R \quad t_2(\mathbf{b}_k).$$

Proof. We have $t_1(\mathbf{a}_k) \quad R \quad t_2(\mathbf{a}_k)$ iff $t_1(\mathbf{a}_k) - t_2(\mathbf{a}_k) \quad R \quad 0$, which, using the fact that $\mathbf{a}_k \sim_{n+1} \mathbf{b}_k$(note that the terms $t_1 - t_2$ have coefficients in V_{n+1}) happens iff $t_1(\mathbf{b}_k) - t_2(\mathbf{b}_k) \quad R \quad 0$ which, in turn is iff $t_1(\mathbf{b}_k) \quad R \quad t_2(\mathbf{b}_k)$.. ∎

Hence, we can number the terms $t_1, t_2, \ldots, t_r$ for some $r > 0$ so that

$$t_1(\mathbf{a}_k) \leq t_2(\mathbf{a}_k) \leq \cdots \leq t_r(\mathbf{a}_k),$$

and

$$t_1(\mathbf{b}_k) \leq t_2(\mathbf{b}_k) \leq \cdots \leq t_r(\mathbf{b}_k),$$

Exactly one of the following cases obtains:

1. $a_{k+1} < t_1(\mathbf{a}_k)$.

2. $a_{k+1} > t_r(\mathbf{a}_k)$.

3. For some $i \geq 1$, $a_{k+1} = t_i(\mathbf{a}_k)$.

4. For some $1 \leq i < r$, $t_i(\mathbf{a}_k) < a_{k+1} < t_{i+1}(\mathbf{a}_k)$.

Respectively, define b_{k+1} to be

1. $t_1(\mathbf{b}_k) - 1$.

2. $t_r(\mathbf{b}_k) = 1$.

3. $t_i(\mathbf{b}_k)$.

4. $(t_i(\mathbf{b}_k) + t_{i+1}(\mathbf{b}_k))/2$.

We assert that $\mathbf{a}_{k+1} \sim_n \mathbf{b}_{k+1}$. For, take a relation $v + \sum_{1 \leq i \leq k+1} v_i a_i \quad R \quad 0$ with $v_i \in V_n$ for $1 \leq i \leq k+1$. By "solving for" a_{k+1}, this happens iff $a_{k+1} \quad R \quad u + \sum_{1 \leq i \leq k} u_i a_i$ for $u, u_1, \ldots, u_k \in V_n'$. By the choice of b_{k+1}, we must have also $b_{k+1} \quad R \quad u + \sum_{1 \leq i \leq k} u_i b_i$, hence also $v + \sum_{1 \leq i \leq k+1} v_i b_i \quad R \quad 0$.

Let us estimate the size of this choice of b_{k+1}. For this we need

Lemma 35 *For each $n \geq 0$ and each element $v \in V_n$, we have $\|v\| \leq 2^{2^{cn}}$ for some $c > 0$.*

Proof. We prove this by induction on n. For $n = 0$ this is immediate. Assume the result for some $n \geq 0$. For any element $v' \in V_n'$, we have, using Lemma 29 $\|v'\| \leq (2^{2^{cn}})^2 \leq 2^{2^{c(n+1)}}$, provided $c > 1$. Applying Lemma 29 one more time, we conclude that for an element $u \in V_{n+1}$, $\|u\| \leq 2 \cdot (2^{2^{c(n+1)}})^2 \leq 2^{2^{c'(n+1)}}$ provided $c' > 4c$. ∎

We are now in a position to estimate $\|b_{k+1}\|$. To take a typical case, (the others are easier),

$$\begin{aligned}
\|b_{k+1}\| &= \|\frac{t_i(\mathbf{b}_k)+t_{i+1}(\mathbf{b}_k)}{2}\| \\
&\leq 2\cdot\|t_i(\mathbf{b}_k)+t_{i+1}(\mathbf{b}_k)\| \\
&\leq 2\cdot\|u+\sum_{1\leq i\leq k} u_i b_i\| \quad \text{with } u, u_i \in V_{n+1} \\
&\leq 2\cdot m\cdot\|u+\sum_{1\leq i\leq k} u_i\| \\
&\leq 2\cdot m\cdot(k+1)\cdot 2^{k\cdot 2^{cn}} \\
&\leq m\cdot 2^{2^{c'(n+k)}}
\end{aligned}$$

We conclude that the structure $(\mathsf{R},+,<,0,1)$ is $H(n,k,m) := m\cdot(k+1)\cdot 2^{2^{c'(n+k)}}$–bounded.

Theorem 36 *We have,*

$$\mathrm{Th}(\mathsf{R},+,<,0,1) \in \bigcup_{c>0} STA(*, 2^{cn}, n).$$

In particular $\mathrm{Th}(\mathsf{R},+,<,0,1) \in EXPSPACE$.

Proof. Define $m_0 := 0$ and for $i>0$, $m_i := 2^{2^{c'(n+i)}}$. It can be verified that $m_i \geq H(n-i, i-1, m_{i-1})$ for each $i>0$. Hence by the general results of Chapter 1, a sentence

$$Q_1x_1\ldots Q_nx_n\varphi(x_1,\ldots,x_n)$$

is equivalent to the corresponding sentence with limited quantifiers

$$(Q_1x_1\leq m_1)\ldots(Q_nx_n\leq m_n)\varphi(x_1,\ldots,x_n).$$

An alternating TM can decide such a sentence with n alternations, one for each quantifier. It can generate the m_i elements corresponding to the ith quantifier in parallel using the appropriate $\vee$ or $\wedge$ branching in time $\log m_i = 2^{c(n+i)}$. Hence the time is bounded overall by $n\cdot 2^{c'n}\leq 2^{c''n}$. ∎

Exercise **37 (Gerth Stølting Brodal)** *Consider the theory of reals with multiplication,* $\mathrm{Th}(\mathsf{R},\cdot,<,0,1)$. *Define an EF–game to show that the structure* $(\mathsf{R},\cdot,<,0,1)$ *is H–bounded for suitable H. Give a decision procedure based on this and analyse its complexity. (*Hint*: Define*

$$\mathbf{a}_k \sim_n \mathbf{b}_k \quad \textit{iff} \quad \prod_i a_i^{v_i} \quad R \quad 1 \leftrightarrow \prod_i b_i^{v_i} \quad R \quad 1,$$

where R is $=$ or $<$ and where $v, v_1, \ldots, v_k \in V_n$. Then proceed as before replacing arithmetic means by geometric means.)

*EXERCISE **38** *In this exercise, we shall consider again, the theory of the integers,* Th(N). *See Exercise 33 above.*

1. *Argue that* $\mathrm{Th}(\mathsf{N}, +, \cdot, <, 0, 1)$ *is undecidable.*

2. *What is wrong with the following informal argument: "Since the integers are a special case of the reals,* $\mathrm{Th}(\mathsf{N}, +, <, 0, 1)$ *is not harder than* $\mathrm{Th}(\mathsf{R}, +, <, 0, 1)$*."*

3. *Define an appropriate EF–game to show that the structure* $(\mathsf{N}, +, <, 0, 1)$ *is* $H(n.k, m) := m \cdot 2^{2^{2^{c(n+k)}}}$ *– bounded. Give a decision procedure for the theory* $\mathrm{Th}(\mathsf{N}, +, <, 0, 1)$ *and analyse its complexity.*

4.3 Lower Bound

In this section, we will sketch a proof of the following matching lower bound:

Theorem 39 *The theory of real addition,* $\mathrm{Th}(\mathsf{R}, +, <, 0, 1)$ *is hard for the alternating class* $\bigcup_{c>0} STA(*, 2^{cn}, n)$ *under* $\leq_{\mathrm{poly}}$ *reductions.*

The basic spirit of this result is the same as that of the fundamental theorem of Cook that *satisfiability of boolean formulas* is NP–complete. For an exposition of Cook's Theorem, see [19, pp. 171–172] or [16, Lecture 25], or § 5.3 where it is essentially reproved. The idea is to describe computations of Turing machines by formulae, in this case, in the language of real addition. However, a simple minded strategy along those lines is doomed to fail, since we cannot afford to introduce a separate variable to describe each configuration – that would immediately involve an exponential blow–up since our TMs here run for exponential time!

Rather, *we describe the entire computation by a single variable*! Suppose for a moment, that we have a non–deterministic TM, M at hand (not an ATM), and suppose it accepts an input x, $|x| = n$ in time bounded by $N := n^k$ for a fixed k. Then we can imagine laying out an accepting computation as a tableau of size $N \times N$. Regarded as a number written in some appropriate base, for simplicity, say 2, this tableau is a number of size $2^{N^2} < 2^{2^{cn}}$. Suppose we are now able to write down formulas in the theory of reals that enable us to make statements about such numbers, in particular to extract the bits in various positions. Then, we will be able to mimic the proof of Cook's Theorem as before. A potential problem is that there are too many bits to manipulate as well, but we shall get around that by using universal quantification.

We shall now describe how to produce short formulae in the language of real addition that allow us to define integers in the range we want (double exponential) and multiplication and exponentiation on such numbers. Thus even in the theory of real addition, we will have the power of limited multiplication and exponentiation.

Lemma 40 *There is a* $c > 0$ *such that for every non–negative integer* n*, there is a formula* $M_n(x, y, z)$ *such that*

$$(\mathsf{R}, a, b, c) \models M_n(x, y, z),$$

iff

- *a is an non–negative integer strictly less than 2^{2^n}.*
- *$ab = c$.*

Moreover, the size of M_n is at most $c(n+1)$ and M_n can be constructed from n in time polynomial in n.

Proof. We proceed by induction. For $n = 0$, we define

$$M_0(x, y, z) := (x = 0 \wedge z = 0) \vee (x = 1 \wedge y = z).$$

It is easy to see that M_0 satisfies the asserted properties. For the induction step, suppose we have constructed M_k for some $k \geq 0$. let a be an integer, $a < 2^{2^{k+1}}$. There exist non–negative integers a_1, a_2, a_3, a_4 such that $a = a_1a_2 + a_3 + a_4$. (Take $a_1 := \lfloor\sqrt{a}\rfloor =: a_2$). Then $ab = c$ iff $(a_1b)a_2 + a_3b + a_4b = c$. Thus we are led to write

$$\begin{array}{rcl} M_{k+1}(x, y, z) & := & \exists \mathbf{u}_5 \exists \mathbf{x}_5 \quad (M_k(x_1, x_2, u) \\ & \wedge & x = u_1 + x_3 + x_4 \\ & \wedge & M_k(x_2, y, u_2) \wedge M_k(x_1, u_2, u_3) \wedge M_k(x_3, y, u_4) \wedge M_k(x_4, y, u_5) \\ & \wedge & z = u_3 + u_4 + u_5). \end{array}$$

Notice how the condition that each x_i is an integer less than 2^{2^k} is enforced inductively by each x_i being the first argument of some M_k. The semantics of this definition are all right, but the problem is that the formula M_{k+1} is at least five times as long as M_k. This would be catastrophic, since it would imply that the formulae M_n grow exponentially in size! There is a nifty trick that solves the problem. We replace all five occurrences of M_k with just one via the following sleight–of–hand:

$$\begin{array}{rcl} M_{k+1}(x, y, z) & := & \exists \mathbf{u}_5 \exists \mathbf{x}_5 \quad (x = u_1 + x_3 + x_4 \wedge z = u_3 + u_4 + u_5 \\ & \wedge & \forall \mathbf{r}_3(((r_1 = x_1 \wedge r_2 = x_2 \wedge r_3 = u_1) \vee (r_1 = x_2 \wedge r_2 = y \wedge r_3 = u_2) \\ & & \vee(r_1 = x_3 \wedge r_2 = y \wedge r_3 = u_4) \vee (r_1 = x_4 \wedge r_2 = y \wedge r_3 = u_5)\vee) \\ & \rightarrow & M_k(r_1, r_2, r_3))). \end{array}$$

With this definition, $|M_{k+1}| = |M_k| + c'$ for some constant $c' > 0$, and hence the result. ∎

Note that $(\mathsf{R}, a) \models M_n(x, 0, 0)$ iff a is a non–negative integer less than 2^{2^n}. Thus the previous proposition enables us to define small integers and multiply them in the language of real addition.

Similarly we can define limited exponentiation on such small integers.

Lemma 41 *There is a constant $c > 0$ such that for all $n \geq 0$, there is a formula $P_n(x, y, z)$ such that $(\mathsf{R}, a, b, c) \models P_n(x, y, z)$ iff*

- *a and c are non–negative integers less than 2^{2^n}.*
- *$c = b^a$.*

Further $|P_n| \leq c \cdot (n+1)^2$, and P_n can be constructed from n in polynomial time.

Proof. We proceed as before by induction on n. For $n = 0$, the formula is:

$$P_0(x, y, z) := (x = 0 \wedge z = 1) \vee (x = 1 \wedge y = 0 \wedge z = 0) \vee (x = 0 \wedge y = 1 \wedge z = 1).$$

For the induction step, we note that if a be an integer, $a < 2^{2^{k+1}}$, then there exist non–negative integers a_1, a_2, a_3, a_4 such that $a = a_1 a_2 + a_3 + a_4$. Hence $b^a = (b^{a_1})^{a_2} \cdot b^{a_3} \cdot b^{a_4}$. We can express the sub–exponentiations using P_k (which we assume inductively to be already at hand) and then we express the multiplications using M_{k+1}. The condition on c can be expressed via $M_{k+1}(z, 0, 0)$. Now, using the same trick as in the previous lemma, we eliminate all but one occurrences of P_k in this formula using universal quantification. The resulting formula has the properties asserted; moreover for the size, it is easy to see that

$$|P_{k+1}| \leq |P_k| + c'(k+1),$$

for some constant $c' > 0$ (we used the fact that $|M_{k+1}| \leq c''(k+2)$). Hence we conclude also that $|P_k| \leq c(k+1)^2$ for some $c > 0$. ∎

Exercise **42** *Show how to write a formula equivalent to P_k but with size at most $c(k+1)$ for some $c > 0$. (See, for example, Hopcroft and Ullman [13, p.359]).*

We shall now use abbreviations like 2^n and ab in the formulae below. This is justified, since we can use the lemmas above to introduce a existentially quantified variable and write down formulae formulae $P_n(2, n, x)$ or $M_n(a, b, x)$ to enforce these conditions.

We shall want to code a ATM configuration by an integer. We shall make some simplifying assumptions about our ATM M. We shall assume that if it accepts, it always does so in a unique state with the tape all blank. Further, we shall assume that the tuples $(q, a, q, a, 0)$ are always present, so that the machine can linger for any number of steps in the same configuration. These assumptions are without loss of generality, since any ATM can be easily modified to behave so. As to the coding itself, the contents of the tape can be directly regarded as a number written in a suitable base (say binary, if the tape alphabet is binary). Since our machine runs for at most 2^n steps on an input of size n, there are at most 2^n tape cells that are ever scanned, so the head position is an integer between 1 and 2^n. Finally, fixing a coding of the finite state set and state labels, and agreeing on a fixed pairing function, we can represent the entire configuration by an integer in the range $1 \ldots 2^{cn}$, for some fixed constant $c > 0$.

Wishing to avoid getting cluttered with the details of the coding, we shall assume that we are in the possession of the following formulas for each $n \geq 0$:

- $D_n(x, i, j)$ such that
$$(\mathsf{R}, a, b, 0/1) \models D_n(x, i, j),$$
iff a is a non–negative integer less than 2^{2^n}, b is a non–negative integer less than 2^n and the bth bit of a is $0/1$.

- $CONFIG_n(x)$ such that $(\mathsf{R}, a) \models CONFIG_n(x)$ iff a codes a possible configuration of M on an input of size n.
- $H_n(x, y)$ such that $(\mathsf{R}, a, b) \models H_n(x, y)$ iff a is a possible configuration of M on an input of size n in which the head is at position b which is a non–negative integer of size at most 2^n.
- For each $q \in Q_M$, a formula $STATE_n^q(x)$ such that $(\mathsf{R}a) \models STATE_n^q(x)$ iff a is the code of a configuration of M on input of size n, in which the machine is in state q.
- $EXISTS_n(x), (FORALL_n(x))$ such that $(\mathsf{R}, a) \models EXISTS_n(x)$ $((\mathsf{R}, a) \models FORALL_n(x))$ iff a is a non–negative integer of size at most 2^n coding an existential (universal) configuration.

Informally, we can see that it is possible to write such formulae given the ability to multiply and exponentiate integers in the relevant range (see [13, p.360]).

We are now in a position to mimic the proof of Cook's Theorem, albeit in a more sophisticated way. We shall write down a formula $TRANSITION_n(x, y)$ such that

$$(\mathsf{R}, a, b) \models TRANSITION_n(x, y),$$

iff a and b describe configurations of the ATM M on input of size at most n, such that there is a computation sequence of M starting at a and ending at b. The formula is

$$TRANSITION_n(x, y) := \exists z \quad COMPUTATION_n(x, y, z),$$

where

$$(\mathsf{R}, a, b, c) \models COMPUTATION_n(x, y, z),$$

iff c is a real that codes the entire computation sequence on an input of size at most n, starting with the configuration described by a and ending with the configuration described by b. The formula $COMPUTATION_n$ is built up as a conjunction of the following sub–formulae (with free variables x, y, z):

1. (*Start:*) The first configuration is a:

   $$\forall i (1 \le i \le 2^n \to D_n(x, i, 0) \leftrightarrow D_n(z, i, 0)).$$

2. (*Final:*) The final configuration is b. This is similar to the above.

3. (*Transition:*) Successive configurations are related by the transition relation of M. The following two formulae in conjunction achieve this. The first says that a tape square not being scanned remains unchanged, and the second that a scanned cell changes in accordance with the transition relation of M (in the latter formula, x' and x'' are variables introduced to represent the configurations at the ith and $i+1$st steps in the computation):

   $$\forall i \forall j (1 \le i, j \le 2^n \wedge \neg H_n(i, j) \to (D_n(\gamma, i + j, 0) \leftrightarrow D_n(\gamma, i + 2^n + j, 0))).$$

$$
\begin{aligned}
\forall i \exists x' x'' \forall j (1 \le i, j \le 2^n & \\
& \wedge \quad D_n(x', j, 0) \leftrightarrow D_n(z, i+j, 0)) \\
& \wedge \quad D_n(x'', j, 0) \leftrightarrow D_n(z, i+2^n+j, 0) \\
& \wedge \quad H_n(x', j) \wedge D_n(x', j, a) \wedge STATE_q(x') \\
& \rightarrow \quad \bigvee_{(q,a,q',b,d)\in\delta_M} (D_n(x'', j, b) \\
& \wedge \quad STATE_n^{q'}(x'') \wedge D_n(x'', j, b) \wedge H_n(x'', j+d)).
\end{aligned}
$$

We now write a sequence of formulae $ACCEPT_{n,k}(\alpha)$ for $k \ge 0$, such that

$$(\mathsf{R}, a) \models ACCEPT_{n,k}(x),$$

iff a codes the description of a configuration on input of size n, which is the root of an accepting computation sub–tree of M of alternation depth at most k. For $k = 0$, there are no alternations and so $ACCEPT_{n,0}(x)$ is the formula:

$$
\begin{aligned}
&(EXISTS_n(x) \rightarrow \exists y \quad TRANSITION_n(x, y) \wedge FINAL_n(y)) \vee \\
&\quad (FORALL(y) \rightarrow \forall y \quad TRANSITION_n(x, y) \wedge FINAL_n(y))
\end{aligned}
$$

Here, $FINAL_n(y)$ is the formula such that

$$(\mathsf{R}, a) \models FINAL(y),$$

iff a codes the unique accepting configuration on input of length n,

Inductively, assuming that we have constructed $ACCEPT_{n,k}$ for some $k \ge 0$, we form $ACCEPT_{n,k+1}(x)$:

$$
\begin{aligned}
&(EXISTS_n(x) \rightarrow \exists y \quad TRANSITION_n(x, y) \wedge FORALL_n(y) \wedge ACCEPT_{n,k}(y)) \vee \\
&\quad FORALL_n(x) \rightarrow \forall y \quad TRANSITION_n(x, y) \wedge EXISTS_n(y) \wedge ACCEPT_{n,k}(y))
\end{aligned}
$$

Finally, given the input x_0, we can , in a straightforward fashion, construct a formula $START_{x_0}(x)$ such that

$$(\mathsf{R}, a) \models START_{x_0}(x),$$

iff a describes the start configuration of M on input x_0. Our final formula then, is

$$\psi_{M,x_0} := \exists x (START_{x_0}(x) \wedge ACCEPT_{|x_0|,|x_0|}(x)).$$

It is easy to check that these formulas have the properties asserted. Since our machine M accepts an input x_0 in time at most $2^{c|x_0|}$ with at most n alternations,

$$\mathsf{R} \models \psi_{M,x_0} \quad \text{iff } M \text{ accepts } x_0.$$

The size of the formula $ACCEPT_{n,k}$ is, roughly, k times the size of the formula $TRANSITION_n$. The formula $TRANSITION_n$ has size bounded by a polynomial in n, as in Cook's Theorem. Hence, the formula $ACCEPT_{n,n}$ will also have size a bounded by a polynomial in n. Moreover, it can also be constructed in polynomial time.

EXERCISE **43 (Gerth Stølting Brodal)** *Show that* $\mathrm{Th}(\mathsf{R}, \cdot, <, 0, 1)$ *is* $\le_{\log}$*–hard for the class* $\bigcup_{c>0} STA(*, 2^{cn}, n)$. *Deduce together with Exercise 37 that it is also complete for this class.* (HINT*: Give a reduction from the theory of real addition.)*

4.4 A Research Problem

What happens if we take the reals with both addition and multiplication together? This is the classical problem of Alfred Tarski who showed that $\mathrm{Th}(\mathsf{R}, +, \cdot, <, 0, 1)$ was decidable [21] (contrast with the theory of the integers). Subsequently, many algorithms have been given for this decision problem. Some landmark results are:

- The first quantitative complexity result was by G.E. Collins [5] who developed the influential method of *cylindrical algebraic decomposition*. The method ran in double exponential time.

- A beautiful seminal paper by Ben–Or, Kozen and Reif [17] opened up the modern approaches to the problem. In it, they generalised the classical theorem of Sturm to count the number of roots of a single variable polynomial and developed an elegant theory of representation of reals by sign sequences. They showed that the decision problem could be solved in exponential space.

- The most recent developments gaining various improvements in time over the previous algorithms are those in [3, 10, 20].

Research Problem 44 *Define a suitable norm on the reals, a suitable Ehrenfeucht–Fraïssé game and show that the structure* $(\mathsf{R}, +, \cdot, <, 0, 1)$ *is H–bounded for a suitable H.*

A reasonable EF–game would be to define $\mathbf{a} \sim_n \mathbf{b}$ iff $p(\mathbf{a})R0 \leftrightarrow p(\mathbf{b})R0$ for all (multivariate) polynomials with coefficients in certain subsets of reals (where R is $=$ or $<$). The main problem is that one has immediately to contend with *real algebraic* numbers, namely those that are defined (implicitly) as roots of polynomial equations. One must define norms on and manipulate such numbers. Relevant would be the growing research on representations of reals by sign sequences via generalisations of Sturm's Theorem in higher dimensions [7]. A plausible and attractive conjecture is that $\mathrm{Th}(\mathsf{R}, +, \cdot, <, 0, 1)$ is also complete for $\bigcup_{c>0} STA(*, 2^{cn}, n)$.

4.5 Bibliographical Notes

The quantifier–elimination method for the theory of real addition is due to Ferrante and Rackoff [8] (although the basic idea is due to Oppen [18]). The lower bound is due to Fisher and Rabin – this is where the "formula tricks" originate.

Chapter 5

Boolean Algebras

In this chapter, we apply our general techniques to our second example, *the theory of Boolean Algebras*, a class of structures also of fundamental importance in Mathematics and Computer Science.

5.1 The Structure of Boolean Algebras

A *Boolean algebra* is a structure $\mathsf{B} := (B, +, \cdot, \overline{()}, 0, 1)$ such that the following set of axioms hold for all $x, y, z \in B$ (for brevity, we write xy to mean $x \cdot y$),

$$\begin{aligned} (x+y)+z &= x+(y+z) & \qquad (xy)z &= x(yz) \\ x+y &= y+x & xy &= yx \\ x(y+z) &= xy+yz & x+yz &= (x+y)(y+z) \\ x+0 &= x & x0 &= 0 \\ x+1 &= 1 & x1 &= x \\ x+\overline{x} &= 1 & x\overline{x} &= 0 \\ x+x &= x & xx &= x \\ 0 &\neq 1 \end{aligned}$$

EXERCISE **45** *Show that the following* De Morgan's laws *are a consequence of these axioms: for very $x, y, z \in B$,*

$$\overline{x+y} = \overline{xy} \qquad \overline{xy} = \overline{x} + \overline{y}.$$

We can define a natural order on a Boolean algebra by stipulating that for $b, b' \in B$,

$$b \leq b' \leftrightarrow bb' = b.$$

EXERCISE **46** *Check that $(B, \leq)$ is a partially–ordered structure.*

The smallest Boolean algebra is the Boolean algebra, 2 whose underlying set has two elements $1, 0$ with $\overline{0} = 1, \overline{1} = 0$. In other contexts, these elements are denoted by $\mathsf{true}, \mathsf{false}$ (logic) or $\top, \bot$ (lattice theory). A broad class of examples is afforded by the following familiar construction: Let X be an arbitrary set, and let $\mathcal{P}(X)$ denote the *power set* of X. Then the structure $\mathcal{P}(\mathsf{X}) := (\mathcal{P}(X), \cup, \cap, \overline{()}, \emptyset, X)$ is a Boolean algebra. For $A \in \mathcal{P}(X)$, by $\overline{A}$, we mean $X \setminus A$. Such a Boolean algebra is called an *algebra of sets*. We shall see below, that these are essentially the *only* kinds of Boolean algebras in a certain class.

Let B be a boolean algebra. An *atom* in B is an element $b \in B$ such that $b \neq 0$ and if $0 \leq b' \leq b$, then $b' = b$ or $b' = 0$. We denote the set of atoms of B by B^a. B is called *atomic* if for every $b \in B$, there exists an atom $b' \in B^a$ such that $b' \leq b$. For an element $b \in \mathsf{B}$ we use the notation $b \downarrow := \{b' \in B^a \mid b' \leq b\}$. B is *complete* if for every $A \subseteq B$, there exists a *join*, namely an element $\sum A$ such that

1. $a \leq \sum A$ for each $a \in A$.

2. If $b \in B$ is such that $a \leq b$ for all $a \in A$, then $\sum A \leq b$ as well.

Moreover, the distributive law must extend to

$$b \cdot \sum A = \sum \{ba \mid a \in A\}.$$

Note that a finite Boolean algebra is both atomic and complete. Complete, atomic Boolean algebras have a definite structure:

Theorem 47 (Representation Theorem) *Every complete, atomic Boolean algebra is isomorphic to an algebra of sets.*

Proof. Given a complete atomic Boolean algebra, B, take X_B to be the set of atoms in B. Now define a mapping $f : \mathsf{B} \to \mathcal{P}(\mathsf{X_B})$ by setting, for $b \in B$,

$$f(b) := b \downarrow = \{b' \in B^a \mid b' \leq b\}.$$

First we check that this map is a homomorphism of Boolean algebras. For $b_1, b_2 \in B$, we have,

$$\begin{aligned} f(b_1 + b_2) &:= \{b' \in B^a \mid b' \leq b_1 + b_2\} \\ &= \{b' \in B^a \mid b' \leq b_1 \text{ or } b' \leq b_2\} \qquad (5.1) \\ &= \{b' \in B^a \mid b' \leq b_1\} \cup \{b' \in B^a \mid b' \leq b_2\} \\ &= f(b_1) \cup f(b_2). \end{aligned}$$

In the asserted equality (5.1), the inclusion $\{b' \in B^a \mid b' \leq b_1 \text{ or } b' \leq b_2\} \subseteq \{b' \in B^a \mid b' \leq b_1 + b_2\}$ is clear. For the reverse inclusion, note that if $b'b_1 \neq b'$, then since b' is an atom, $b'b_1 = 0$, and similarly for $b'b_2$. But then $b'(b_1 + b_2) = 0$ which contradicts $b' \leq b_1 + b_2$.

Similarly, we check that

$$f(b_1 b_2) = f(b_1) \cap f(b_2),$$

and

$$f(\overline{b_1}) = X_B \setminus f(b_1).$$

Next to check that f is injective, we merely note that $\mathrm{Ker} f = \{0\}$. Here we use the fact that B is atomic.

Finally we argue that f is surjective; in fact we assert that for any $A \subseteq X_B$, we have $A = f(\sum A)$ – here we use the fact that B is complete. Clearly $A \subseteq f(\sum A)$. For the reverse inclusion, suppose $b \leq \sum A$ for some atom b. If $ba = 0$ for all $a \in A$, then also

$$b = b \cdot \sum A = \sum\{ba \mid a \in A\} = 0,$$

which contradicts the fact that b is an atom. Hence for some $a \in A$, we have $ba = a$. But then since b is an atom, also $b = a$. ∎

Applied to finite algebras, the Representation Theorem shows that for every positive integer n, there is, upto isomorphism, a unique Boolean algebra with cardinality 2^n, namely, $\mathcal{P}([\mathsf{n}])$. This algebra has exactly n atoms.

Unfortunately not all Boolean algebras have this form. A Boolean algebra is said to be *atomless* if it has no atoms. An example of such an algebra is the Boolean algebra generated by all sub–intervals $[a, b)$ of the unit interval $[0, 1)$. Any element in this algebra is a finite union of such sub–intervals, the boolean algebra operations are $\cup$, $\cap$ and set complement (relative to $[0, 1)$), $0 := [0, 0) = \emptyset$ and $1 := [0, 1)$. This Boolean algebra has no atoms. There is a much more sophisticated Representation Theorem that is valid for such algebras as well, the famous *Stone Representation Theorem*, [12, § 18].

Below, we shall restrict ourselves to the study of the sub–class of atomic algebras, which we denote by $\mathcal{BA}$.

Finally we introduce some notation and definitions about sub–algebras of a Boolean algebra. Let $\mathsf{B} := (B, +, \cdot, \overline{()}, 0, 1)$ be a Boolean algebra.

A subset $A \subseteq B$ is said to *partition* an element $a \in B$ if $a = \sum A$ and $bc = 0$ for distinct $b, c \in A$. We say that A partitions B if it partitions 1. A *bit term* on k variables is a term (in the language of Boolean algebras)of the form $t(x_1, x_2, \ldots, x_k) := y_1 \cdot y_2 \cdots y_k$ where, for each $i \in [k]$, we have $y_i \in \{x_i, \overline{x_i}\}$. There are exactly 2^k distinct bit terms on k variables. For a vector of elements $\mathbf{a}_k \in B^k$, a *bit* of $\mathbf{a}_k$ is an element $t(\mathbf{a}_k)$ for a bit term t (i.e. the standard interpretation of the bit term t in $(\mathsf{B}, \mathbf{a}_k)$). (By convention, the empty set has one bit, namely 1.). Denote by $B(\mathbf{a}_k) := \{t(\mathbf{a}_k)\}_t$, the set of all bits of $\mathbf{a}_k$.

Proposition 48 *Let* $\mathbf{a}_k \in B^k$.

1. $B(\mathbf{a}_k)$ *partitions* B.
2. *Let* $t(\mathbf{x}_k), t'(\mathbf{x}_k)$ *be distinct bit terms. Then*

$$t(\mathbf{a_k}) \downarrow \cap t'(\mathbf{a}_k) \downarrow = \emptyset.$$

EXERCISE **49** *Verify these assertions.*

5.2 Decision Procedures

Following the strategy outlined in Chapter 1, our first task is to restrict attention to "small" algebras. We define the norm of a Boolean algebra to be its cardinality (∞ for non–finite algebras). The following is the basic result showing that the class $\mathcal{BA}$ is suitably bounded.

Theorem 50 *For every $n \geq 0$ and for every $\mathsf{B} \in \mathcal{BA}$, there exists a Boolean algebra, B' such that $\mathsf{B} \equiv_{n,0} \mathsf{B}'$, and moreover, $\|\mathsf{B}'\| \leq 2^{2^n}$. Hence the class $\mathcal{BA}$ is $h(n) := 2^{2^n}$–bounded.*

Proof. We use Proposition 26 and Proposition 23. For non–negative integers n, k, define a relation $\sim_{n,k}$ on k–pointed Boolean algebras by stipulating that

$$(\mathsf{A}, \mathbf{a}_k) \sim_{n,k} (\mathsf{B}, \mathbf{b}_k)$$

iff for each bit term $t(x_1, \cdots, x_k)$, and the corresponding bits $t(\mathbf{a}_k) \in A$ and $t(\mathbf{b}_k) \in B$,

$$\min(2^n, |t(\mathbf{a}_k) \downarrow |) = \min(2^n, |t(\mathbf{b}_k) \downarrow |).$$

In words, for every two corresponding bits a and b of $\mathbf{a}_k$ and $\mathbf{b}_k$ respectively, the number of atoms below a and b are the same upto a maximum of 2^n.

It is easy to check that $(\mathsf{A}, \mathbf{a}_k) \sim_{0,k} (\mathsf{B}, \mathbf{b}_k)$ implies that the map $a_i \mapsto b_i, i \in [k]$ is a partial isomporphism. For example, suppose $a_1 + a_2 = a_3$. Then for each bit term $t(x_1, x_2, x_3, \ldots, x_k) := x_1 \cdot x_2 \cdot \overline{x_3} \cdots$, we have $t(\mathbf{a}_k) = 0$. This of course implies that $|t(\mathbf{a}_k) \downarrow | = 0$. Since $(\mathsf{A}, \mathbf{a}_k) \sim_{0,k} (\mathsf{B}, \mathbf{b}_k)$, also $|t(\mathbf{b}_k) \downarrow | = 0$. But since our algebras are atomic, this can only happen if in fact $t(\mathbf{b}_k) = 0$ for each such term. It follows that $b_1 + b_2 = b_3$. The fact that we are considering atomic algebras is crucial for this conclusion.

The assumption of atomicity is also crucial for the assertion that the map is injective. To see this, let us look at the kernel. We assert that if $b_i = 0$ for some i, then in fact $a_i = 0$ also. Indeed, all the k bit terms of the form $t(x_1, \ldots, x_i, \ldots, x_k) = x_i \cdot t'(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_k)$ satisfy $t(\mathbf{b}_k) = 0$. Hence also $|t(\mathbf{a}_k) \downarrow | = 0$. This is only possible if $t(\mathbf{a}_k) = 0$ for all such terms. This implies that $a_i = 0$, since the bits $a_i \cdot B(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_k)$ partition a_i.

Next, suppose $(\mathsf{A}, \mathbf{a}_k) \sim_{n,k} (\mathsf{B}, \mathbf{b}_k)$, and let $a_{k+1} \in A$ be arbitrary. We would like to find a $b_{k+1} \in B$ such that $(\mathsf{A}, \mathbf{a}_{k+1}) \sim_{n-1,k+1} (\mathsf{B}, \mathbf{b}_{k+1})$. We use the simple combinatorial fact in Exercise 51 following this proof. Consider a bit term $t(x_1, \cdots, x_k)$. In the exercise, set $m := 2^{n-1}$, and take $A := t(\mathbf{a}_k) \downarrow$ and $B := t(\mathbf{b}_k) \downarrow$. Let $A' \subseteq A$ be the set $t(\mathbf{a}_k) \cdot a_{k+1} \downarrow$. Denote the corresponding set $B' \subseteq B$ by B_t. (Note that the sets B_t for different terms t are disjoint.) Define $b_{k+1} := \sum \bigcup_t B_t$ where the union is over all bit terms on k variables. By construction we have $(\mathsf{A}, \mathbf{a}_{k+1}) \sim_{n-1,k+1} (\mathsf{B}, \mathbf{b}_{k+1})$.

By Proposition 23, we conclude that the relation $\sim_{n,k}$ refines $\equiv_{n,k}$ for all $n, k \geq 0$. In particular, $\mathsf{A} \sim_{n,0} \mathsf{B}$ implies that $\mathsf{A} \equiv_{n,0} \mathsf{B}$. For an arbitrary (atomic) Boolean algebra B with $\|\mathsf{B}\| \geq 2^{2^n}$, we have that $\mathsf{B} \sim_{n,0} \mathcal{P}([2^n])$, hence also $\mathsf{B} \equiv_{n,0} \mathcal{P}([2^n])$. Therefore, the class of (atomic) Boolean algebras $\mathcal{BA}$ is 2^{2^n}–bounded. ∎

EXERCISE **51** *Let m be a positive integer, and let A and B be two sets such that*

$$\min(2m, |A|) = \min(2m, |B|).$$

Then, for any subset $A' \subseteq A$, show that there exists a subset $B' \subseteq B$ such that

$$\min(m, |A'|) = \min(m, |B'|),$$

and

$$\min(m, |A| - |A'|) = \min(m, |B| - |B'|).$$

*(*HINT*:* For each $m \geq 0$, define an equivalence relation on sets by setting $A \sim_m B$ iff $\min(2^m, |A|) = \min(2^m, |B|)$. Observe that $A \sim_m B$ iff $|A| = |B|$ or both A, B have size at least 2^m. Suppose $A \sim_m B$. Let $A' \subseteq A$ be arbitrary, and suppose wlog that $|A'| \leq |\overline{A'}|$. If $|A| = |B|$, then simply pick $B' \subseteq B$ of size $|B'| = |A'|$. Do the easy verification that $A' \sim_{m-1} B'$ and $\overline{A'} \sim_{m-1} \overline{B'}$. If A, B both have at least 2^m, pick $B' \subseteq B$ of size $\min(2^{m-1}, |A'|)$. using the facts that $|A'| \leq |\overline{A'}|$ and $|A|, |B| \geq 2^m$ argue that once again, $A' \sim_{m-1} B'$ and $\overline{A'} \sim_{m-1} \overline{B'}$.*)*

The second part of our general strategy is trivial in this case: since we have arrived in the situation where we need consider only finite algebras, our sentence $Q_1 x_1 \cdots Q_k x_k \varphi(x_1, \ldots, x_k)$ is directly equivalent to a formula with limited quantifiers, namely where each variable ranges over all the finitely many elements of the algebra! Hence, we obtain easily,

Theorem 52

$$\mathrm{Th}(\textit{complete atomic algebras}) \in \bigcup_{c>0} \Pi STA(*, 2^{cn}, n),$$

or equivalently,

$$\mathrm{Sat}(\textit{complete atomic algebras}) \in \bigcup_{c>0} \Sigma STA(*, 2^{cn}, n).$$

Proof. By universal, or existential quantification respectively, we generate the integers $1 \ldots 2^n$, and thus implicitly, the Boolean algebras $\mathcal{P}([1]) \ldots \mathcal{P}([2^n])$ in time at most cn for some $c > 0$. Next we generate all possible values for each quantified variable, once again by the appropriate branching in time at most 2^{cn}. We can, for instance, represent elements in a Boolean algebra $\mathcal{P}(\mathsf{m})$ for $1 \leq m \leq 2^n$, by a bit vector of length m. All boolean operations can then be done bit–wise, each requiring time m. Hence the atomic formulas can be verified in time 2^{cn} for some constant $c > 0$. ∎

If we have finite algebras already at hand, the same observations show that

Proposition 53 *For a finite class of finite algebras $\mathcal{C}$,*

$$\mathrm{Th}(\mathcal{C}) \in \bigcup_{c>0} \Pi STA(*, cn, n).$$

In particular, $\mathrm{Th}(\mathcal{C}) \in PSPACE$.

5.3 Some Lower Bounds

We begin by showing a well–known result.

Theorem 54 Th(2) *is $PSPACE$–complete.*

Proof. Using proposition 53, it only remains to show that Th(2) is $PSPACE$–hard for $\leq_{\rm poly}$–reductions. Using the characterisation that $PSPACE = APTIME$, this is equivalent to showing that $APTIME \leq_{\rm poly} \mathrm{Th}(2)$.

The idea of the proof is to simulate the computation of an alternating Turing machine running in polynomial time by a formula interpreted over the Boolean algebra 2. This is in the same spirit as Cook's Theorem (see [19, pp. 171–172], or Lecture 25 of [16] on which our exposition is based). Let M be an alternating TM which accepts an input x (if at all), in time bounded by $N := |x|^k$. We shall make some simplifying assumptions about M. We assume that if M accepts an input, it does so by entering a unique accepting configuration, namely with a tape that is all blank except and in a unique distinguished final existential state q_f and the head on the leftmost cell. Further, we assume that the tuples $(q, a, q, a, 0)$ are always present, allowing the transition resulting in no action. We can make these assumption without loss of generailty, since an arbitrary ATM can be easily modified to behave so.

A configuration α in an accepting computation of such a machine can be described by a string over the tape alphabet Γ of length N, the position of the head, the state q and the value of the state label $g(q)$. This description can be given by a vector α of boolean i.e. 2–valued variables with the following decriptions and (intuitive) interpretations when the variables take the value 1:

- For each $1 \leq i \leq N$, and each $a \in \Gamma$, a variable $\alpha^S_{i,a}$. "The ith symbol on the tape is a."
- For each $1 \leq i \leq N$, a variable α^H_i. "The head is at position i."
- For each state $q \in Q$, a variable α^Q_q. "The state of the ATM is q."
- Variables $\alpha^\exists$ and $\alpha^\forall$. "The configuration is existential or universal respectively."

We shall write a formula $TRANSITION(\alpha, \beta)$ with the property that

$$(2, \mathbf{a}, \mathbf{b}) \models TRANSITION(\alpha, \beta),$$

iff the assignment of the values $\mathbf{a}$ to the variables α and $\mathbf{b}$ to the variables β decribe configurations related in the following way: there is a sequence of N configurations $\alpha =: \gamma_0, \gamma_1, \ldots, \gamma_N := \beta$ such that each $\gamma_i \vdash \gamma_{i+1}$, for $0 \leq i < N$.

The idea is once again to exactly mirror the construction in Cook's Theorem. We introduce N vectors of variables $\gamma_\mathbf{i}$ for $0 \leq i \leq N$ with each $\gamma_\mathbf{i}$ as above. The formula $TRANSITION(\alpha, \beta)$ will have the form:

$$\exists\gamma_\mathbf{0}\exists\gamma_\mathbf{1}\cdots\exists\gamma_\mathbf{N}\varphi(\alpha, \beta, \gamma_\mathbf{0}, \gamma_\mathbf{1}, \cdots, \gamma_\mathbf{N}).$$

The formula $\varphi(\alpha, \beta, \gamma_\mathbf{0}, \gamma_\mathbf{1}, \cdots, \gamma_\mathbf{N})$ will be the conjunction of the following sub–formulae all involving the free variables $\alpha, \beta, \gamma_\mathbf{0}, \gamma_\mathbf{1}, \cdots, \gamma_\mathbf{N}$:

- ($Start$): A formula φ_α which will assert that $\gamma_0 = \alpha$. This is easy:

$$\bigwedge_{i\in[N],a\in\Gamma} \gamma^S_{0,i,a} = \alpha^S_{i,a} \wedge \bigwedge_{i\in[N]} \gamma^H_{0,i} = \alpha^H_i \wedge \bigwedge_{q\in Q} \gamma^Q_{0,q} = \alpha^Q_q \wedge \gamma^\exists_0 = \alpha^\exists \wedge \gamma^\forall_0 = \alpha^\forall.$$

- ($Final$): A formula φ_β which will assert that $\gamma_N = \beta$. This is exactly similar to the one above, with β replacing α.

- ($Consistency$): A formula φ_c asserting various consistency requirements that must hold between the set of variables: namely the conjunction of each of the following:

  - At each time step, each cell of the tape must hold a unique symbol:

$$\bigwedge_{t\in[N],i\in[N]} (\bigvee_{a\in\Gamma} \gamma^S_{t,i,a} = 1 \wedge \bigvee_{a\neq a'\in\Gamma} \gamma^S_{t,i,a} = 0 \vee \gamma^S_{t,i,a'} = 0).$$

  - At each time step, the head must be in a unique position: This is similar to the above with the appropriate set of variables.
  - At each time step, the machine must be in a unique state: More of the same.
  - At each time step, the configuration must be either existential or universal, not both.

$$\bigwedge_{t\in[N]} (\gamma^\exists_t = 1 \wedge \gamma^\forall_t = 0) \vee (\gamma^\exists_t = 0 \wedge \gamma^\forall_t = 1).$$

- ($Transition$): A formula $\varphi_\vdash$ asserting that Each configuration γ_i has a transition via the transition relation of M to γ_{i+1}, for $0 \leq i < N$: This is the most crucial of the formulae and is the conjunction of the following two formulae. The first asserts that if a cell is not being scanned, it remains unchanged, the second that a transition of M changes the configuration as specified by the transition relation δ_M.

$$\bigwedge_{0\leq t<N} (\gamma^S_{t,i,a} = 1 \wedge \gamma^H_{t,i} = 0) \rightarrow \gamma^S_{t+1,i,a} = 1.$$

$$\bigwedge_{0\leq t<N,i\in[N]} (\gamma^S_{t,i,a} = 1 \wedge \gamma^H_{t,i} = 1 \wedge \gamma^Q_{t,q} = 1 \rightarrow$$

$$\bigvee_{(q,a,q',a',d)\in\delta_M} \gamma^S_{t+1,i,a'} = 1 \wedge \gamma^H_{t+1,i+d} = 1 \wedge \gamma^Q_{t+1,q'} = 1 \wedge \gamma^{g(q')}_{t+1} = 1)).$$

One can argue as in the proof of Cook's Theorem that the formula $TRANSITION$ has the properties stated.

We now write a sequence of formulae $ACCEPT_k(\alpha)$ for $k \geq 0$, such that

$$(2, \mathbf{a}) \models ACCEPT_k(\alpha),$$

iff the assignment of the values $\mathbf{a}$ to the variables α describes a configuration which is the root of an accepting computation sub–tree of M of alternation depth at most k. For $k = 0$, there are no alternations and so $ACCEPT_0(\alpha)$ is the formula:

$$\begin{array}{c}(\alpha^{\exists} = 1 \rightarrow \exists\beta \quad TRANSITION(\alpha, \beta) \wedge FINAL(\beta)) \vee \\ (\alpha^{\forall} = 1 \rightarrow \forall\beta \quad TRANSITION(\alpha, \beta) \wedge FINAL(\beta))\end{array}$$

Here, $FINAL(\beta)$ is the formula asserting that β is the unique accepting configuration:

$$\bigwedge_{i \in [N]} \beta^S_{i,0} = 1 \wedge \beta^H_1 = 1 \wedge \beta^Q_{q_f} = 1 \wedge \beta^{\exists} = 1.$$

Inductively, assuming that we have constructed $ACCEPT_k$ for some $k \geq 0$, we form $ACCEPT_{k+1}(\alpha)$:

$$\begin{array}{c}(\alpha^{\exists} = 1 \rightarrow \exists\beta \quad TRANSITION(\alpha, \beta) \wedge \beta^{\forall} = 1 \wedge ACCEPT_k(\beta)) \vee \\ (\alpha^{\forall} = 1 \rightarrow \forall\beta \quad TRANSITION(\alpha, \beta) \wedge \beta^{\exists} = 1 \wedge ACCEPT_k(\beta))\end{array}$$

Finally, given the input x, we can , in a strightforward fashion, construct a formula $START_x(\alpha)$ such that

$$(2, \mathbf{a}) \models START_x(\alpha),$$

iff the assignment of the values $\mathbf{a}$ to the variables α describes an assignment that is the start configuration of M on input x. Our final formula then, is

$$\psi_{M,x} := \exists\alpha(START_x(\alpha) \wedge ACCEPT_{|x|^k}(\alpha)).$$

It is easy to check that these formulas have the properties asserted. Our machine M which accepts an input x in time $|x|^k$ will make at most $|x|^k$ alternations. Hence,

$$2 \models \psi_{M,x} \quad \text{iff } M \text{ accepts } x.$$

The size of the formula $ACCEPT_k$ is, roughly, k times the size of the formula $TRANSITION$. The formula $TRANSITION$ has size bounded by a polynomial in $|x|$, as in Cook's Theorem. Hence, the formula $ACCEPT_{|x|^k}$ will also have size a bounded by a polynomial in $|x|$. Moreover, it can also be constructed in polynomial time. ∎

A finer result can be proved along the same lines, namely the one in Chapter 2, Exercise 16, characterising the Polynomial Time Hierarchy.

Proposition 55 *If $\mathcal{C}$ is a finite class of finite Boolean algebras, then* $\mathrm{Th}(\mathcal{C})$ *is $\leq_{\mathrm{poly}}$–complete for $PSPACE$.*

Proof. From Proposition 53, it only remains to show hardness. We establish this by a reduction from Th(2). This reduction is rather simple. For a formula $\varphi := Q_1x_1 \ldots Q_kx_k\phi(x_1, \ldots, x_k)$ with ϕ quantifier–free, involving variables $x_1, \ldots, x_k$, let φ' be defined as

$$\varphi' := Q_1x_1 \ldots Q_kx_k\phi(x_1, \ldots, x_k) \wedge ((x_1 = 0 \vee x_1 = 1) \wedge \cdots \wedge (x_k = 0 \vee x_k = 1)).$$

It is easy to see that $\varphi \in \mathrm{Th}(2)$ iff $\varphi' \in \mathrm{Th}(\mathcal{C})$. ∎

5.4 Extensions

A comprehensive classification of various sub–classes of Boolean algebras according to the complexity of their associated theories is given by Kozen [15]. This entails a much finer study of the structure of Boolean algebras than we have developed here. For completeness, we state the main results from [15].

Theorem 56 (Kozen [15]) *The following theories of sub–classes of Boolean algebras are in the class* $\bigcup_{c>0} STA(*, 2^{cn}, n)$*:*

- Th(*all Boolean algebras*).
- Th(*complete Boolean algebras*).
- Th(*atomic Boolean algebras*).
- Th(*atomless Boolean algebras*).

For the lower bounds, Kozen [15] develops techniques to encode properties of alternating TM computations as formulas in the language of Boolean algebras. These enable him to show a matching lower bound:

Theorem 57 *If* $\mathcal{C}$ *is a class of Boolean algebras that is either infinite or contains an infinite algebra, then* $\mathrm{Th}(\mathcal{C})$ *is* $\leq_{\log}$*–hard for* $\bigcup_{c>0} STA(*, 2^{cn}, n)$*.*

A notable corollary, by combining Theorems 56 and 57, is that the theories of complete, atomic, atomless Boolean algebras, as well as *the full theory of Boolean algebras is of the same complexity as the theory of real addition*, namely complete for the alternating class $\bigcup_{c>0} STA(*, 2^{cn}, n)$!

5.5 A Research Problem

A *Heyting algebra* is a structure $\mathsf{H} := (H, +, \cdot, \rightarrow, 0, 1)$ such that $(H, +, \cdot, 0, 1)$ is a distributive lattice (as with Boolean algebras) and the operation $\rightarrow$ is defined by the condition

$$x \leq a \rightarrow b \quad \text{iff} \quad xa \leq b.$$

(For the *cognoscenti*, this condition is best regarded in categorical terms as stating that the functor $a \rightarrow (\cdot)$ is the *right adjoint* to the functor $a \cdot (\cdot)$ as automorphisms of the poset $(H, \leq)$ considered as a category.) Boolean algebras are a special subclass of Heyting algebras, where we can define $a \rightarrow b := \overline{a} + b$. Like the class of Boolean algebras, the class of Heyting algebras too can be defined *equationally*.

Heyting algebras are the fundamental algebraic objects associated with *constructive* or *intuitionistic* logic, as Boolean algebras are the fundamental algebraic objects associated with classical logic. See van Dalen[23] for a gentle introduction to intuitionism.

A special subclass of Heyting algebras is the class of *linear orders*. A $(L, \leq, 0, 1)$ with least element 0 and greatest element 1 can be made into a Heyting algebra by defining the operations via

$$a + b := \max(a, b) \qquad ab := \min(a, b),$$

and

$$a \to b := \begin{cases} 1, & \text{if } a \leq b; \\ b, & \text{otherwise}; \end{cases}.$$

We shall denote by $\mathsf{l}([n])$, the unique linear order on n elements regarded as a Heyting algebra as above. The class of all Heyting algebras generated by taking the closure of a class $\mathcal{V}$ of Heyting algebras under the operations of direct products, sub–algebras and homomorphic images is denoted by $HSP(V)$. If $\mathcal{L}$ is the clas of all linear algebras, we denote $\mathcal{LC} := HSP(\mathcal{L})$. This class can be alternatively defined by adding the extra axiom

$$(x \to y) + (y \to x) = 1,$$

to the axioms of a Heyting algebra. For each $n \geq 1$, the subclass of $\mathcal{LC}$ defined by $\mathcal{LC}_n := HSP(\mathsf{l}([n]))$ can also be defined equationally. These sub–classes are obviously related in the following way:

$$\mathcal{LC}_1 \subseteq \mathcal{LC}_2 \subseteq \cdots \subseteq \mathcal{LC}.$$

Note that $\mathcal{LC}_2$ is the same as the class of all Boolean algebras.

The importance of the class $\mathcal{LC}$ from a complexity theorist's viewpoint is the following result of K. Idziak and P. Idziak [14]:

Theorem 58 (K. Idziak and P. Idziak) *For a class of algebras $\mathcal{C}$, denote by $\mathcal{C}^f$, the subclass of finite algebras in $\mathcal{C}$. Then*

1. *The only non–trivial sub–class of Heyting algebras with a decidable theory is the class of Boolean algebras.*

2. *The largest class $\mathcal{C}$ of Heyting algebras such that $\mathrm{Th}(\mathcal{C}^f)$ is decidable is $\mathcal{LC}$.*

The first part asserts that to get decidable theories, we had better restrict ourselves to considering only classes of finite algebras (with the notable exception of Boolean algebras!), and the second part asserts that $\mathcal{LC}$ is the largest class with a decidable theory of its finite members.

Research Problem 59 *Determine the complexities of the following theories of Heyting algebras:* $\mathrm{Th}(\mathcal{LC}^f)$, *and* $\mathrm{Th}(\mathcal{LC}_n^f)$ *for* $n \geq 3$.

To pursue a strategy similar to the one we adopted for Boolean algebras, it would be useful to have a representation theory for Heyting algebras. There is indeed a Representation Theorem for Heyting algebras as algebras of certain subsets of a partially ordered set.

Let $(P, \leq)$ be a partially ordered set. An *ideal* is a subset $X \subseteq P$ such that $y \leq x \in X$ implies $y \in X$. The set of all ideals, $I(P)$ ordered by set inclusion forms a Heyting algebra $\mathsf{I}(P) := (I(P), \cup, \cap, \rightarrow, \emptyset, P)$ where $X \rightarrow Y$ is defined to be the largest ideal that does not contain $X \setminus Y$, that is, the largest ideal W, such that $W \cap X \subseteq Y$.

An element $m \neq 0$ in a Heyting algebra is said to be *join–irreducible* if $m = a + b$ implies $m = a$ or $m = b$. The set of all join–irreducible elements in a Heyting algebra H is denoted H^*.

Theorem 60 (Representation Theorem) *Every finite Heyting algebra* H *is isomorphic to* $\mathsf{I}(\mathsf{H}^*)$*.*

The Representation Theorem specialised to the class $\mathcal{LC}$ and its sub–classes says even more. A poset $(P, \leq)$ is said to be a *forest* if for every $a \in P$, the set $\{b \in P \mid b \leq a\}$ is linearly ordered i.e. is a *chain*. A finite forest is said to have *height* n if the largest chain in it has at most n elements.

Proposition 61 (Representation Theorem for $\mathcal{LC}$) *Let* H *be a Heyting algebra. Then* $\mathsf{H} \in \mathcal{LC}$ *iff* H^* *is a forest. Moreover, for* $n \geq 1$*,* $\mathsf{H} \in \mathcal{LC}_n$ *iff* H^* *is a forest of height at most* n*.*

5.6 Bibliographical Notes

Strongly recommended are the the highly readable lectures of Paul Halmos [12]; they are old, but of vintage quality! A modern introduction to ordered structures is the text of Davey and Priestley [6] which also contains some chapters on Boolean algebras. The complexity theoretic results which inspired this chapter are due to Dexter Kozen [15]. Background information on the model–theoretic aspects of Boolean algebras can be obtained from the classic text of Chang and Kiesler [4]. The material on Heyting algebras is based on [14]. Background material on intuitionism can be obtained from van Dalen [23] and Troelstra and van Dalen [22].

Bibliography

[1] D. Kozen A. Chandra and L. Stockmeyer. Alternation. *J. Ass. Comput. Machinery*, 28(1), 1981.

[2] L. Berman. The complexity of logical theories. *Theoretical Computer Science*, 11:71–77, 1980.

[3] J. Canny. Some algebraic and geometric computations in pspace. ACM STOC, 1988.

[4] C.C. Chang and J. Kiesler. *Model Theory.* North–Holland, Amsterdam, 1973.

[5] G.E. Collins. Quantifier elimination for real closed fields by cylindric algebraic decomposition. volume 33 of *Lecture Notes in COmputer Science*, pages 134–183, Springer–Verlag, Berlin.

[6] B.A. Davis and H.A. Priestley. *Introduction to Lattices and Order.* Cambridge mathematical Textbooks. Cambridge University Press, 1990.

[7] B. Mishra P. Pedersen F. Cucker, H. Lanneau and M.-F. Roy. Nc algorithms for real algebraic numbers. *Applicable Algebra in Engineering, Communication and Computing*, 3:79–98, 1992.

[8] J. Ferrante and C. Rackoff. A decision procedure for the first order theory of real addition with order. *SIAM J. Comput.*, 4(1):69–76, March 1975.

[9] J. Ferrante and C. Rackoff. *The Computational Complexity of Logical Theories*, volume 718 of *Lecture Notes in Mathematics*. Springer–Verlag, Berlin, Heidelberg, New York, 1979.

[10] D. Grigor'ev. Complexity of deciding tarski algebra. *J. Symb. Computation*, 5:65–108, 1988.

[11] J. Flum H.-D. Ebbinghaus and W. Thomas. *Mathematical Logic.* Undergraduate Texts in Mathematics. Springer–Verlag, New York, Berlin, Heidelberg, 1984.

[12] P. R. Halmos. *Lectures on Boolean Algebras.* van Nostrand Mathematical Studies. D. van Nostrand Company Inc., 1963.

[13] J.E. Hopcroft and J.D. Ullman. *Introducton to Automata Theory, Languages and Computation.* Addison–Wesley, Reading, Mass., 1979.

[14] K. Idziak and P. Idziak. Decidability problem for finite heyting algebras. *J. Symbolic Logic*, 53(3):729–735, 1988.

[15] D.C. Kozen. Complexity of boolean algebras. *Theoretical Computer Science*, 10:221–247, 1980.

[16] D.C. Kozen. *The Design and Analysis of Algorithms.* Springer–Verlag, 1992.

[17] D. Kozen M. Ben-Or and J. Reif. The complexity of elementary algebra and geometry. *J. Comp. Syst. Sciences*, 32:251–264, 1986.

[18] D.C. Oppen. A $2^{2^{2^{pn}}}$ upper bound on the complexity of presburger's arithmetic. *J. Comp. and Syst. Sciences*, 16:323–332, 1978.

[19] C.H. Papdimitriou. *Computational Complexity.* Addison–Wesley, 1994.

[20] J. Renegar. On the computational complexity and geometry of the first order theory of reals, parts i,ii and iii. *J. Symb. Comput.*, 13(3):255–352, 1992.

[21] A. Tarski. *A Decision Method for Elementary Algebra and Geometry.* University of California Press, Berkeley, 1948.

[22] A. Troelstra and D. van Dalen. *Constructivism in Mathematics: An Introduction*, volume 121 of *Studies in Logic and the Foundations of Mathematics.* North–Holland, 1988.

[23] D. van Dalen. *Logic and Structure.* Springer–Verlag, Berlin Heidelberg New York, 1980.

Recent Publications in the BRICS Lecture Series

LS-95-5 **Devdatt P. Dubhashi.** ***Complexity of Logical Theories*****. September 1995. x+46 pp.**

LS-95-4 **Dany Breslauer and Devdatt P. Dubhashi.** ***Combinatorics for Computer Scientists*****. August 1995. viii+184 pp.**

LS-95-3 **Michael I. Schwartzbach.** ***Polymorphic Type Inference*****. June 1995. viii+24 pp.**

LS-95-2 **Sven Skyum.** ***Introduction to Parallel Algorithms*****. June 1995. viii+16 pp.**

LS-95-1 **Jaap van Oosten.** ***Basic Category Theory*****. January 1995. vi+75 pp.**