Bounded Reverse Mathematics

by

Phuong Nguyen

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Computer Science
University of Toronto

Abstract

Bounded Reverse Mathematics

Phuong Nguyen

Doctor of Philosophy

Graduate Department of Computer Science

University of Toronto

2008

First we provide a unified framework for developing theories of Bounded Arithmetic that are associated with uniform classes inside polytime ($\mathbf{P}$) in the same way that Buss's theory $\mathbf{S}^1_2$ is associated with $\mathbf{P}$. We obtain finitely axiomatized theories many of which turn out to be equivalent to a number of existing systems. By formalizing the proof of Barrington's Theorem (that the functions computable by polynomial-size bounded-width branching programs are precisely functions computable in **ALogTime**, or equivalently $\mathbf{NC}^1$) we prove one such equivalence between the theories associated with **ALogTime**, solving a problem that remains open in [Ara00, Pit00]. Our theories demonstrate an advantage of the simplicity of Zambella's two-sorted setting for small theories of Bounded Arithmetic. Then we give the first definitions for the relativizations of small classes such as $\mathbf{NC}^1$, $\mathbf{L}$, $\mathbf{NL}$ that preserve their inclusion order. Separating these relativized classes is shown to be as hard as separating the corresponding non-relativized classes. Our framework also allows us to obtain relativized theories that characterize the newly defined relativized classes. Finally we formalize and prove a number of mathematical theorems in our theories. In particular, we prove the discrete versions of the Jordan Curve Theorem in the theories $\mathbf{V}^0$ and $\mathbf{V}^0(2)$, and establish some facts about the distribution of prime numbers in the theory $\mathbf{VTC}^0$. Our $\mathbf{V}^0$- and $\mathbf{V}^0(2)$-proofs improve a number of existing upper bounds for the propositional complexity of combinatorial principles related to grid graphs. Overall, this thesis is a contribution to Bounded Reverse Mathematics, a theme

whose purpose is to formalize and prove (discrete versions of) mathematical theorems in the weakest possible theories of bounded arithmetic.

Acknowledgements

My deepest thanks go to Steve Cook who has been very kindly and patiently guiding me throughout my graduate studies. Steve has always helped me to see a clearer and bigger picture; he has set an excellent example for doing research as well as teaching. Time often seems to fly fast and there are always many interesting problems for us.

I would like to thank my committee members: Alasdair Urquhart, Toni Pitassi, Charlie Rackoff, and my external examiner Sam Buss for their careful reading, comments and suggestions on the thesis.

A large part of this thesis has appeared in [NC05, NC07, Ngu07, ACN07]. I would like to thank my co-authors, Steve Cook and Klaus Aehlig, for their permission to include the results here. In addition, the comments from the referees of these papers are very helpful. For some results I also benefit from discussions with Ho Minh Toan and Steven Perron.

I have been fortunate to have many office mates, colleagues and friends who have been helpful in different ways. The list is long so I do not enumerate them here.

This thesis is dedicated to my family who have been anxiously waiting for its completion.

Contents

1 Introduction **1**

1.1 Theories for Small Complexity Classes 3

1.1.1 Our Theories 3

1.1.2 Equivalence to Existing Systems 5

1.1.3 Relativized Theories 7

1.2 Bounded Reverse Mathematics 9

1.2.1 Proving the Discrete Jordan Curve Theorem 10

1.2.2 Distribution of Prime Numbers 12

1.3 Organization 14

2 Preliminaries **15**

2.1 Two-Sorted First-Order Logic 15

2.2 Two-Sorted Complexity Classes 16

2.3 $\mathbf{V}^0$ 19

2.4 $\overline{\mathbf{V}}^0$: A Universal Conservative Extension of $\mathbf{V}^0$ 22

3 Theories for Small Classes **25**

3.1 $\mathbf{VTC}^0$ 25

3.2 Theories for other Subclasses of $\mathbf{P}$ 28

3.2.1 Obtaining Theories for the Classes in (1.1) 28

3.2.2 The Theory $\overline{\mathbf{VC}}$ 29

3.2.3 Aggregate Functions . . . 31
3.2.4 Proof of the Definability Theorem for $\mathbf{VTC}^0$ . . . 37
3.3 $\mathbf{V}^0(m)$ and **VACC** . . . 38
3.4 $\mathbf{VNC}^1$ . . . 39
3.4.1 $\mathbf{VTC}^0 \subseteq \mathbf{VNC}^1$ . . . 41
3.5 **VNL** . . . 48
3.6 **VL** . . . 49
3.7 **VP** . . . 53
3.7.1 $\mathbf{VP} = \mathbf{TV}^0$ . . . 54
3.8 $\mathbf{VAC}^k$ and $\mathbf{VNC}^k$ . . . 57

4 Some Function Algebras **60**
4.1 Bounded Number Recursion . . . 61
4.2 Number Recursion for Permutations . . . 63
4.3 The String Comprehension Operation . . . 67

5 $\mathbf{VNC}^1 \overset{\text{RSUV}}{\simeq} \mathbf{QALV}$ **71**
5.1 RSUV Isomorphism . . . 71
5.2 The Theory **VALV** . . . 72
5.2.1 **QALV** . . . 73
5.2.2 $\mathbf{QALV} \overset{\text{RSUV}}{\simeq} \mathbf{VALV}$ . . . 74
5.3 **VALV** is Equivalent to $\mathbf{VNC}^1$ . . . 76
5.3.1 The Reduction to the Word Problem for S_5 . . . 78
5.3.2 Nonsolvability of S_5 . . . 80
5.3.3 Formalizing the Proof of Barrington's Theorem . . . 81

6 Theories for Relativized Classes **85**
6.1 Relativizing Subclasses of **P** . . . 85
6.1.1 $\mathbf{L}(\alpha)$ Reducibility . . . 89

6.2 Relativizing the Theories 90

7 The Discrete Jordan Curve Theorem **94**

7.1 Input as a Set of Edges 94

7.1.1 The Proof of the Main Theorem for $\mathbf{V}^0(2)$ 96

7.2 Input as a Sequence of Edges 99

7.2.1 There are at Least Two Regions 99

7.2.2 There Are at Most Two Regions 109

7.3 Proving the st-Connectivity Principle 110

8 Distribution of Prime Numbers **112**

8.1 A Lower Bound Proof for $\pi(n)$ 113

8.2 Approximating $\ln(x)$ 114

8.3 A Lower Bound Proof of $\pi(x)$ in $\mathbf{VTC}^0$ 119

8.4 Outline of an Upper Bound Proof of $\pi(n)$ 122

8.5 An Upper Bound Proof of $\pi(x)$ in $\mathbf{VTC}^0$ 123

8.6 Bertrand's Postulate and a Lower Bound for $\pi(2n) - \pi(n)$ 125

8.6.1 Formalization in $\mathbf{VTC}^0$ 127

8.7 Comparison with Earlier Work 128

9 Conclusion **130**

Index **140**

CLASS	THEORY	UNDERLYING PRINCIPLE/REMARK	REFERENCE
P	**PV**	Cobham's characterization	[Coo75]
	$\mathbf{S}^1_2$	$\mathbf{\Sigma}^b_1$ length induction	[Bus86b]
	$\mathbf{V}^1$**-HORN**	Horn-SAT Problem	[CK03]
	$\mathbf{TV}^0$	$\mathbf{\Sigma}^B_0$ string induction (see Section 3.7)	[Coo05]
	VP	Circuit Value Problem	Section 3.7
NC	**BL**, $\mathbf{D}^1_2$	Divide-and-conquer	[All91]
	TNC	$\mathbf{\Sigma}^b_1$-$\mathbf{L}_2$**IND**, $\mathbf{\Pi}^b_1$-**SEP**	[CT92]
	TAC	$\mathbf{\Sigma}^b_1$-$\mathbf{L}_2$**IND**	[CT95]
	$\mathbf{R}^1_2$, $\mathbf{U}^1_1$(BD)	$\mathbf{\Sigma}^b_1$-$\mathbf{L}_2$**IND**	[Tak93]
	$\mathbf{U}^1$	$\mathbf{\Sigma}^B_1$ length induction	[Coo05]
	VNC	Circuit Value Problem	Section 3.8
$\mathbf{AC}^k$ $(k \geq 1)$	$\mathbf{TAC}^k$	restricted nesting depth $\mathbf{\Sigma}^b_1$-$\mathbf{L}_2$**IND** (not closed under logical consequence)	[CT95]
	$\mathbf{VAC}^k$	Circuit Value Problem	Section 3.8
$\mathbf{NC}^k$ $(k \geq 2)$	$\mathbf{TNC}^{k-1}$	restricted nesting depth $\mathbf{\Sigma}^b_1$-$\mathbf{L}_2$**IND** (not closed under logical consequence)	[CT95]
	$\mathbf{VNC}^k$	Circuit Value Problem	Section 3.8
NL	$\mathbf{S}^{NLog}$	Encoding **NL** machines	[CT92]
	$\mathbf{V}^1$**-KROM**	Krom-SAT problem	[CK04]
	VNL	Reachability Problem	Section 3.5
L	$\mathbf{S}^{Log}$	Encoding logspace TMs	[CT92]
	TLS	(Out-degree 1) Reachability Problem	[CT95]
	$\mathbf{\Sigma}^B_0$**-Rec**	(Out-degree 1) Reachability Problem (see Section 3.6)	[Zam97]
	VL	(Out-degree 1) Reachability Problem	Section 3.6

CLASS	THEORY	UNDERLYING PRINCIPLE/REMARK	REFERENCE
$\mathbf{NC}^1$	$\mathbf{ALV}$	Formula Value Problem	[Clo90]
	$\mathbf{ALV}'$	Bounded width branching program (see Section 5.2.1)	[Clo93]
	$\mathbf{T}^0\mathbf{NC}^0$	Formula Value Problem	[CT95]
	$\mathbf{AID}$	Formula Value Problem	[Ara00]
	$\mathbf{T}_1$	Formula Value Problem	[Pit00]
	$\mathbf{VNC}^1$	Formula Value Problem	[CM05] Section 3.4
	$\mathbf{VALV}$	Bounded width branching program	Section 5.2
$\mathbf{TC}^0$	$(I\Sigma_0^{1,b})^{count}$	Counting number of 1-bits	[Kra95b]
	$\mathbf{TTC}^0$	$x \cdot y$ and esb bit comprehension (esb: *essentially sharply bounded*)	[CT95]
	$\overline{\mathbf{R}}^0$	$x \cdot y$ and $\mathbf{\Sigma}_0^b$ replacement	[Joh96]
	TV	Counting number of 1-bits	[Joh98]
	$\mathbf{\Delta}_1^b$-$\mathbf{CR}$	$x \cdot y$ and $\mathbf{\Delta}_1^b$ bit comprehension rule	[JP00]
	$\mathbf{VTC}^0$	Counting number of 1-bits	[NC04] Section 3.1
$\mathbf{ACC}$	$\mathbf{VACC}$	Counting modulo m for $m \geq 2$	Section 3.3
$\mathbf{AC}^0(m)$	$\mathbf{V}^0(m)$	Counting modulo m	Section 3.3
$\mathbf{AC}^0(6)$	$\mathbf{TAC}^0(6)$	2-BRN (or 3-BRN) (see (5.5))	[CT95]
	$\mathbf{V}^0(6)$	Counting modulo 6	Section 3.3
$\mathbf{AC}^0(2)$	$\mathbf{TAC}^0(2)$	1-BRN (see (5.5))	[CT95]
	A2V	Parity	[Joh98]
	$\mathbf{V}^0(2)$	Parity	Section 3.3
$\mathbf{AC}^0$	$\mathbf{TAC}^0$	esb bit comprehension	[CT95]
	$\mathbf{V}^0$	$\mathbf{\Sigma}_0^B$ comprehension	[Zam96, Coo02]

Chapter 1

Introduction

Bounded Arithmetic is the meeting point of Computational Complexity Theory and classical first-order logic. Problems in Computational Complexity Theory can be investigated using the first-order theories in Bounded Arithmetic. For example, Buss's theories $\mathbf{S}_2^i$ are closely related to the polynomial time hierarchy $\mathbf{PH}$: the functions computable by polytime Turing machines with Σ_{i-1}^p oracles (where $i \geq 1$) are precisely functions definable in Buss's theory $\mathbf{S}_2^i$ using $\mathbf{\Sigma}_i^b$ formulas [Bus86b]. It has been shown [KPT91, Bus95, Zam96] that the polynomial time hierarchy $\mathbf{PH}$ provably collapses if and only if the hierarchy $\mathbf{S}_2 = \bigcup \mathbf{S}_2^i$ collapses (i.e., $\mathbf{S}_2$ is finitely axiomatizable).

While Buss's systems nicely characterize $\mathbf{PH}$, there had not been satisfactory systems for many complexity classes inside polytime ($\mathbf{P}$). These classes pose many fundamental questions in theoretical computer science. For instance, an easier question than $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ is $\mathbf{AC}^0(6) \stackrel{?}{=} \mathbf{NP}$, or even $\mathbf{AC}^0(6) \stackrel{?}{=} \mathbf{PH}$. Moreover, subclasses of $\mathbf{P}$ are closely related to propositional proof systems such as Frege systems or Gentzen's system $\mathbf{PK}$ for predicate logic that are described in standard logic textbooks. In many cases, reasoning in these propositional proof systems involves precisely concepts that belong to the corresponding classes. (The first-order logic theories that we discuss below provide more uniform reasoning than the proof systems in the sense that the proofs in the theories can be translated into proofs in the corresponding propositional systems. Proofs in first-order theories are also easier to describe and understand, because they use the familiar axioms such as induction or minimization.)

In this thesis we start by presenting theories whose provably total functions are precisely functions of the following classes:

$$\mathbf{AC}^0 \subseteq \mathbf{AC}^0(m) \subseteq \mathbf{TC}^0 \subseteq \mathbf{NC}^1 \subseteq \mathbf{L} \subseteq \mathbf{NL} \subseteq \mathbf{NC} \subseteq \mathbf{P} \tag{1.1}$$

Previous theories (for example, in [All91, CT92, CT95, Joh98, Ara00, JP00]) developed for many of these classes often do not have nice set of axioms. Most of these theories are single-sorted and contain the usual language of arithmetic. In particular, they predefine the multiplication function $x \times y$ which is computationally hard for classes such as $\mathbf{TC}^0$, $\mathbf{AC}^0(2)$, $\mathbf{AC}^0$. Therefore the theories associated with classes that do not (or are not known to) contain $x \times y$ must have complicated sets of axioms to make sure that $x \times y$ is not a total function.

Although the function $x \times y$ is useful for formalizing machine computations, we only need it for "small" values of x, y, for example when x, y are indices to the input string. A clean separation of "small" from "big" objects is provided by Zambella's two-sorted setting [Zam96]. In this setting there are two sorts of objects: sets which can be viewed as binary strings (for inputs, outputs or computations) and numbers which are used mainly for indexing the strings. The multiplication function in the vocabulary is now predefined only for the number sort (i.e., "small" objects). In fact, the only predefined function on strings is the length function $|X|$. (The addition function for strings, though not creating a problem as does the multiplication function, will not be predefined.) Having $|X|$ as the only one predefined function on strings ($|X|$ is necessary anyway because we need to know the length of the inputs) gives us the flexibility to choose appropriate axioms that characterize the computations in the classes of interest.

Thus our theories will have Zambella's two-sorted setting. The simplicity of this setting also makes the Paris–Wilkie translation of proofs in the theories into propositional proofs mentioned above easier to describe, but we will not discuss this issue here.

1.1 Theories for Small Complexity Classes

A number of logical theories have been developed to characterize the complexity classes in (1.1). Many are developed in [CT95]; some others are listed below (see also the table on pages viii–ix):

- For **P**: **PV** [Coo75], $\mathbf{S}^1_2$ [Bus86b], $\mathbf{V}^1$-**HORN** [CK03], $\mathbf{TV}^0$ [Coo05].
- For **NC**: **BL**, $\mathbf{D}^1_2$ [All91], **TNC** [CT92], $\mathbf{R}^1_2$ [Tak93].
- For **NL**: $\mathbf{S}^{NLog}$ [CT92], $\mathbf{V}^1$-**KROM** [CK04].
- For **L**: $\mathbf{S}^{Log}$ [CT92], $\mathbf{\Sigma}^B_0$-**Rec** [Zam97].
- For $\mathbf{NC}^1$: **ALV** [Clo90], **ALV**$'$ [Clo93], **AID** [Ara00], $\mathbf{T}_1$ [Pit00], $\mathbf{VNC}^1$ [CM05].
- For $\mathbf{TC}^0$: $(I\Sigma^{1,b}_0)^{count}$ [Kra95b], $\overline{\mathbf{R}}^0$, TV [Joh96, Joh98], $\mathbf{\Delta}^b_1$-**CR** [JP00], $\mathbf{VTC}^0$ [Ngu04, NC04].
- For $\mathbf{AC}^0(2)$: **A2V** [Joh98].

Each theory mentioned above is developed in a unique way, and their associations with the corresponding classes are shown using various characterizations of the latter. This thesis provides a unified framework for developing theories for the classes in (1.1). In general, we show how to obtain a theory whose provably total functions are precisely the functions in a uniform subclass of **P** (more precisely, the $\mathbf{AC}^0$-closure of some polytime functions).

1.1.1 Our Theories

Showing that the provably total functions of a theory $\mathcal{T}$ are precisely the functions in a class **C** consists of two tasks: (i) showing that $\mathcal{T}$ can define every function $F(X)$ in **C**, and (ii) showing that all functions definable in $\mathcal{T}$ belong to **C**. Part (i) has often been done directly by using the definition of $\mathbf{C}$ based on some computing model. For (ii) essentially we need to show that for each theorem of $\mathcal{T}$ of the form

$$\forall X \exists Y \varphi(X, Y) \tag{1.2}$$

(where φ is a $\mathbf{\Sigma}_0^B$ formula, see Chapter 2) there is a function $F(X)$ in $\mathbf{C}$ that "witnesses" the existence of Y in φ, i.e.,

$$\forall X \varphi(X, F(X)).$$

This can be done by examining a (free-cut free) $\mathcal{T}$-proofs of (1.2).

In this thesis we follow the approach used in [Coo05], which goes back to earlier work, e.g., [Par71]. In this approach, both (i) and (ii) are reduced to the task of developing a universal conservative extension $\overline{\mathcal{T}}$ of $\mathcal{T}$, where $\overline{\mathcal{T}}$ contains symbols for all functions in $\mathbf{C}$. (Intuitively, (i) follows from the fact that $\overline{\mathcal{T}}$ is conservative over $\mathcal{T}$, and (ii) follows from the fact that $\mathcal{T}$ extends $\mathcal{T}$.) Here the language $\mathcal{L}_{\mathbf{FC}}$ of $\overline{\mathcal{T}}$ is obtained systematically using the notion of $\mathbf{AC}^0$-reduction [BIS90].

This approach is used in [Coo05] where Cook introduced the universal conservative extension $\overline{\mathbf{V}}^0$ of the theory $\mathbf{V}^0$ [Zam96, Coo02] which is associated with $\mathbf{AC}^0$. Showing that $\overline{\mathbf{V}}^0$ is a conservative extension of $\mathbf{V}^0$ is relatively straightforward; for example, the fact that $\overline{\mathbf{V}}^0$ is conservative over $\mathbf{V}^0$ follows from the fact that $\mathbf{\Sigma}_0^B(\mathcal{L}_{\mathbf{FAC}^0})$-formulas can be translated into equivalent $\mathbf{\Sigma}_0^B$-formulas in the language of $\mathbf{V}^0$. Our proof of conservativity in the general framework will be more complicated, because such translation might not be possible for other languages $\mathcal{L}_{\mathbf{FC}}$ (e.g., $\mathcal{L}_{\mathbf{FTC}^0}$).

We will prove generally that a theory $\mathbf{VC}$ that is axiomatized by $\mathbf{V}^0$ and an appropriate defining axiom for a polytime function F characterizes the $\mathbf{AC}^0$-closure of F in the same way that $\mathbf{V}^0$ characterizes $\mathbf{AC}^0$. (Our proof also applies to a collection of functions $\{F_i\}$.) Thus, by taking appropriate function F and its defining axiom, we obtain for each class $\mathbf{C}$ in (1.1) a theory $\mathbf{VC}$ whose vocabulary is the "base" vocabulary $\mathcal{L}_A^2$ of $\mathbf{V}^0$. The universal conservative extension $\overline{\mathbf{VC}}$ of $\mathbf{VC}$ extends $\overline{\mathbf{V}}^0$ and has symbols for all functions which are $\mathbf{AC}^0$-reducible to F, i.e., all functions in $\mathbf{C}$. The defining axioms for these functions are obtained simply by looking at their $\mathbf{AC}^0$ reduction to F. (The theories for $\mathbf{AC}^0(m)$ are called $\mathbf{V}^0(m)$ and $\overline{\mathbf{V}}^0(m)$).

Our first example is the pair of theories $\mathbf{VTC}^0$ and $\overline{\mathbf{VTC}}^0$. (The theory $\mathbf{VTC}^0$ is first developed in [Ngu04, NC04], but our proof given here is different from [Ngu04, NC04].) $\mathbf{VTC}^0$ is axiomatized by $\mathbf{V}^0$ together with the axiom *NUMONES* which can be seen as a defining axiom for *numones*, the function that counts the number of 1-bits in a binary string and that is $\mathbf{AC}^0$-complete for $\mathbf{TC}^0$. The language $\mathcal{L}_{\mathbf{FTC}^0}$ of $\overline{\mathbf{VTC}}^0$ has function symbols for the $\mathbf{AC}^0$-closure of *numones*.

Our choice of F for other classes such as $\mathbf{NC}^1$, $\mathbf{NL}$ is simply a polytime computation that solves a complete problem for the corresponding class. For example, the complete problem for $\mathbf{NC}^1$ is the Balanced Boolean Sentence Value problem, and the complete problem for $\mathbf{NL}$ is the Reachability (or Connectivity) problem in directed graphs. As in the case of *NUMONES*, the defining axiom for F is often easy to describe.

Since $\mathbf{V}^0$ is finitely axiomatizable [CK03], so are our theories (except for $\mathbf{VACC} = \bigcup_{m\geq 2}\mathbf{V}^0(m)$ and $\mathbf{VNC} = \bigcup_{k\geq 1}\mathbf{VNC}^k$). Also, $\mathbf{VC}$ and $\overline{\mathbf{VC}}$ are "minimal" theories that characterize $\mathbf{C}$, in the sense that $\overline{\mathbf{VC}}$ is axiomatized by "straightforward" defining axioms for functions in $\mathbf{C}$, i.e., the axioms describing the $\mathbf{AC}^0$ reductions of the functions to the chosen complete problem of $\mathbf{C}$. Furthermore, most of our theories are shown to be equivalent to a number of existing systems, demonstrating the robustness of our general framework.

1.1.2 Equivalence to Existing Systems

It is shown in [Ngu04, NC04] that $\mathbf{VTC}^0$ is equivalent to Johannsen–Pollett's theory $\mathbf{\Delta}_1^b$-$\mathbf{CR}$, a single-sorted theory defined in [JP00] using the *Comprehension Rule* for $\mathbf{\Delta}_1^b$ formula. It follows that $\mathbf{\Delta}_1^b$-$\mathbf{CR}$ is finitely axiomatizable and hence collapses to some segment $\mathbf{\Delta}_1^b$-$\mathbf{CR}_i$ where applications of the Comprehension Rule have nesting depth at most i, for some constant $i \in \mathbb{N}$. (In fact, it can be shown that $\mathbf{\Delta}_1^b$-**CR** collapses to $\mathbf{\Delta}_1^b$-$\mathbf{CR}_0$, i.e., no nesting application of the Comprehension Rule is needed.) This answers an open question from [JP00].

The two-sorted theory $\mathbf{V}^1$-**KROM** [Kol04, CK04] is defined using the fact from Finite Model Theory that Krom formulas express precisely **NL** relations [Grä92]. It has been shown [Kol04] that **VNL** is equivalent to $\mathbf{V}^1$-**KROM**.

Another two-sorted theory that is inspired by results from Finite Model Theory is $\mathbf{V}^1$-**HORN** [CK03] which is developed based on the fact that Horn formulas express precisely polytime relations. It has been shown that $\mathbf{V}^1$-**HORN** is equivalent to **PV** [Coo75] and also $\mathbf{V}^1$-**HORN** = $\mathbf{TV}^0$ [Coo05]. ($\mathbf{TV}^0$ is the two-sorted theory corresponding to the missing 0-th level of Buss's hierarchy $\mathbf{T}_2^i$.) In Section 3.7 we will show that our theory **VP** is equivalent to $\mathbf{TV}^0$ [Coo05]. It will follows that **VP** is equivalent to the existing theories $\mathbf{TV}^0$, $\mathbf{V}^1$-**HORN**, and **PV**.

$\mathbf{VNC}^1$ [CM05] is the two-sorted version of the single-sorted theory **AID** [Ara00] which in turn is defined using the fact that the Balanced Boolean Sentence problem is complete for $\mathbf{NC}^1$ [Bus87b]. Here we obtain an alternative formulation for $\mathbf{VNC}^1$. In addition, in Chapter 5 we will show that $\mathbf{VNC}^1$ is equivalent to **QALV** [Coo98], the quantified version of Clote's equational theory $\mathbf{ALV}'$ [Clo93]. This implies that $\mathbf{ALV}'$ is equivalent to **ALV** (another theory of Clote [Clo90]), and **QALV** is equivalent to **AID**, answering an open question from [Ara00, Pit00].

The theory $\mathbf{ALV}'$ [Clo93] is defined based on Barrington's Theorem [Bar89] that the bounded width branching programs compute exactly $\mathbf{NC}^1$ functions. We introduce a universal theory **VALV** whose vocabulary consists of all functions computable by width 5 branching programs. It is straightforward to show that **VALV** and **QALV** are equivalent (i.e., RSUV isomorphic), so the main task is to show that **VALV** is a conservative extension of $\mathbf{VNC}^1$. Essentially, we need to formalize Barrington's reduction and prove its correctness in **VALV**.

The defining axioms for the functions of **VALV** come from the so-called *function algebra* for $\mathbf{NC}^1$ functions based on Barrington's Theorem (see [CK02]). In Chapter 4 we prove a number of other function algebras characterizing several subclasses of **L**. For $\mathbf{AC}^0(2)$ and $\mathbf{AC}^0(6)$ these can be regarded as the two-sorted version of the function algebras discussed in [CT95] that go back to [PW85]. The function algebra for **L** can be viewed as the two-sorted version of Lind's characterization of **L** (or Clote's operation $\mathrm{B}_2\mathrm{PR}$ in [CT92]) and has been discussed in [Per05]. These function algebras can be used to develop universal theories that are equivalent to $\overline{\mathbf{VC}}$ (e.g., **VALV** is equivalent to $\overline{\mathbf{VNC}}^1$, see Chapter 5), but we will not go into further detail here.

It might be possible to show that our theories $\mathbf{VNC}^k$ and $\mathbf{VAC}^k$ are equivalent (for certain classes of formulas) respectively to the systems $\mathbf{TNC}^k$ and $\mathbf{TAC}^k$ defined in [CT95]. However we do not attempt to prove such equivalences here. $\mathbf{TNC}^k$ and $\mathbf{TAC}^k$ are defined using a complicated syntactic notion called *essentially sharply bounded* (esb) formulas, and proofs in $\mathbf{TNC}^k$ or $\mathbf{TAC}^k$ are restricted to having some constant upper bound on the nesting depth of the rules such as esb-LIND. Because of this restriction, it has been noticed that $\mathbf{TNC}^k$ and $\mathbf{TAC}^k$ are not really "theories" in the sense that they are not closed under logical consequence. Of course one may consider the theories that are axiomatized by their $\mathbf{\Sigma}_1^b$ consequences, but we will not go into further detail here.

1.1.3 Relativized Theories

Existing definitions of the relativizations of some important subclasses of **P** are not satisfactory in the sense that they do not preserve the following nonrelativized inclusions simultaneously:

$$\mathbf{NC}^1 \subseteq \mathbf{L} \subseteq \mathbf{NL} \subseteq \mathbf{AC}^1 \tag{1.3}$$

For example [LL76], if the Turing machines are allowed to be nondeterministic when writing oracle queries, then there is an oracle α so that $\mathbf{NL}(\alpha) \not\subseteq \mathbf{P}(\alpha)$. Later definitions of $\mathbf{NL}(\alpha)$ adopt the requirement specified in [RST84] that the nondeterministic oracle machines be deterministic whenever the oracle tape (or oracle stack) is nonempty. Then the inclusion $\mathbf{NL}(\alpha) \subseteq \mathbf{P}(\alpha)$ relativizes, but not all inclusions in (1.3).

Because the nesting depth of oracle gates in an oracle $\mathbf{NC}^1$ circuit can be bigger than one, the model of relativization that preserves the inclusion $\mathbf{NC}^1 \subseteq \mathbf{L}$ must allow an oracle logspace Turing machine to have access to more than one oracle query tape [Orp84, Bus86a, Wil88]. For the model defined by Wilson [Wil88], the partially constructed oracle queries are stored in a stack. The machine can write queries only on the oracle tape at the top of the stack. It can start a new query on an empty oracle tape (thus *pushing* down the current oracle tape, if there is any), or query the content of the top tape which then becomes empty and the stack is *popped*.

Following Cook [Coo85], the circuits accepting languages in relativized $\mathbf{NC}^1$ are those with logarithmic depth where the Boolean gates have bounded fanin and an oracle gate of m inputs contributes $\log(m)$ to the depths of its parents. Then in order to relativize the inclusion $\mathbf{NC}^1 \subseteq \mathbf{L}$, the oracle logspace machines defined by Wilson [Wil88] are required to satisfy the condition that at any time,

$$\sum_{i=1}^{k} max\{\log(|q_i|), 1\} = \mathcal{O}(\log(n))$$

where $q_1, q_2, \ldots, q_k$ are the contents of the stack and $|q_i|$ are their lengths. For the simulation of an oracle $\mathbf{NC}^1$ circuit by such an oracle logspace machine the upper bound $\mathcal{O}(\log(n))$ cannot be improved.

Although the above definition of $\mathbf{L}(\alpha)$ (and $\mathbf{NL}(\alpha)$) ensures that $\mathbf{NC}^1(\alpha) \subseteq \mathbf{L}(\alpha)$, unfortunately we know only that $\mathbf{NL}(\alpha) \subseteq \mathbf{AC}^2(\alpha)$ [Wil88]; the inclusion $\mathbf{NL}(\alpha) \subseteq \mathbf{AC}^1(\alpha)$ is left open.

We observe that if the height of the oracle stack is bounded by a constant (while the lengths of the queries are still bounded by a polynomial in the length of the inputs), then an oracle $\mathbf{NL}$ machine can be simulated by an oracle $\mathbf{AC}^1$ circuit, i.e., $\mathbf{NL}(\alpha) \subseteq \mathbf{AC}^1(\alpha)$. In fact, $\mathbf{NL}(\alpha)$ can then be shown to be the $\mathbf{AC}^0(\alpha)$ closure of the Reachability problem for directed graphs. Similarly, $\mathbf{L}(\alpha)$ is the $\mathbf{AC}^0(\alpha)$ closure of the Reachability problem for directed graphs whose out-degree is at most one.

The $\mathbf{AC}^0(\alpha)$ closure of the Boolean Sentence Value problem (which is $\mathbf{AC}^0$ complete for $\mathbf{NC}^1$) turns out to be the languages computable by uniform oracle $\mathbf{NC}^1$ circuits (defined as before) where the nesting depth of oracle gates is now bounded by a constant. We redefine $\mathbf{NC}^1(\alpha)$ using this new restriction on the oracle gates; the new definition is more suitable in the context of $\mathbf{AC}^0(\alpha)$ reducibility (the previous definition of $\mathbf{NC}^1(\alpha)$ seems suitable when one considers $\mathbf{NC}^1(\alpha)$ reducibility). Consequently, we obtain the first definition of $\mathbf{NC}^1(\alpha)$, $\mathbf{L}(\alpha)$ and $\mathbf{NL}(\alpha)$ that preserves the inclusions in (1.3).

The $\mathbf{AC}^0$-complete problems for $\mathbf{AC}^0(m)$ and $\mathbf{TC}^0$ remain complete for the relativized classes under $\mathbf{AC}^0(\alpha)$-reduction, and the same is true for $\mathbf{L}$ and $\mathbf{NL}$ under the new definitions of their relativizations. Therefore the existence of any oracle that separates two classes in the list $\mathbf{AC}^0(m), \mathbf{TC}^0, \mathbf{NC}^1, \mathbf{L}, \mathbf{NL}, \mathbf{AC}^1$ implies their nonrelativized separation: If the nonrelativized classes are equal, their complete problems would be equivalent under $\mathbf{AC}^0$-reductions, hence under $\mathbf{AC}^0(\alpha)$-reductions, and therefore the relativized classes would coincide. So separating the relativized classes is as hard as separating their nonrelativized counterparts. This nicely generalizes known results [Wil88, Sim77, Wil89].

Having defined the relativizations of classes in (1.3), our general framework discussed in previous section (and in detail in Chapter 3) is ready to produce their associated theories. Here we use $\mathbf{AC}^0(\alpha)$-reduction instead of $\mathbf{AC}^0$-reduction.

1.2 Bounded Reverse Mathematics

An application of the theories of Bounded Arithmetic is in formalizing arguments that require concepts of certain complexity. An example is Razborov's $\mathbf{S}^1_2$–proof of Håstad's Switching Lemma [Raz95]. The quest for a (dis)proof of the unboundedness of the prime numbers in $\mathbf{I\Delta}_0$ can also be listed here. This recently shaped research direction [Coo07] is called Bounded Reverse Mathematics because of its similarity to the Reverse Mathematics program initiated by Friedman and Simpson (see [Sim99]). In Reverse Mathematics the theories can define all primitive recursive functions, while here we are concerned with much lower complexity.

In fact, a large part of this thesis that we have discussed so far can be seen as devoted to Bounded Reverse Mathematics. For example, to show that $\mathbf{VTC}^0 \subseteq \mathbf{VNC}^1$ (Section 3.4.1) we need to formalize in $\mathbf{VNC}^1$ the construction of $\mathbf{NC}^1$ circuits that compute *numones*. Here we follow [Bus87a].

As another example, proving that a theory $\mathbf{VC}$ is equivalent to some existing theory $\mathbf{T}$ requires showing (i) that the finite set of axioms of $\mathbf{VC}$ (namely the axioms of $\mathbf{V}^0$, and most importantly the defining axiom for the corresponding $\mathbf{AC}^0$-complete function of the associated class $\mathbf{C}$) are provable (or interpretable) in $\mathbf{T}$, and (ii) that the axioms of $\mathbf{T}$ are provable (or interpretable) in $\mathbf{VC}$. For instance, as we discussed before, one direction in the proof of the equivalence between $\mathbf{VNC}^1$ and $\mathbf{QALV}$ requires essentially a formalization and proof of correctness for Barrington's reduction [Bar89] in $\mathbf{QALV}$ (or equivalently $\mathbf{VALV}$).

In general we are interested in proving (the discrete versions of) mathematical theorems in (the weakest possible) theories of Bounded Arithmetic. In the last part of this thesis we will consider the Jordan Curve Theorem (that a simple closed curve divides the two dimensional plane into exactly two connected components) and some facts about the distribution of prime numbers (i.e., Chebyshev's Theorem which states that the number of primes less than n is $\Theta(n/\ln(n))$, and the fact that there are $\Theta(n/\ln(n))$ prime numbers between n and $2n$).

1.2.1 Proving the Discrete Jordan Curve Theorem

The Jordan Curve Theorem (JCT) states that a simple closed curve divides the two dimensional plane into exactly two connected components. We prove this theorem when the curve lies on an $n \times n$ grid. The results in this chapter are inspired by Thomas Hales' talk on his computer–verified proof of the original theorem [Hal05]. Hales' proof starts with the above discrete version of the theorem, and is based on Thomassen's proof [Tho92] which derives the JCT from the non-planarity of $K_{3,3}$.

In [Bus06] Buss considered the st-connectivity for grid graphs which states that it is not possible to have a red path and a blue path that connect opposite corners of the grid unless the paths intersect. This principle can be expressed as tautologies in two ways depending on how the paths are presented: the harder tautologies $STCONN(n)$ [Bus06] express the red and blue edges as two sets, with the condition that every node except the corners has degree 0 or 2 (thus allowing disjoint cycles as well as paths). The easier tautologies $STSEQ(n)$ express the paths as sequences of edges.

In 1997 Cook and Rackoff [CR97] showed, using the idea of winding numbers, that the easier tautologies $STSEQ(n)$ have polynomial size $\mathbf{TC}^0$-**Frege**-proofs. Buss [Bus06] showed that the harder tautologies $STCONN(n)$ also have polynomial size $\mathbf{TC}^0$-**Frege**-proofs, improving the earlier result. His proof shows how the red and blue edges in each column of the grid graph determine an element of a certain finitely-generated group. The first and last columns determine different elements, but assuming the red and blue paths do not cross, adjacent columns must determine the same element. This leads to a contradiction.

We give proofs of the principles in the theories $\mathbf{V}^0$ and $\mathbf{V}^0(2)$, which imply upper bounds on the propositional proof complexity of the principles. In Section 7.1 we show that $\mathbf{V}^0(2)$ proves the part of the discrete JCT asserting a closed curve divides the plane into at least two connected components, for the (harder) case in which the curve and paths are given as sets of edges. The proof is based on the idea that a vertical line passing through a grid curve can detect which regions are inside and outside the curve by the parity of the number of horizontal edges it intersects. It follows that $\mathbf{V}^0(2)$ proves the st-connectivity principle for edge sets.

As a corollary we conclude that the $STCONN(n)$ tautologies (as well as Urquhart's Hex tautologies, see [Bus06]) have polynomial size $\mathbf{AC}^0(2)$-**Frege**-proofs, thus strengthening Buss's [Bus06] result that is stated for the stronger $\mathbf{TC}^0$-**Frege** system. Our result is stronger in two senses: the proof system is weaker, and we show the existence of uniform proofs by showing the st-connectivity principle is provable in $\mathbf{V}^0(2)$. In fact, showing provability in a theory such as $\mathbf{V}^0(2)$ is often easier than directly showing its corollary that the corresponding tautologies have polynomial size proofs. This is because we can use the fact that the theory proves the induction scheme and the minimization scheme for formulas expressing concepts in the corresponding complexity class.

In Section 7.2 we prove the surprising result that when the input curve and paths are presented as sequences of grid edges then even the very weak theory $\mathbf{V}^0$ proves the Jordan Curve Theorem. The proof is technically complicated because we can use only $\mathbf{AC}^0$ concepts. The key idea is to show that in every column of the grid, the horizontal edges of the curve alternate between pointing right and pointing left. It follows that $\mathbf{V}^0$ proves the st-connectivity principle for sequences of edges. As a corollary we conclude that the $STSEQ(n)$ tautologies have polynomial size $\mathbf{AC}^0$-**Frege**-proofs. This strengthens the early result [CR97] (based on winding numbers) that $STSEQ(n)$ have polynomial size $\mathbf{TC}^0$-**Frege**-proofs.

1.2.2 Distribution of Prime Numbers

It is shown in [HAB02] that there is a uniform $\mathbf{TC}^0$ algorithm for integer division, or in other words, there is an $\mathbf{FO}(M)$ formula (i.e., a first-order formula with the counting quantifier) that express the relation $Z = \lfloor X/Y \rfloor$. The results in this chapter come out of the effort (which has been so far unsuccessful) to formalize this algorithm in $\mathbf{VTC}^0$.

The $\mathbf{TC}^0$ algorithm given in [HAB02] uses the Chinese Remainder Theorem which requires a lower bound for the number of prime numbers of certain magnitude, e.g., there are $\Omega(n/\ln(n))$ prime numbers between n and $2n$. The lower bound is taken for granted when designing the $\mathbf{TC}^0$ circuit or defining the $\mathbf{FO}(M)$ formula because we know that it exists by the Prime Number Theorem. In formalizing the algorithm in $\mathbf{VTC}^0$, however, we first need to establish such a lower bound. We are able to prove the existence of a sufficient number of primes for the algorithm from [HAB02], and our results can be seen as the first step toward proving the correctness of the algorithm in $\mathbf{VTC}^0$.

Let $\pi(n)$ denote the number of primes that are $\leq n$. Chebyshev's Theorem states that $\pi(n) = \Theta(n/\ln(n))$. Indeed, with simple proofs it can be shown that for sufficiently large n,

$$\frac{\ln(2)}{2}\frac{n}{\ln(n)} \leq \pi(n) \leq 2\ln(2)\frac{n}{\ln(n)} \tag{1.4}$$

We will give a $\mathbf{VTC}^0$ proof of Chebyshev's Theorem, though with a bigger constant factor than $2\ln(2)$ for the upper bound. (This constant can be improved using the same method but at the cost of increasing the threshold for n to some unpleasantly high value.)

We will also give a $\mathbf{VTC}^0$ proof for the facts that

$$\pi(2n) - \pi(n) = \Omega(n/\ln(n)) \qquad \text{and} \qquad \pi(2n) - \pi(n) \geq 1 \quad (\text{for } n \geq 1)$$

Here we use the idea from [Mos49]. The proof from [Mos49], however, uses the upper bound for $\pi(n)$ shown in (1.4). As mentioned before, we do not have such tight upper bound for $\pi(n)$. So our $\mathbf{VTC}^0$ proof is derived from [Mos49] by a more careful case analysis.

The original proofs that we follow all use "big" objects such as $(2n)!/n!n!$, which is $\mathcal{O}(4^n)$. We avoid computing such big objects by computing their logarithms instead. Notice that the function $\log(x) = \lfloor \log_2(x) \rfloor$ is definable in $\mathbf{I\Delta}_0$ [Ben62, HP93, Bus98, CN06], however it provides a very crude approximation to $\log_2(x)$ and seems insufficient for our purpose. We are lead to define a finer approximation, and since

$$\ln(x) = \int_1^x \frac{1}{y} dy$$

a sufficient approximation to $\ln(n)$ can be calculated in $\mathbf{VTC}^0$ using the *numones* function.

Our formalizations here are similar to Woods' formalization of the proof of the unboundedness of prime numbers in the theory $\mathbf{I\Delta}_0 + \mathbf{PHP}(\mathbf{\Delta}_0)$ [Woo81]. For example, [Woo81] also defines an approximation to $\ln(x)$ (for $x \leq (\log(a))^c$ for some $c \in \mathbb{N}$ and for some a). Our approximation to $\ln(x)$ is more direct, and we prove in addition the lower bound for $\pi(2n) - \pi(n)$. Note that the method used in [Woo81] and in this thesis gives an $\mathbf{I\Delta}_0$-proof of Bertrand's Postulate for numbers n where $n \leq (\log(a))^c$, while the formalization in [D'A92] proves the postulate only for $n \leq \log(a)$. (The fact that we can approximate $\ln(x)$ for values of x larger than those in [Woo81] is because in $\mathbf{I\Delta}_0$ it is possible to "count" a set of cardinality up to only $(\log(a))^c$, while in $\mathbf{VTC}^0$ we can count up to a.)

We discovered some earlier results [CD94, Cor95] just as the thesis is to be submitted. They are discussed in Section 8.7.

1.3 Organization

In Chapter 2 we formally define the two-sorted setting and the theories $\mathbf{V}^0$, $\overline{\mathbf{V}}^0$. The materials in this chapter are from [Coo05] and [CN06, Chapter 5]. The theories $\mathbf{VC}$ are developed in Chapter 3. They have appeared in [NC05] and [CN06, Chapter 9]. The function algebras for a number of subclasses of $\mathbf{L}$ are discussed in Chapter 4. Chapter 5 proves the equivalence between $\mathbf{VNC}^1$ and $\mathbf{QALV}$. The results of this chapter will appear in [Ngu07]. The new definitions of the relativizations of classes in (1.3) and their theories are given in Chapter 6 and have been presented in [ACN07]. The proofs of the discrete Jordan Curve Theorem in $\mathbf{V}^0$ and $\mathbf{V}^0(2)$ are in Chapter 7; they have been presented in [NC07]. The formalizations in $\mathbf{VTC}^0$ of the facts about distribution of prime numbers are given in Chapter 8. Finally, Chapter 9 contains some concluding remarks.

Chapter 2

Preliminaries

We present a two-sorted setting for first-order theories and complexity classes. Then we define the base theory $\mathbf{V}^0$ and its universal conservative extension $\overline{\mathbf{V}}^0$. The materials of this chapter are from [Coo05, CN06].

2.1 Two-Sorted First-Order Logic

There are two kinds of variables: $x, y, z, \ldots$ (*number* variables) are intended to range over $\mathbb{N}$; and $X, Y, Z, \ldots$ (*set*, or *string* variables) are intended to range over finite subsets of $\mathbb{N}$ (which are represented as binary strings). The basic two-sorted vocabulary is

$$\mathcal{L}_A^2 = [0, 1, +, \cdot, |\ | \ ; \ =_1, \ =_2, \leq, \in]$$

where $0, 1, +, \cdot, =_1, \leq$ are for arithmetic over $\mathbb{N}$; $|X|$ is the length function (1 plus the largest element in X, or 0 if X is empty) which is roughly the length of the binary string representing X; $t \in X$ (or $X(t)$) is the membership relation; and $=_2$ is equality for strings. We often write $=$ for both $=_1$ and $=_2$, the exact meaning is clear from the context.

Number terms are built from 0, 1, $x, y, z, \ldots$ and the length term $|X|$ using $+, \cdot$. The only string terms are $X, Y, \ldots$. The atomic formulas are $s = t, s \leq t, X = Y, X(t)$ for number terms s, t and string variables X, Y. Formulas are built from atomic formulas using $\wedge, \vee, \neg$ and both number and string quantifiers $\exists x, \forall x, \exists X, \forall X$. Bounded quantifiers are: $\exists x \leq t\ \varphi$ stands for $\exists x(x \leq t \wedge \varphi)$, $\forall x \leq t\ \varphi$ stands for $\forall x(x \leq t \supset \varphi)$, $\exists X \leq t\ \varphi$ stands for $\exists X(|X| \leq t \wedge \varphi)$, and $\forall X \leq t\ \varphi$ stands for $\forall X(|X| \leq t \supset \varphi)$, where t is an $\mathcal{L}_A^2$ number term that does not contain x (or X).

$\mathbf{\Sigma}_0^B$ is the set of all $\mathcal{L}$-formulas where all number quantifiers are bounded and with no string quantifiers. $\mathbf{\Sigma}_1^B$ formulas begin with zero or more bounded existential string quantifiers, followed by a $\mathbf{\Sigma}_0^B$ formula. These classes are extended to $\mathbf{\Sigma}_i^B$, $i \geq 2$, (and $\mathbf{\Pi}_i^B$, $i \geq 0$) in the usual way. (Thus $\mathbf{\Sigma}_1^B$ corresponds to *strict* $\Sigma_1^{1,b}$ in [Kra95a]). We will consider vocabularies $\mathcal{L} \supseteq \mathcal{L}_A^2$. We will use s, t as metasymbols for number terms, S, T for string terms. Also, the sets $\mathbf{\Sigma}_i^B(\mathcal{L})$ and $\mathbf{\Pi}_i^B(\mathcal{L})$ are defined in the same way as $\mathbf{\Sigma}_i^B$ and $\mathbf{\Pi}_i^B$.

2.2 Two-Sorted Complexity Classes

To define circuit complexity classes we use **FO** (or equivalently DLOGTIME) uniformity (see [BIS90, Imm99]).

Definition 2.1. *For $k \geq 0$, $\mathbf{AC}^k$ (resp. $\mathbf{NC}^k$) is the class of languages accepted by uniform families of polynomial-size Boolean circuits that have depth $\mathcal{O}((\log n)^k)$ (where n is the number of inputs) whose gates have unbounded (resp. bounded) fan-in. $\mathbf{TC}^0$ (resp. $\mathbf{AC}^0(m)$) is the class of languages computable by uniform family of polynomial-size, constant-depth circuits with threshold gates (resp. modulo m gates).* **L** *(resp.* **NL***) denotes the class of languages computable by deterministic (resp. nondeterministic) logspace Turing machines, and* **P** *is the class of languages computable in polynomial time by deterministic Turing machines.*

In defining the complexity of a relation $R(\vec{x}, \vec{X})$ or function $f(\vec{x}, \vec{X})$ or $F(\vec{x}, \vec{X})$, the arguments x_i are represented in unary notation (a string of x_i ones), and X_j are represented as bit strings. We think of the number arguments as auxiliary inputs useful for indexing the bit strings. Here we are interested in functions that grow polynomially in length.

Definition 2.2. *A function $F(\vec{x}, \vec{X})$ (resp. $f(\vec{x}, \vec{X})$) is polynomially bounded (or p-bounded) if there is a polynomial $p(n)$ such that $|F(\vec{x}, \vec{X})| \leq p(max(\vec{x}, |\vec{X}|))$ (resp. $f(\vec{x}, \vec{X}) \leq p(max(\vec{x}, |\vec{X}|))$).*

The complexity of a string function is related to its *bit graph* (defined below) rather than its graph. For example, consider the factoring function

$$F(X) = \langle Y_1, m_1, Y_2, m_2, \ldots, Y_k, m_k \rangle$$

where Y_i are distinct prime factors of X and $\prod_{i=1}^{k} Y_i^{m_1} = X$, for $X \geq 2$. Then the graph of F is a polytime relation, while F is not known to be in $\mathbf{P}$.

Definition 2.3. *The* bit graph *of a string function F is $B_F(i, \vec{x}, \vec{X}) \equiv F(\vec{x}, \vec{X})(i)$.*

Definition 2.4 (Function Class)**.** *If $\mathbf{C}$ is a two-sorted complexity class of relations, then the corresponding function class* **FC** *consists of all p-bounded number functions whose graphs are in $\mathbf{C}$, together with all p-bounded string functions whose bit graphs are in $\mathbf{C}$.*

Uniform $\mathbf{AC}^0$ (or just $\mathbf{AC}^0$) has several equivalent definitions: **LTH** (the log time hierarchy on alternating Turing machines) and **FO** (describable by a first-order formula using $<$ and *Bit* predicates). Here we have [Zam96, Imm99, CN06]:

Theorem 2.5 ($\mathbf{\Sigma}_0^B$ Representation Theorem)**.** *A relation $R(\vec{x}, \vec{X})$ is in $\mathbf{AC}^0$ iff it is represented by some $\mathbf{\Sigma}_0^B$ formula $\varphi(\vec{x}, \vec{X})$.*

The following example is from [Ben62, HP93, Bus98, CN06]:

Example 2.6. *The relation (on numbers) $y = z^x$ is in $\mathbf{AC}^0$.*

Also, binary addition $F_+(X,Y) = X + Y$ is in $\mathbf{FAC}^0$, and binary multiplication $F_\times(X,Y) = X \cdot Y$ is in $\mathbf{FTC}^0$ but not in $\mathbf{FAC}^0$.

Theorem 2.5 motivates the following notion of $\mathbf{AC}^0$-reducibility [Coo05]. The idea is that a function F is $\mathbf{AC}^0$-reducible to a collection $\mathcal{L}$ of functions if F can be computed by a uniform polynomial-size constant-depth family of circuits which have unbounded fan-in gates computing functions from $\mathcal{L}$, in addition to Boolean gates. All classes that we consider are closed under $\mathbf{AC}^0$ reduction.

Definition 2.7. *A string function F (resp. number function f) is $\mathbf{\Sigma}_0^B$-definable from $\mathcal{L}$ if it is polynomially bounded, and its bit graph (resp. graph) is represented by a $\mathbf{\Sigma}_0^B(\mathcal{L})$ formula.*

Definition 2.8 ($\mathbf{AC}^0$ Reduction)**.** *A string function F (resp. number function f) is $\mathbf{AC}^0$-reducible to $\mathcal{L}$ if there is a sequence of string functions $F_1, \ldots, F_n$ ($n \geq 0$) such that*

$$F_i \textit{ is } \mathbf{\Sigma}_0^B\textit{-definable from } \mathcal{L} \cup \{F_1, \ldots, F_{i-1}\}, \textit{ for } i = 1, \ldots, n; \tag{2.1}$$

and that F (resp. f) is $\mathbf{\Sigma}_0^B$-definable from $\mathcal{L} \cup \{F_1, \ldots, F_n\}$. A relation R is $\mathbf{AC}^0$-reducible to $\mathcal{L}$ if there is a sequence $F_1, \ldots, F_n$ as above, and R is represented by a $\mathbf{\Sigma}_0^B(\mathcal{L} \cup \{F_1, \ldots, F_n\})$ formula.

If in the above definition $\mathcal{L}$ consists only of functions in $\mathbf{FAC}^0$, then a single iteration ($n = 1$) is enough to obtain any function in $\mathbf{FAC}^0$, and it can be shown that no more functions are obtained by further iterations. However, if we start with a function such as $numones(z, X)$ (the number of elements of X that are $< z$), then repeated iterations generate the complexity class $\mathbf{TC}^0$. As far as we know there is no bound on the number of iterations needed, because (as far as we know) there is no fixed d such that every member of $\mathbf{TC}^0$ can be defined by a polynomial-size family of circuits of depth d.

The next lemma is immediate from definition.

Lemma 2.9. *Let $\mathcal{L}$ be a set of functions, and $\mathbf{C}$ be the class of relations which are $\mathbf{AC}^0$-reducible to $\mathcal{L}$. Then $\mathbf{FC}$ is the class of functions which are $\mathbf{AC}^0$-reducible to $\mathcal{L}$.*

B1. $x+1 \neq 0$	**B7.** $(x \leq y \wedge y \leq x) \supset x = y$
B2. $x+1 = y+1 \supset x = y$	**B8.** $x \leq x + y$
B3. $x + 0 = x$	**B9.** $0 \leq x$
B4. $x + (y+1) = (x+y)+1$	**B10.** $x \leq y \vee y \leq x$
B5. $x \cdot 0 = 0$	**B11.** $x \leq y \leftrightarrow x < y + 1$
B6. $x \cdot (y+1) = (x \cdot y) + x$	**B12.** $x \neq 0 \supset \exists y \leq x(y+1 = x)$
L1. $X(y) \supset y < \|X\|$	**L2.** $y + 1 = \|X\| \supset X(y)$
SE. $[\|X\| = \|Y\| \wedge \forall i < \|X\|(X(i) \leftrightarrow Y(i))] \ \supset \ X = Y$	

Figure 2.1: 2-**BASIC**

2.3 $\mathbf{V}^0$

A theory $\mathcal{T}$ over $\mathcal{L}$ is polynomial-bounded if (i) it extends $\mathbf{V}^0$ (defined below), (ii) it can be axiomatized by a set of bounded formulas, and (iii) all functions in $\mathcal{L}$ are p-bounded. All theories considered in this thesis are polynomial-bounded. It follows from Parikh's Theorem that provably total functions of a polynomial-bounded theory are p-bounded.

Definition 2.10 (Comprehension Axiom)**.** *If Φ is a set of formulas, then the* compre-hension axiom scheme *for Φ, denoted by Φ-**COMP**, is the set of all formulas*

$$\exists X \leq y \forall z < y(X(z) \leftrightarrow \varphi(z)), \tag{2.2}$$

where $\varphi(z)$ is any formula in Φ, and X does not occur free in $\varphi(z)$.

Definition 2.11 ($\mathbf{V}^0$)**.** $\mathbf{V}^0$ *is the theory over $\mathcal{L}_A^2$ axiomatized by the sets* 2-**BASIC** *(Figure 2.1) and* $\mathbf{\Sigma}_0^B$-**COMP**.

It is known that $\mathbf{V}^0$ is finitely axiomatizable [CK03]. Therefore the theories that we introduce in Chapter 3 are all finitely axiomatizable.

Definition 2.12 (Number Induction Axiom)**.** *If Φ is a set of two-sorted formulas, then* Φ-**IND** *axioms are the formulas*

$$[\varphi(0) \wedge \forall x,\ \varphi(x) \supset \varphi(x+1)] \supset \forall z \varphi(z)$$

where φ is a formula in Φ.

Definition 2.13 (Number Minimization Axiom)**.** *The number minimization axioms (or* least number principle *axioms) for a set Φ of two-sorted formulas are denoted* Φ-**MIN** *and consist of the formulas*

$$\varphi(y) \supset \exists x \leq y(\varphi(x) \wedge \neg \exists z < x \varphi(z))$$

where φ is a formula in Φ.

Using the function $|X|$ it can be shown that $\mathbf{V}^0$ proves both $\mathbf{\Sigma}_0^B$-**IND** and $\mathbf{\Sigma}_0^B$-**MIN**. In fact we have:

Theorem 2.14. *Let $\mathcal{T}$ be an extension of $\mathbf{V}^0$ and Φ be a set of formulas in $\mathcal{T}$. Suppose that $\mathcal{T}$ proves the* Φ-**COMP** *axiom scheme. Then $\mathcal{T}$ also proves the* Φ-**IND** *and* Φ-**MIN**.

It follows that $\mathbf{V}^0$ extends $\mathbf{I\Delta}_0$. It is known, furthermore, that $\mathbf{V}^0$ is conservative over $\mathbf{I\Delta}_0$. See [CN06, Chapter 5] for a proof of these facts.

We can generalize the $\mathbf{\Sigma}_0^B$-comprehension axiom scheme to multiple dimensions. We use the pairing function $\langle x, y\rangle$ defined in (2.3), and write $\langle x_1, x_2, \ldots, x_k\rangle$ for $\langle x_1, \langle x_2, \langle \ldots \rangle\rangle\rangle$ and $X(\vec{x})$ for $X(\langle \vec{x} \rangle)$.

$$\langle x, y\rangle =_{\text{def}} (x+y)(x+y+1)+2y \tag{2.3}$$

Definition 2.15 (Multiple Comprehension Axiom)**.** *For a set Φ of formulas, the* multiple comprehension axiom scheme *for Φ, denoted by* Φ-**MULTICOMP**, *is the set of all formulas*

$$\exists X \leq \langle y_1, \ldots, y_k\rangle \forall z_1 < y_1 \ldots \forall z_k < y_k(X(z_1, \ldots, z_k) \leftrightarrow \varphi(z_1, \ldots, z_k)) \tag{2.4}$$

where $\varphi(z)$ is any formula in Φ which may contain other free variables, but not X.

The next lemma is straightforward:

Lemma 2.16. *Suppose that $\mathcal{T} \supseteq \mathbf{V}^0$ is a theory with vocabulary $\mathcal{L}$ which proves the $\mathbf{\Sigma}_0^B(\mathcal{L})$-**COMP** axioms. Then $\mathcal{T}$ proves the $\mathbf{\Sigma}_0^B(\mathcal{L})$-**MULTICOMP** axioms.*

Definition 2.17 (Two-Sorted Definability)**.** *Let $\mathcal{T}$ be a theory with vocabulary $\mathcal{L} \supseteq \mathcal{L}_A^2$, and Φ a set of $\mathcal{L}$-formulas. A number function f (not in $\mathcal{L}$) is* Φ-definable in $\mathcal{T}$ *if there is a formula $\varphi(\vec{x}, y, \vec{X})$ in Φ such that $\mathcal{T} \vdash \forall\vec{x}\forall\vec{X}\exists! y\varphi(\vec{x}, y, \vec{X})$ and*

$$y = f(\vec{x}, \vec{X}) \leftrightarrow \varphi(\vec{x}, y, \vec{X}) \tag{2.5}$$

A string function F (not in $\mathcal{L}$) is Φ-definable in $\mathcal{T}$ *if there is a formula $\varphi(\vec{x}, \vec{X}, Y)$ in Φ such that $\mathcal{T} \vdash \forall\vec{x}\forall\vec{X}\exists! Y\varphi(\vec{x}, \vec{X}, Y)$ and*

$$Y = F(\vec{x}, \vec{X}) \leftrightarrow \varphi(\vec{x}, \vec{X}, Y) \tag{2.6}$$

(2.5) *(resp.* (2.6)*) is a* defining axiom *for f (resp. F). We say that f (or F) is* definable in $\mathcal{T}$ *if it is Φ-definable in $\mathcal{T}$ for some Φ. Also, f (or F) is* provably total *in $\mathcal{T}$ iff it is $\mathbf{\Sigma}_1^1$-definable in $\mathcal{T}$.*

Example 2.18. *The function* $\log(x)$*, where* $\log(0) = 0$ *and* $\log(x) = \lfloor \log_2(x) \rfloor$ *if $x \geq 1$, is provably total in $\mathbf{V}^0$. This is because the relation $2^x = y$ is representable by a $\mathbf{\Delta}_0$ formula (Example 2.6).*

Theorem 2.19. a) *Let $\mathcal{T}$ be a theory and $\mathcal{T}'$ be the theory obtained from $\mathcal{T}$ by adding a definable function in $\mathcal{T}$ together with its defining axiom. Then $\mathcal{T}'$ is a conservative extension of $\mathcal{T}$.*

b) *Suppose that $\mathcal{T}_1, \mathcal{T}_2, \ldots$ is a sequence of theories where $\mathcal{T}_{i+1}$ is a conservative extension of $\mathcal{T}_i$ for $i \geq 1$. Then $\mathcal{T} = \bigcup_i \mathcal{T}_i$ is a conservative extension of $\mathcal{T}_1$.*

Proof Sketch. **a)** Every model of $\mathcal{T}$ can be expanded to a model of $\mathcal{T}'$.

b) Follow from **a)** by compactness. □

Using Theorem 2.5 it can be shown that the provably total functions of $\mathbf{V}^0$ are precisely $\mathbf{FAC}^0$. Now we define two $\mathbf{AC}^0$ functions that are useful for encoding sequences of strings and numbers.

Definition 2.20 (*Row* and *seq*). *The function $Row(x, Z)$ (also denoted $Z^{[x]}$) has the bit-defining axiom*

$$|Row(x, Z)| \leq |Z| \ \wedge \ (Row(x, Z)(i) \leftrightarrow i < |Z| \wedge Z(x, i)) \tag{2.7}$$

The number function $seq(x, Z)$ (also denoted $(Z)^x$) has the defining axiom:

$$y = seq(x, Z) \leftrightarrow (y < |Z| \wedge Z(x, y) \wedge \forall z < y \neg Z(x, z)) \vee (\forall z < |Z| \neg Z(x, z) \wedge y = |Z|)$$

2.4 $\overline{\mathbf{V}}^0$: A Universal Conservative Extension of $\mathbf{V}^0$

All theories $\overline{\mathbf{VC}}$ introduced in Chapter 3 are extensions of $\overline{\mathbf{V}}^0$ defined here.

To obtain a universal conservative extension of $\mathbf{V}^0$, the idea is to introduce Skolem functions that are definable in $\mathbf{V}^0$ in order to eliminate the quantifiers in the axioms of $\mathbf{V}^0$. First, we introduce some $\mathbf{AC}^0$ functions in order to eliminate the quantifiers in the 2-**BASIC** axioms. The existential quantifier in **B12** is eliminated using the predecessor function *pd*:

$$\textbf{B12}'.\ pd(0) = 0 \qquad\qquad \textbf{B12}''.\ x \neq 0 \supset pd(x) + 1 = x \tag{2.8}$$

The extensionality axiom **SE** contains an implicit existential quantifier $\exists i < |X|$. We introduce the function $f_{\mathbf{SE}}(X, Y)$ which is the smallest number $< |X|$ that distinguishes X and Y, and $|X|$ if no such number exists:

$$(f_{\mathbf{SE}}(X, Y) \leq |X|) \ \wedge \ (z < f_{\mathbf{SE}}(X, Y) \supset (X(z) \leftrightarrow Y(z))) \ \wedge$$
$$(f_{\mathbf{SE}}(X, Y) < |X| \supset (X(f_{\mathbf{SE}}(X, Y)) \not\leftrightarrow Y(f_{\mathbf{SE}}(X, Y)))) \tag{2.9}$$

(The defining axiom (2.9) is an instance of the axiom (2.12) below, where $\varphi(z, X, Y) \equiv X(z) \not\leftrightarrow Y(z)$, and $t(X, Y) = |X|$.) In $\overline{\mathbf{V}}^0$ the axiom **SE** is replaced by **SE**$'$:

$$(|X| = |Y| \wedge f_{\mathbf{SE}}(X, Y) = |X|) \supset X = Y. \tag{2.10}$$

Now we introduce other $\mathbf{AC}^0$ functions. For each formula $\varphi(z, \vec{x}, \vec{X})$ and $\mathcal{L}_A^2$-term $t(\vec{x}, \vec{X})$, let $F_{\varphi,t}(\vec{x}, \vec{X})$ be the string function with bit definition

$$F_{\varphi,t}(\vec{x}, \vec{X})(z) \leftrightarrow z < t(\vec{x}, \vec{X}) \wedge \varphi(z, \vec{x}, \vec{X}) \tag{2.11}$$

Also, let $f_{\varphi,t}(\vec{x}, \vec{X})$ be the least $y < t$ such that $\varphi(y, \vec{x}, \vec{X})$ holds, or t if no such y exists. Then $f_{\varphi,t}$ has defining axiom (we write f for $f_{\varphi,t}$, t for $t(\vec{x}, \vec{X})$, and $\ldots$ for $\vec{x}, \vec{X}$):

$$f(\ldots) \leq t \ \wedge \ [v < f(\ldots) \supset \neg\varphi(v, \ldots)] \ \wedge [f(\ldots) < t \supset \varphi(f(\ldots), \ldots)] \tag{2.12}$$

Definition 2.21. *$\mathcal{L}_{\mathbf{FAC}^0}$ is the smallest set that satisfies*

1) *$\mathcal{L}_{\mathbf{FAC}^0}$ includes $\mathcal{L}_A^2 \cup \{pd, f_{\mathbf{SE}}\}$.*
2) *For each open formula $\varphi(z, \vec{x}, \vec{X})$ over $\mathcal{L}_{\mathbf{FAC}^0}$ and term $t = t(\vec{x}, \vec{X})$ of $\mathcal{L}_A^2$ there is a string function $F_{\varphi,t}$ and a number function $f_{\varphi,t}$ in $\mathcal{L}_{\mathbf{FAC}^0}$.*

Definition 2.22. *$\overline{\mathbf{V}}^0$ is the theory over $\mathcal{L}_{\mathbf{FAC}^0}$ with the following set of axioms:* **B1**-**B11***,* **L1***,* **L2** *(Figure 2.1),* (2.8)*,* (2.9)*,* (2.10)*, and* (2.11) *for each function $F_{\varphi,t}$ and* (2.12) *for each function $f_{\varphi,t}$ of $\mathcal{L}_{\mathbf{FAC}^0}$.*

The next lemma is straightforward:

Lemma 2.23. a) *For every $\mathbf{\Sigma}_0^B(\mathcal{L}_{\mathbf{FAC}^0})$ formula φ there is an open $\mathcal{L}_{\mathbf{FAC}^0}$-formula φ^+ such that $\overline{\mathbf{V}}^0 \vdash \varphi \leftrightarrow \varphi^+$.*

b) *For every $\mathbf{\Sigma}_0^B(\mathcal{L}_{\mathbf{FAC}^0})$ formula φ there is a $\mathbf{\Sigma}_0^B(\mathcal{L}_A^2)$ formula φ' such that $\overline{\mathbf{V}}^0 \vdash \varphi \leftrightarrow \varphi'$.*

Theorem 2.24. *$\overline{\mathbf{V}}^0$ is a conservative extension of $\mathbf{V}^0$.*

Proof. Let $\varphi(x)$ be a $\mathbf{\Sigma}_0^B$ formula. By Lemma 2.23 **a)** there is an open $\mathcal{L}_{\mathbf{FAC}^0}$-formula $\varphi^+(x)$ such that $\overline{\mathbf{V}}^0 \vdash \varphi(x) \leftrightarrow \varphi^+(x)$. Now the string function $F_{\varphi^+,y}$ satisfies the comprehension axiom (2.2) for φ. In other words, $\overline{\mathbf{V}}^0$ proves $\mathbf{\Sigma}_0^B$-**COMP**. Hence $\overline{\mathbf{V}}^0$ extends $\mathbf{V}^0$. The conservativity follows from Lemma 2.23 **b)**. (See [CN06, Section 5.6].) $\square$

Chapter 3

Theories for Small Classes

We start by defining $\mathbf{VTC}^0$ and $\overline{\mathbf{VTC}}^0$ and stating the results for these two theories in Section 3.1. The proofs are given in Section 3.2 for the general framework where we develop theories $\mathbf{VC}$ and $\overline{\mathbf{VC}}$ for subclasses $\mathbf{C}$ of $\mathbf{P}$. The last step for applying the general framework to $\mathbf{VTC}^0$ is proved in Section 3.2.4. The subsequent sections define other theories which are instances of $\mathbf{VC}$. In Section 3.3 we define the theories $\mathbf{V}^0(m)$ and their union $\mathbf{VACC}$. In Section 3.4 we define $\mathbf{VNC}^1$ and prove that $\mathbf{VTC}^0 \subseteq \mathbf{VNC}^1$; the proof of this inclusion is a formalization of Buss's [Bus87b] arguments which give $\mathbf{NC}^1$ circuits that compute the function *numones*. The theories $\mathbf{VNL}$ and $\mathbf{VL}$ are introduced in Sections 3.5 and 3.6 respectively. In Section 3.7 we define $\mathbf{VP}$ and show that $\mathbf{VP} = \mathbf{TV}^0$. Finally in Section 3.8 we discuss theories for classes in the $\mathbf{AC}^k$, $\mathbf{NC}^k$ hierarchies.

3.1 $\mathbf{VTC}^0$

We define $\mathbf{VTC}^0$ and the universal theory $\overline{\mathbf{VTC}}^0$ over the language of $\mathbf{FTC}^0$ functions. In Section 3.2 we will introduce a scheme of theories $\mathbf{VC}$ and $\overline{\mathbf{VC}}$ (where $\overline{\mathbf{VC}}$ is a universal theory) and prove that $\overline{\mathbf{VC}}$ is conservative over $\mathbf{VC}$. It follows from Lemma 3.18 in Section 3.2.4 that $\mathbf{VTC}^0$ and $\overline{\mathbf{VTC}}^0$ are instances of $\mathbf{VC}$ and $\overline{\mathbf{VC}}$, respectively.

So $\overline{\mathbf{VTC}}^0$ is a conservative extension of $\mathbf{VTC}^0$, and this implies that the provably total functions of $\mathbf{VTC}^0$ are precisely $\mathbf{FTC}^0$ (see Theorem 3.11).

First, $numones(z, X)$ is the number of elements of X that are $< z$:

Definition 3.1. *$numones(z, X)$ is the number function with defining axioms:*

$$numones(0, X) = 0 \tag{3.1}$$

$$X(z) \supset numones(z + 1, X) = numones(z, X) + 1 \tag{3.2}$$

$$\neg X(z) \supset numones(z + 1, X) = numones(z, X). \tag{3.3}$$

Proposition 3.2. *$\mathbf{TC}^0$ is the $\mathbf{AC}^0$ closure of numones. $\mathbf{FTC}^0$ is the $\mathbf{FAC}^0$ closure of numones.*

Proof Sketch. The fact that $\mathbf{TC}^0$ is the $\mathbf{AC}^0$ closure of *numones* can be proved by induction using the fact [BIS90] that $\mathbf{TC}^0 = \mathbf{FO}(M)$, i.e., $\mathbf{TC}^0$ is the class of relations that are expressible by first-order formulas with the majority quantifiers. The second half of the proposition follows from the first and Lemma 2.9. □

The theory $\mathbf{VTC}^0$ is axiomatized by $\mathbf{V}^0$ and a $\mathbf{\Sigma}_1^B$ defining axiom for *numones*. Recall that $(Y)^z$ is the z-th element of the bounded sequence of numbers coded by Y (Definition 2.20). In the formula δ_{NUM} below, Y encodes a computation of $numones(x, X)$: for $z \leq x$, $(Y)^z = numones(z, X)$.

Definition 3.3 ($\mathbf{VTC}^0$)**.** *$\mathbf{VTC}^0$ is the theory over $\mathcal{L}_A^2$ that is axiomatized by $\mathbf{V}^0$ and $NUMONES \equiv \forall X \forall x \exists Y \delta_{NUM}(x, X, Y)$, where*

$$\begin{aligned}\delta_{NUM}(x, X, Y) \equiv\ & (Y)^0 = 0 \wedge \\ & \forall z < x,\ (X(z) \supset (Y)^{z+1} = (Y)^z + 1) \wedge (\neg X(z) \supset (Y)^{z+1} = (Y)^z)\end{aligned} \tag{3.4}$$

Theorem 3.4 (Definability Theorem for $\mathbf{VTC}^0$)**.** *A function is in $\mathbf{FTC}^0$ if and only if it is provably total in $\mathbf{VTC}^0$.*

Below we will introduce $\overline{\mathbf{VTC}}^0$, a universal theory that contains all $\mathbf{TC}^0$ functions and their defining axioms (based on the fact that $\mathbf{FTC}^0$ is the $\mathbf{AC}^0$ closure of *numones*, see Proposition 3.2). The Definability Theorem for $\mathbf{VTC}^0$ follows from Theorem 3.7 below (see the proof of Theorem 3.8).

Recall the functions pd, $f_{\mathbf{SE}}$ and the notations $f_{\varphi,t}, F_{\varphi,t}$ from Section 2.4.

Definition 3.5 ($\mathcal{L}_{\mathbf{FTC}^0}$). *$\mathcal{L}_{\mathbf{FTC}^0}$ is the smallest set that satisfies*

1) *$\mathcal{L}_{\mathbf{FTC}^0}$ includes $\mathcal{L}^2_A \cup \{pd, f_{\mathbf{SE}}, numones\}$*
2) *For each open formula $\varphi(z, \vec{x}, \vec{X})$ over $\mathcal{L}_{\mathbf{FTC}^0}$ and term $t = t(\vec{x}, \vec{X})$ of $\mathcal{L}^2_A$, there is a string function $F_{\varphi,t}$ and a number function $f_{\varphi,t}$ in $\mathcal{L}_{\mathbf{FTC}^0}$.*

Definition 3.6. $\overline{\mathbf{VTC}}^0$ *is the theory over $\mathcal{L}_{\mathbf{FTC}^0}$ with the following quantifier-free axioms:* **B1**–**B11**, **L1**, **L2** *(Figure 2.1),* (2.8), (2.9), (2.10), *the defining axioms* (3.1), (3.2) *and* (3.3) *for numones, and* (2.11) *for each function $F_{\varphi,t}$ and* (2.12) *for each function $f_{\varphi,t}$ of $\mathcal{L}_{\mathbf{FTC}^0}$.*

The Definability Theorem for $\mathbf{VTC}^0$ follows from the next theorem:

Theorem 3.7. $\overline{\mathbf{VTC}}^0$ *is a conservative extension of* $\mathbf{VTC}^0$. *A function is in* $\mathbf{FTC}^0$ *if and only if it is $\Sigma^B_1(\mathcal{L}^2_A)$-definable in* $\overline{\mathbf{VTC}}^0$.

It is rather straightforward to show that $\overline{\mathbf{VTC}}^0$ extends $\mathbf{VTC}^0$. However, proving that $\overline{\mathbf{VTC}}^0$ is conservative over $\mathbf{VTC}^0$ is not as easy as proving that $\overline{\mathbf{V}}^0$ is conservative over $\mathbf{V}^0$ (see Theorem 2.24). This is because we do not know whether every open formula of $\mathcal{L}_{\mathbf{FTC}^0}$ is equivalent in $\overline{\mathbf{VTC}}^0$ to a $\Sigma^B_0(numones)$ formula. (If this is indeed the case, then the languages in $\mathbf{TC}^0$ would be computable by threshold circuits where the nesting depth of the threshold gates are bounded by some constant. It would then be easy to show that the functions in $\mathcal{L}_{\mathbf{FTC}^0}$ are definable in $\mathbf{VTC}^0$.) In the next section we prove the above theorem in a more general setting that applies to many other classes. The proof of Theorem 3.7 is completed in Section 3.2.4.

3.2 Theories for other Subclasses of P

Consider a polytime function F, and let $\mathbf{C}$ be the class of two-sorted relations which are $\mathbf{AC}^0$-reducible to F. Then $\mathbf{FC}$ is the set of functions $\mathbf{AC}^0$-reducible to F (Lemma 2.9). Our goal is to develop a theory $\mathbf{VC}$ that characterizes $\mathbf{C}$.

Suppose that $F(X)$ has a defining axiom

$$F(X) = Y \leftrightarrow (|Y| \leq t \wedge \varphi(X, Y)) \tag{3.5}$$

for some term t and $\mathbf{\Sigma}_0^B(\mathcal{L}_A^2)$ formula φ. Suppose also that

$$\mathbf{V}^0 \vdash \forall Y_1 \forall Y_2 (|Y_1| \leq t \wedge |Y_2| \leq t \wedge \varphi(X, Y_1) \wedge \varphi(X, Y_2) \supset Y_1 = Y_2)$$

Notice that δ_{NUM} (3.4) can be seen as a special case of φ.

The theory $\mathbf{VC}$ has vocabulary $\mathcal{L}_A^2 \cup \{Row\}$ and is axiomatized by $\mathbf{V}^0(Row)$ and the following axiom (which is really a defining axiom for the function $F^\star$, see Section 3.2.3):

$$\forall b \forall X \exists Y \forall u < b \varphi(X^{[u]}, Y^{[u]}) \tag{3.6}$$

Our main result of this chapter is the following theorem, which follows from Theorem 3.11. In Sections 3.3–3.8 we introduce instances of $\mathbf{VC}$ that are associated with the remaining classes in (1.1). Theorem 3.8 serves as a meta-theorem that applies for each of these theories.

Theorem 3.8 (Definability Theorem for $\mathbf{VC}$). *A function is provably total in* $\mathbf{VC}$ *iff it is in* $\mathbf{FC}$.

How do we obtain a function F and its defining axiom (3.6) for each class in (1.1)? We will address this issue before proving the above theorem.

3.2.1 Obtaining Theories for the Classes in (1.1)

It turns out that for each class $\mathbf{C}$ of interest, there is a polytime Turing machine M such that the function

$$F_{\mathsf{M}}(X) = \text{“the computation of } \mathsf{M} \text{ on input } X\text{”}$$

is complete for $\mathbf{C}$. For the case of $\mathbf{TC}^0$, M is the machine $\mathsf{M}_{numones}$ that computes $numones(|X|, X)$ by computing $numones(z, X)$ inductively on z, and $F_{\mathsf{M}}(X)$ is essentially the string Y in (3.4) (page 26).

The $\mathbf{\Sigma}_0^B(\mathcal{L}_A^2)$ defining axiom (3.5) for F_{M} can be obtained using the following $\mathbf{AC}^0$ functions (whose presence in a $\mathbf{\Sigma}_0^B$ formula can be eliminated using their $\mathbf{\Sigma}_0^B$ bit definitions):

- $Init_{\mathsf{M}}(X)$ is the initial configuration of M given input X,
- $Next_{\mathsf{M}}(U)$ is the next configuration of the configuration U, and
- $Cut(t, Z)$ is the set of all elements of Z that are less than t:

$$Cut(t, Z) = \{z : z \in Z \wedge z < t\} \tag{3.7}$$

Let t be an $\mathcal{L}_A^2$ term that bounds the running time of M. We have

$$F(X) = Y \leftrightarrow |Y| \leq \langle t, t \rangle \wedge Y^{[0]} = Cut(t, Init_{\mathsf{M}}(X)) \wedge \\ \forall x < t,\ Y^{[x+1]} = Cut(t, Next_{\mathsf{M}}(Y^{[x]}))$$

3.2.2 The Theory $\overline{\mathbf{VC}}$

The language $\mathcal{L}_{\mathbf{FC}}$ is the smallest set containing $\mathcal{L}_{\mathbf{FAC}^0} \cup \{F\}$ and satisfying the following condition: for each open formula $\varphi(z, \vec{x}, \vec{X})$ over $\mathcal{L}_{\mathbf{FC}}$ and term $t = t(\vec{x}, \vec{X})$ of $\mathcal{L}_A^2$, there is a string function $F_{\varphi,t}$ and a number function $f_{\varphi,t}$ in $\mathcal{L}_{\mathbf{FC}}$.

Note that by Lemma 2.23 **a)** the $\mathbf{\Sigma}_0^B$ defining axiom (3.5) for F is equivalent in $\overline{\mathbf{V}}^0$ to a quantifier-free formula over $\mathcal{L}_{\mathbf{FAC}^0}$.

Notation Let φ_F denote the quantifier-free $\mathcal{L}_{\mathbf{FAC}^0}$-formula that is equivalent (in $\overline{\mathbf{V}}^0$) to the defining axiom (3.5) of F, as stated in Lemma 2.23 **a**.

Definition 3.9. $\overline{\mathbf{VC}}$ *is the extension of* $\overline{\mathbf{V}}^0$ *with the additional axioms* $F(X) = Y \leftrightarrow \varphi_F$ *and* (2.11)/(2.12) *for each (new) function* $F_{\varphi,t}/f_{\varphi,t}$ *of* $\mathcal{L}_{\mathbf{FC}}$.

Lemma 3.10. $\overline{\mathbf{VC}}$ *extends* $\mathbf{VC}$*.*

Proof. Each $\mathbf{\Sigma}_0^B(\mathcal{L}_{\mathbf{FC}})$ formula φ is equivalent in $\overline{\mathbf{VC}}$ to an open formula φ' of $\mathcal{L}_{\mathbf{FC}}$. So the string X in the comprehension axiom (2.2) for φ can be taken to be $F_{\varphi',t}$ for a suitable $\mathcal{L}_A^2$ term t. Hence $\overline{\mathbf{VC}} \vdash \mathbf{\Sigma}_0^B(\mathcal{L}_{\mathbf{FC}})$-**COMP**, and therefore $\overline{\mathbf{VC}}$ extends $\mathbf{V}^0$.

The fact that $\overline{\mathbf{VC}} \vdash \mathbf{\Sigma}_0^B(\mathcal{L}_{\mathbf{FC}})$-**COMP** also shows that (3.6) is provable in $\overline{\mathbf{VC}}$: Take Y such that $|Y| \leq \langle b, t\rangle \wedge (Y(u,i) \leftrightarrow F(X^{[u]})(i))$. As a result, $\overline{\mathbf{VC}}$ extends $\mathbf{VC}$. □

Theorem 3.8 follows from the following theorem, which in turn follows from Corollary 3.17.

Theorem 3.11. a) $\overline{\mathbf{VC}}$ *is a conservative extension of* $\mathbf{VC}$*.*
b) *The functions of* $\mathcal{L}_{\mathbf{FC}}$ *are* $\mathbf{\Sigma}_1^B(\mathcal{L}_A^2)$*-definable in* $\overline{\mathbf{VC}}$*.*

Proof of Definability Theorem for $\mathbf{VC}$*.* The fact that each function in $\mathbf{FC}$ is $\mathbf{\Sigma}_1^1$-definable in $\mathbf{VC}$ follows immediately from Theorem 3.11. For the other direction, suppose that a string function $F(\vec{x}, \vec{X})$ is $\mathbf{\Sigma}_1^1$-definable in $\mathbf{VC}$. (The case of a number function is similar.) So there is a $\mathbf{\Sigma}_1^1$ formula $\exists \vec{Y} \varphi(\vec{x}, \vec{X}, \vec{Y}, Z)$, where φ is a $\mathbf{\Sigma}_0^B$ formula, so that (see Definition 2.17)

$$F(\vec{x}, \vec{X}) = Z \leftrightarrow \exists \vec{Y} \varphi(\vec{x}, \vec{X}, \vec{Y}, Z)$$

and that

$$\mathbf{VC} \vdash \forall \vec{x} \forall \vec{X} \exists! Z \exists \vec{Y} \varphi(\vec{x}, \vec{X}, \vec{Y}, Z)$$

By Lemma 2.23 (and because $\overline{\mathbf{VC}}$ extends $\overline{\mathbf{V}}^0$) there is an open $\mathcal{L}_{\mathbf{FAC}^0}$-formula ψ so that

$$\overline{\mathbf{VC}} \vdash \varphi(\vec{x}, \vec{X}, \vec{Y}, Z) \leftrightarrow \psi(\vec{x}, \vec{X}, \vec{Y}, Z)$$

Hence (using Theorem 3.11 **a**) we have

$$\overline{\mathbf{VC}} \vdash \forall \vec{x} \forall \vec{X} \exists! Z \exists \vec{Y} \psi(\vec{x}, \vec{X}, \vec{Y}, Z)$$

Now by Herbrand's Theorem, the existence of Z and $\vec{Y}$ is witnessed by some functions from $\mathcal{L}_{\mathbf{FC}}$. In particular, F is a function in $\mathbf{FC}$. □

3.2.3 Aggregate Functions

Now we set out to prove Theorem 3.11. First, consider part **a**. (Part **b** will follow from Theorem 3.16 below.) Let $\mathcal{L}_1 = \mathcal{L}_{\mathbf{FAC}^0} \cup \{F\}$, and for $n \geq 1$, $\mathcal{L}_{n+1}$ be obtained from $\mathcal{L}_n$ by adding the functions $f_{\varphi,t}$ and $F_{\varphi,t}$ for each open formula φ of $\mathcal{L}_n$ and $\mathcal{L}_A^2$ term t. For $n \geq 1$ let $\mathcal{T}_n$ be the extension of **VC** obtained by adding the functions in $\mathcal{L}_n$ and their defining axioms (specified in Definition 3.9). Because $\overline{\mathbf{VC}}$ extends **VC** (Lemma 3.10), we have

$$\overline{\mathbf{VC}} = \bigcup_{n\geq 1} \mathcal{T}_n$$

Thus, to show that $\overline{\mathbf{VC}}$ is conservative over **VC**, by Theorem 2.19 **b)** it suffices to show that for $n \geq 1$:

$$\mathcal{T}_{n+1} \text{ is a conservative extension of } \mathcal{T}_n \tag{3.8}$$

By Theorem 2.19 **a)**, to prove (3.8) it suffices to show that the new functions $f_{\varphi,t}$, $F_{\varphi,t}$ in $\mathcal{L}_{n+1}$ are definable in $\mathcal{T}_n$. The graph of each new function $f_{\varphi,t}$ is an open formula of $\mathcal{L}_n$, so to prove the definability of $f_{\varphi,t}$ in $\mathcal{T}_n$ it suffices to show that $\mathcal{T}_n \vdash \mathbf{\Sigma}_0^B(\mathcal{L}_n)$-**MIN**. Similarly, to prove the definability of each new function $F_{\varphi,t}$ it suffices to show that $\mathcal{T}_n \vdash \mathbf{\Sigma}_0^B(\mathcal{L}_n)$-**COMP**. Thus, using Theorem 2.14, (3.8) follows from:

$$\mathcal{T}_n \vdash \mathbf{\Sigma}_0^B(\mathcal{L}_n)\text{-}\mathbf{COMP} \tag{3.9}$$

The idea is to prove (3.9) by induction on n. It turns out that we need a slightly stronger induction hypothesis, which is stated using the notion of *aggregate functions* defined below. Informally, for a string function F (or a number function f), the aggregate function $F^\star$ (resp. $f^\star$), is the string function that gathers the values of F (resp. f) for a polynomially long sequence of arguments. Recall the functions *Row* and *seq* from Definition 2.20.

Definition 3.12 (Aggregate Function). *Let $F(x_1, \ldots, x_k, X_1, \ldots, X_n)$ be a string func-*

tion. Then $F^\star(b, Z_1, \ldots, Z_k, X_1, \ldots, X_n)$ *is the set*

$$\{\langle u, v\rangle \ : \ u < b \wedge v \in F((Z_1)^u, \ldots, (Z_k)^u, X_1^{[u]}, \ldots, X_n^{[u]})\}$$

Similarly, for a number function $f(x_1, \ldots, x_k, X_1, \ldots, X_n)$*,*

$$f^\star(b, \vec{Z}, \vec{X}) = \{\langle u, f((Z_1)^u, \ldots, (Z_k)^u, X_1^{[u]}, \ldots, X_n^{[u]})\rangle \ : \ u < b\}$$

Notice that if F (or f) is polynomially bounded, so is $F^\star$ (resp. $f^\star$). Universal defining axioms for $F^\star$ and $f^\star$ are as follows:

$$F^\star(b, \vec{Z}, \vec{X})(u, v) \ \leftrightarrow \ u < b \wedge v < |F(\overrightarrow{(Z)^u}, \overrightarrow{X^{[u]}})| \wedge F(\overrightarrow{(Z)^u}, \overrightarrow{X^{[u]}})(v) \tag{3.10}$$

$$f^\star(b, \vec{Z}, \vec{X})(u, v) \ \leftrightarrow \ u < b \wedge v = f(\overrightarrow{(Z)^u}, \overrightarrow{X^{[u]}}) \tag{3.11}$$

Example 3.13 (*numones**)**.**

$$\begin{aligned} \mathit{numones}^\star(b, Z, X) = Y \ \leftrightarrow \ & (|Y| \le \langle b, 1 + |X|\rangle \wedge \\ & \forall w < \langle b, 1 + |X|\rangle, Y(w) \leftrightarrow \exists u < b, \ w = \langle u, \mathit{numones}((Z)^u, X^{[u]})\rangle) \end{aligned} \tag{3.12}$$

In Lemma 3.18, we will show that numones* *is provably total in* $\mathbf{VTC}^0$*.*

The function *seq* can be eliminated from (3.10) and (3.11) using its defining axiom (see Definition 2.20). For the rest of this section, let $\mathcal{T}$ be a theory over $\mathcal{L}$, where

$$\mathcal{L}_A^2 \cup \{\mathit{Row}\} \subseteq \mathcal{L}, \qquad \mathcal{T} \vdash \mathbf{\Sigma}_0^B(\mathcal{L})\text{-}\mathbf{COMP}, \qquad \text{and} \qquad \mathcal{T} \text{ extends } \mathbf{V}^0(\mathit{Row}) \tag{3.13}$$

Also, we will be interested in whether F (resp. f) satisfy

$$\text{both } F \text{ and } F^\star \text{ are } \mathbf{\Sigma}_1^B\text{-definable in } \mathcal{T} \text{ and } \mathcal{T}(F, F^\star) \text{ proves (3.10)} \tag{3.14}$$

$$\text{(resp. both } f \text{ and } f^\star \text{ are } \mathbf{\Sigma}_1^B\text{-definable in } \mathcal{T} \text{ and } \mathcal{T}(f, f^\star) \text{ proves (3.11))} \tag{3.15}$$

Lemma 3.14. *Let* $\mathcal{T}$ *and* $\mathcal{L}$ *be as in* (3.13)*. Let* F *(or* f*) be a function* $\mathbf{\Sigma}_0^B$*-definable from* $\mathcal{L}$ *(recall Definition 2.7). Then the function* $F^\star$ *(or* $f^\star$*) is* $\mathbf{\Sigma}_0^B$*-definable from* $\mathcal{L}$*. In addition,* (3.14) *holds (resp.* (3.15) *holds). In fact, both* F *and* $F^\star$ *(resp.* f *and* $f^\star$*) are* $\mathbf{\Sigma}_0^B(\mathcal{L})$*-definable (and hence provably total) in* $\mathcal{T}$*.*

Proof. The fact that $F^\star$ (resp. $f^\star$) is $\mathbf{\Sigma}_0^B$-definable from $\mathcal{L}$ is obvious. The definability of F and $F^\star$ (resp. f and $f^\star$) in $\mathcal{T}$ follows from the fact that $\mathcal{T}$ proves multiple comprehension for $\mathbf{\Sigma}_0^B(\mathcal{L})$ formulas (by (3.13) and Lemma 2.16). For example, suppose that $f(\vec{x}, \vec{X})$ is bounded by t and has a $\mathbf{\Sigma}_0^B(\mathcal{L})$ graph $\varphi(\vec{x}, y, \vec{X})$. Then f can be defined in $\mathcal{T}$ by first defining using $\mathbf{\Sigma}_0^B(\mathcal{L})$-**COMP** the set Y such that

$$|Y| \leq t + 1 \wedge \forall y < t + 1, Y(y) \leftrightarrow \varphi(\vec{x}, y, \vec{X})$$

Now $y = |Y| \dot{-} 1$. $\square$

The next theorem is useful in proving the induction step of (3.9). The condition in (3.14) (resp. (3.15)) that F and $F^\star$ (resp. f and $f^\star$) be $\mathbf{\Sigma}_1^B$-definable in $\mathcal{T}$ can be replaced by the (weaker) condition that they are p-bounded and definable in $\mathcal{T}$.

Theorem 3.15. *Let $\mathcal{T}$, $\mathcal{L}$ and F (resp. f) be as in* (3.13) *and* (3.14) *(resp.* (3.15)*). Then $\mathcal{T}(F)$ proves $\mathbf{\Sigma}_0^B(\mathcal{L} \cup \{F\})$-**COMP** (resp. $\mathcal{T}(f)$ proves $\mathbf{\Sigma}_0^B(\mathcal{L} \cup \{f\})$-**COMP**).*

Proof. We will consider the case of extending $\mathcal{L}$ by a string function F. The case where $\mathcal{L}$ is extended by a number function is handled similarly by using number variables w_i instead of the string variables W_i in the argument below.

First, since $\mathcal{T}$ proves $\mathbf{\Sigma}_0^B(\mathcal{L})$-**COMP**, by Lemma 2.16 it proves the Multiple Comprehension axioms for $\mathbf{\Sigma}_0^B(\mathcal{L})$ formulas.

Claim For any $\mathcal{L}$-terms $\vec{s}, \vec{T}$ that contain variables $\vec{z}$, $\mathcal{T}(F)$ proves

$$\exists Y \forall z_1 < b_1 \ldots \forall z_m < b_m,\ Y^{[\vec{z}]} = F(\vec{s}, \vec{T}) \tag{3.16}$$

Proof of the Claim. Since $\mathcal{T}$ proves the Multiple Comprehension axiom scheme for $\mathbf{\Sigma}_0^B(\mathcal{L})$ formulas, it proves the existence of $\vec{X}$ such that $X_j^{[\vec{z}]} = T_j$, for $1 \leq j \leq n$. It also proves the existence of Z_i such that $(Z_i)^{\langle\vec{z}\rangle} = s_i$, for $1 \leq i \leq k$. Now the value of Y that satisfies (3.16) is just $F^\star(\langle\vec{b}\rangle, \vec{Z}, \vec{X})$. $\square$

Let $\mathcal{L}' = \mathcal{L} \cup \{F\}$. We show by induction on the quantifier depth of a $\mathbf{\Sigma}_0^B(\mathcal{L}')$ formula ψ that $\mathcal{T}(F)$ proves

$$\exists Z \le \langle b_1, \ldots, b_m \rangle \forall z_1 < b_1 \ldots \forall z_m < b_m,\ Z(\vec{z}) \leftrightarrow \psi(\vec{z}) \tag{3.17}$$

where $\vec{z}$ are all free number variables of ψ. It follows that $\mathcal{T}(F) \vdash \mathbf{\Sigma}_0^B(\mathcal{L}')$-**COMP**.

For the base case, ψ is quantifier-free. The idea is to replace every occurrence of a term $F(\vec{s}, \vec{T})$ in ψ by a new string variable W which has the intended value of $F(\vec{s}, \vec{T})$. The resulting formula is $\mathbf{\Sigma}_0^B(\mathcal{L})$, and we can apply the hypothesis.

Formally, suppose that $F(\vec{s}_1, \vec{T}_1), \ldots, F(\vec{s}_k, \vec{T}_k)$ are all occurrences of F in ψ. Note that the terms $\vec{s}_i, \vec{T}_i$ may contain $\vec{z}$ as well as nested occurrences of F. Assume further that $\vec{s}_1, \vec{T}_1$ do not contain F, and for $1 < i \le k$, any occurrence of F in $\vec{s}_i, \vec{T}_i$ must be of the form $F(\vec{s}_j, \vec{T}_j)$, for some $j < i$. We proceed to eliminate F from ψ by using its defining axiom.

Let $W_1, ..., W_k$ be new string variables. Let $\overrightarrow{s_1'} = \vec{s}_1$, $\overrightarrow{T_1'} = \vec{T}_1$, and for $2 \le i \le k$, $\overrightarrow{s_i'}$ and $\overrightarrow{T_i'}$ be obtained from $\vec{s}_i$ and $\vec{T}_i$ respectively by replacing every maximal occurrence of any $F(\vec{s}_j, \vec{T}_j)$, for $j < i$, by $W_j^{[\vec{z}]}$. Thus F does not occur in any $\overrightarrow{s_i'}$ and $\overrightarrow{T_i'}$, but for $i \ge 2$, $\overrightarrow{s_i'}$ and $\overrightarrow{T_i'}$ may contain $W_1, \ldots, W_{i-1}$.

By the claim above, for $1 \le i \le k$, $\mathcal{T}(F)$ proves the existence of W_i such that

$$\forall z_1 < b_1 \ldots \forall z_m < b_m,\ W_i^{[\vec{z}]} = F(\overrightarrow{s_i'}, \overrightarrow{T_i'}) \tag{3.18}$$

Let $\psi'(\vec{z}, W_1, \ldots, W_k)$ be obtained from $\psi(\vec{z})$ by replacing each maximal occurrence of $F(\vec{s}_i, \vec{T}_i)$ by $W_i^{[\vec{z}]}$, for $1 \le i \le k$. Then, by Multiple Comprehension for $\mathbf{\Sigma}_0^B(\mathcal{L})$ and the fact that $\mathcal{L}$ contains *Row*,

$$\mathcal{T} \vdash \exists Z \le \langle b_1, \ldots, b_m \rangle \forall z_1 < b_1 \ldots \forall z_m < b_m,\ Z(\vec{z}) \leftrightarrow \psi'(\vec{z}, W_1, \ldots, W_k).$$

Such Z satisfies (3.17) when each W_i is defined by (3.18).

The induction step is straightforward. Consider for example the case $\psi(\vec{z}) \equiv \forall x < t \lambda(\vec{z}, x)$. By the induction hypothesis,

$$\mathcal{T}(F) \vdash \exists Z' \forall z_1 < b_1 \ldots \forall z_m < b_m \forall x < t,\ Z'(\vec{z}, x) \leftrightarrow \lambda(\vec{z}, x).$$

Now, by Lemma 2.16

$$\mathbf{V}^0 \vdash \exists Z \forall z_1 < b_1 \ldots \forall z_m < b_m,\ Z(\vec{z}) \leftrightarrow \forall x < t Z'(\vec{z}, x).$$

Thus $\mathcal{T}(F) \vdash \exists Z \forall \vec{z} < \vec{b}\, Z(\vec{z}) \leftrightarrow \psi(\vec{z})$. □

Part **a)** of Theorem 3.11 follows from Lemma 3.14 and Theorem 3.15 (see the proof below). The next theorem is to prove Theorem 3.11 part **b**. Here we are interested in triples $\langle \mathcal{T}, \mathcal{L}, \mathcal{L}' \rangle$ such that (we will often have $\mathcal{L}' = \mathcal{L}_A^2$)

$$\text{for each } \mathbf{\Sigma}_0^B(\mathcal{L}) \text{ formula } \theta \text{ there is a } \mathbf{\Sigma}_1^1(\mathcal{L}') \text{ formula } \eta \text{ such that } \mathcal{T} \vdash \theta \leftrightarrow \eta \tag{3.19}$$

Theorem 3.16. *Let $\mathcal{T}$, $\mathcal{L}$ and F (resp. f) satisfy* (3.13) *and* (3.14) *(resp.* (3.15)*). Suppose that $\mathcal{L}_A^2 \subseteq \mathcal{L}' \subseteq \mathcal{L}$ such that* (3.19) *holds. Then* (3.19) *holds for $\langle \mathcal{T}(F), \mathcal{L} \cup \{F\}, \mathcal{L}' \rangle$ (resp. $\langle \mathcal{T}(f), \mathcal{L} \cup \{f\}, \mathcal{L}' \rangle$).*

Proof. We prove for the case of the string function F. The case for the number function f is similar. Suppose that

$$\theta \equiv \mathrm{Q}_1 z_1 < r_1 \ldots \mathrm{Q}_n z_n < r_n \psi(\vec{z})$$

is a $\mathbf{\Sigma}_0^B(\mathcal{L}, F)$ formula, where $Q_1, \ldots, Q_n \in \{\exists, \forall\}$ and ψ is a quantifier-free formula. Let $\vec{s}_i$, $\vec{T}_i$, $\overrightarrow{s_i'}$, $\overrightarrow{T_i'}$ and $\psi'(\vec{z}, W_1, \ldots, W_k)$ be as described in the proof of Theorem 3.15, and let λ_i be the formula (3.18) for $1 \leq i \leq k$. Define

$$\theta'(W_1, ..., W_k) \equiv \mathrm{Q}_1 z_1 < r_1 \ldots \mathrm{Q}_n z_n < r_n \psi'(\vec{z}, W_1, ..., W_k).$$

Then, θ is equivalent in $\mathcal{T}(F)$ to

$$\exists W_1 \ldots \exists W_k,\ ((\bigwedge \lambda_i) \wedge \theta'(W_1, \ldots, W_k))$$

By the given assumption that each $\mathbf{\Sigma}_0^B(\mathcal{L})$ is equivalent in $\mathcal{T}$ to a $\mathbf{\Sigma}_1^1(\mathcal{L}')$ formula, we may replace the whole matrix of the formula above by a $\mathbf{\Sigma}_1^B(\mathcal{L}')$ formula. □

Corollary 3.17. *Let $\mathcal{T}_0, \mathcal{L}_0, \mathcal{L}'$ be such that $\mathcal{T}_0$ and $\mathcal{L}_0$ satisfy* (3.13) *and $\langle \mathcal{T}_0, \mathcal{L}_0, \mathcal{L}' \rangle$ satisfy* (3.19)*. Let $\mathcal{T}_0 \subset \mathcal{T}_1 \subset \mathcal{T}_2 \subset \ldots$ be a sequence of extensions of $\mathcal{T}_0$, where each $\mathcal{T}_{i+1}$ is obtained from $\mathcal{T}_i$ by adding the defining axiom for a provably total function F (or f) that satisfies* (3.14) *(resp.* (3.15)*) (with $\mathcal{T}_i$ in place of $\mathcal{T}$). Let*

$$\mathcal{T}_\infty = \bigcup_{i \geq 0} \mathcal{T}_i, \qquad \mathcal{L}_\infty = \bigcup_{i \geq 0} \mathcal{L}_i$$

Then (i) $\mathcal{T}_\infty$ is a conservative extension of $\mathcal{T}_0$, (ii) the additional functions in $\mathcal{T}_\infty$ are $\mathbf{\Sigma}_1^1(\mathcal{L}')$-definable in $\mathcal{T}_0$.

Proof. For (i), by the hypothesis that $\mathcal{T}_0$ proves $\mathbf{\Sigma}_0^B(\mathcal{L}_0)$-**COMP**, it is easy to prove by induction on i, using Lemma 3.14 and Theorem 3.16, that $\mathcal{T}_i \vdash \mathbf{\Sigma}_0^B(\mathcal{L}_i)$-**COMP**, and that the new function F_{i+1}/f_{i+1} in $\mathcal{L}_{i+1}$, as well as $F_{i+1}^\star/f_{i+1}^\star$ are provably total in $\mathcal{T}_i$. As a result, $\mathcal{T}_{i+1}$ is a conservative extension of $\mathcal{T}_i$ (by Theorem 2.19 **a**). Hence $\mathcal{T}_\infty$ is a conservative extension of $\mathcal{T}_0$ by Theorem 2.19 **b**.

For (ii), using Theorem 3.16 we can prove by induction that each $\mathbf{\Sigma}_0^B(\mathcal{L}_i)$ formula is provably equivalent in $\mathcal{T}_i$ to a $\mathbf{\Sigma}_1^1(\mathcal{L}')$ formula. Hence the $\mathbf{\Sigma}_1^1(\mathcal{L}_i)$ defining axiom for F_{i+1}/f_{i+1} in $\mathcal{T}_{i+1}$ is equivalent (in $\mathcal{T}_\infty$) to a $\mathbf{\Sigma}_1^1(\mathcal{L}')$ formula which can be taken as the defining axiom for F_{i+1}/f_{i+1} in $\mathcal{T}_0$ (because $\mathcal{T}_\infty$ is conservative over $\mathcal{T}_0$). □

Proof of Theorem 3.11. First we apply Corollary 3.17 for $\mathcal{L}' = \mathcal{L}_A^2$, $\mathcal{L}_0 = \mathcal{L}_{\mathbf{FAC}^0}$, $\mathcal{T}_0 = \mathbf{VC}(\mathcal{L}_{\mathbf{FAC}^0})$, $\mathcal{T}_1 = \mathbf{VC}(F, \mathcal{L}_{\mathbf{FAC}^0})$ and $\langle \mathcal{T}_i \rangle_{i \geq 2}$ is a sequence of extensions of $\mathcal{T}_0$ such that (i) $\overline{\mathbf{VC}} = \bigcup_{i \geq 0} \mathcal{T}_i$ and (ii) each $\mathcal{T}_{i+1}$ contains only one extra function F or f not already in $\mathcal{T}_i$. (The condition (ii) is not important, but it is stated so that $\mathcal{T}_i$ satisfies the hypothesis of Corollary 3.17.)

Condition (3.19) holds for $\langle \mathcal{T}_0, \mathcal{L}_{\mathbf{FAC}^0}, \mathcal{L}_A^2 \rangle$ because every $\mathbf{\Sigma}_0^B(\mathcal{L}_A^2)$ formula is equivalent (in $\overline{\mathbf{V}}^0$) to a $\mathbf{\Sigma}_0^B$ formula (Lemma 2.23).

It is easy to see that (3.14) holds for $\mathcal{T}_0$ and F. By Lemma 3.14, (3.14) (or (3.15)) also holds for each new function $F_{\varphi,t}$ (or $f_{\varphi,t}$) in $\mathcal{T}_i$ for $i \geq 1$. In other words, the hypothesis of Corollary 3.17 is satisfied.

The conclusions of Theorem 3.11 now follow from Corollary 3.17 for the sequence $\mathbf{VC}(\mathcal{L}_{\mathbf{FAC}^0}), \mathcal{T}_1, \ldots$ and the fact that $\mathbf{VC}(\mathcal{L}_{\mathbf{FAC}^0})$ is a conservative extension of $\mathbf{VC}$. $\square$

3.2.4 Proof of the Definability Theorem for $\mathbf{VTC}^0$

Notice that (3.6) is really a defining axiom for $F^\star$, while *NUMONES* (3.4) is just a defining axiom for *numones*. So in order to apply the results of Section 3.2 for the theory $\mathbf{VTC}^0$, essentially we need to show that $numones^\star$ is provably total in $\mathbf{VTC}^0$; i.e., we need the following lemma:

Lemma 3.18. $\mathbf{VTC}^0(Row) \vdash \exists Y \forall u < b\ \delta_{NUM}(t(u), X^{[u]}, Y^{[u]})$.

Proof. The idea is to construct Y using $\mathbf{\Sigma}_0^B(Row)$-$\mathbf{COMP}$ from the counting array Y' for a "big" string X', where X' is the concatenation of the initial segments of the rows $X^{[0]}, \ldots, X^{[b-1]}$ of X. Formally, let s be an $\mathcal{L}_A^2$ number term that dominates $t(u)$, for all $u < b$. Let X' be defined by

$$X'(us + z) \leftrightarrow z < t(u) \wedge X^{[u]}(z), \quad \text{for } z < s, u < b.$$

In other words, for $u < b$, the bit string $X'(us) \ldots X'(us + t(u) - 1)$ is a copy of $X^{[u]}(0) \ldots X^{[u]}(t(u) - 1)$, and $X'(us + t(u)), \ldots, X'((u + 1)s - 1)$ are all 0. Therefore, for $z \leq t(u)$,

$$numones(z, X^{[u]}) = numones(us + z, X') - numones(us, X')$$

i.e., we will define Y so that

$$(Y^{[u]})^z + numones(us, X') = numones(us + z, X')$$

Let Y' be the counting array for X': $(Y')^z = numones(z, X')$. Hence, $(Y^{[u]})^z = y \leftrightarrow y + (Y')^{us} = (Y')^{us+z}$. Consequently, Y exists in $\mathbf{V}^0$ by $\mathbf{\Sigma}_0^B$ Multiple Comprehension. $\square$

3.3 $\mathbf{V}^0(m)$ and VACC

We use the following fact:

Proposition 3.19. *For $m \geq 2$, $\mathbf{AC}^0(m)$ is the $\mathbf{AC}^0$ closure of mod_m, where*

$$mod_m(x, X) = numones(x, X) \mod m$$

The theory $\mathbf{V}^0(m)$ is defined using the formula $\delta_{\mathbf{MOD}_m}(x, X, Y)$, which states that Y is a "counting modulo m" array for X:

$$\delta_{\mathbf{MOD}_m}(x, X, Y) \equiv Y(0,0) \ \wedge \ \forall z < x,$$
$$(X(z) \supset (Y)^{z+1} = ((Y)^z + 1) \mod m) \wedge (\neg X(z) \supset (Y)^{z+1} = (Y)^z).$$

Note that here we write $\varphi(y \mod m)$ for the formula

$$\exists r < m, \exists q \leq y, \ y = qm + r \wedge \varphi(r).$$

Definition 3.20. *For each $m \geq 2$, let $\mathbf{MOD}_m \equiv \forall X \forall x \exists Y \delta_{\mathbf{MOD}_m}(x, X, Y)$. The theory $\mathbf{V}^0(m)$ has vocabulary $\mathcal{L}_A^2$ and is axiomatized by $\mathbf{V}^0$ and the axiom $\mathbf{MOD}_m$. Also, $\mathbf{VACC} = \bigcup \{\mathbf{V}^0(m) \mid m \geq 2\}$.*

The theory $\mathbf{V}^0(2)$ can be equivalently defined using the axiom $\forall X \exists Y \delta_{parity}(X, Y)$ instead of $\mathbf{MOD}_2$, where $\delta_{parity}(X, Y)$ asserts that for $0 \leq i < |X|$, bit $Y(i+1)$ is 1 iff the number of 1's among bits $X(0), ..., X(i)$ is odd:

$$\delta_{parity}(X, Y) \equiv \neg Y(0) \wedge \forall i < |X| (Y(i+1) \leftrightarrow (X(i) \oplus Y(i))) \tag{3.20}$$

where $\oplus$ is exclusive OR. The function $mod_2(x, X)$ is also called $parity(x, X)$ and has the defining axiom

$$parity(x, X) = y \leftrightarrow \exists Y \leq |X|, \ \delta_{parity}(X, Y) \wedge (Y(x) \supset y = 1) \wedge (\neg Y(x) \supset y = 0) \tag{3.21}$$

Similar to Lemma 3.18, it can be shown that $mod_m^\star$ is provably total in $\mathbf{V}^0(m)$.

3.4 $\mathbf{VNC}^1$

The theory $\mathbf{VNC}^1$ [CM05, NC05] originated from Arai's single sorted theory **AID** [Ara00]. The idea comes from the fact that the problem of evaluating a balanced Boolean formula given the values of its propositional variables is complete for $\mathbf{NC}^1$ (the problem is still complete for $\mathbf{NC}^1$ when the formula is not required to be balanced, see [Bus87b]).

Consider the following encoding of a monotone Boolean formula using the heap data structure. We view the formula as a balanced binary tree with $(2a-1)$ nodes: a leaves numbered $a, (a+1), \ldots, (2a-1)$; and $(a-1)$ inner nodes numbered $1, 2, \ldots, (a-1)$. The two children of an inner node x are $2x$ and $(2x+1)$ (as in the heap data structure). Each inner node x is labeled with either $\wedge$ or $\vee$. Therefore the circuit can be encoded by (a, G), where $G(x)$ specifies the label of node x: $G(x)$ holds iff node x is an $\wedge$-gate.

In the formula $\delta_{MFV}(a, G, I, Y)$ given below (MFV stands for Monotone Formula Value), Y encodes an evaluation of the circuit (a, G) given input I, i.e., $Y(x)$ is the value of gate x (see Figure 3.1):

$$\begin{aligned}\delta_{MFV}(a, G, I, Y) \equiv\ & \forall x < a,\ (Y(x+a) \leftrightarrow I(x))\ \wedge\ [0 < x\ \supset\ Y(x) \leftrightarrow \\ & [(G(x) \wedge Y(2x) \wedge Y(2x+1)) \vee (\neg G(x) \wedge (Y(2x) \vee Y(2x+1)))]] \qquad (3.22)\end{aligned}$$

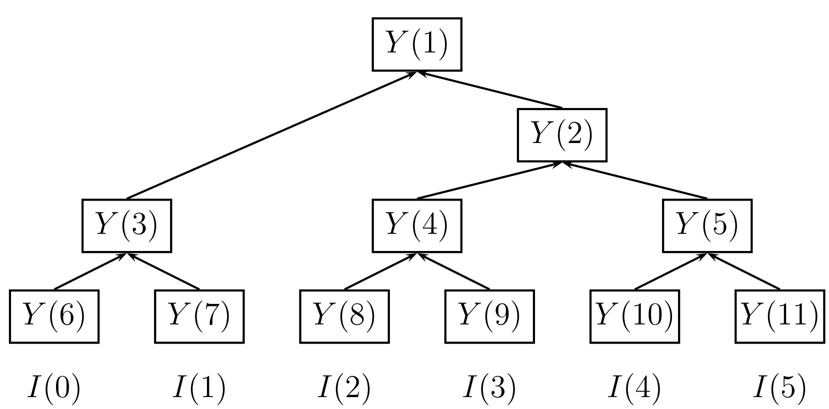

Figure 3.1: Computing $Fval(6, G, I)$ (G is not shown).

Definition 3.21. $\mathbf{VNC}^1$ *is the theory over* $\mathcal{L}_A^2$ *axiomatized by* $\mathbf{V}^0$ *and* MFV*, where* $MFV \equiv \forall a \forall G \forall I \exists Y\, \delta_{MFV}(a, G, I, Y)$.

Proposition 3.22. *Fval is* $\mathbf{AC}^0$*-many-one complete for* $\mathbf{NC}^1$*, where*

$$Fval(a, G, I) = Y \leftrightarrow |Y| \leq 2a \wedge \delta_{MFV}(a, G, I, Y) \tag{3.23}$$

Proof. The proposition follows from the fact [Bus87b] that the Boolean Sentence Value Problem is in **ALogTime** (which is the same as **FO**-uniform $\mathbf{NC}^1$) and the fact [BIS90, Lemma 6.2] that every language in **ALogTime** is $\mathbf{AC}^0$-many-one reducible to *Fval*. □

By Theorem 3.8, to show that the provably functions of $\mathbf{VNC}^1$ are precisely functions in $\mathbf{FNC}^1$, it suffices to show that the axiom $\forall b \forall X \forall G \forall I \exists Y \forall u < b \delta_{MFV}((X)^u, G^{[u]}, I^{[u]}, Y^{[u]})$ is provable in $\mathbf{VNC}^1$. This follows from Theorem 3.25 below.

The original definition of $\mathbf{VNC}^1$ [CM05] uses $\mathbf{\Sigma}_0^B$-*TreeRec*, the set of axioms of the form

$$\exists Y \forall x < a,\ [(Y(x + a) \leftrightarrow \psi(x)) \wedge (0 < x \supset (Y(x) \leftrightarrow \varphi(x)[Y(2x), Y(2x + 1)]))] \tag{3.24}$$

where $\psi(x)$ is a $\mathbf{\Sigma}_0^B$ formula, $\varphi(x)[p, q]$ is a $\mathbf{\Sigma}_0^B$ formula which contains two Boolean variables p and q, and Y does not occur in ψ and φ. We will show that our definition of $\mathbf{VNC}^1$ is equivalent to the definition from [CM05]. Since MFV is an instance of the $\mathbf{\Sigma}_0^B$-*TreeRec* axiom scheme, we need only to show that $\mathbf{\Sigma}_0^B$-*TreeRec* is provable in $\mathbf{VNC}^1$ (Theorem 3.23); Theorems 3.24 and 3.25 below will show that indeed $\mathbf{VNC}^1$ proves several generalizations of $\mathbf{\Sigma}_0^B$-*TreeRec*.

Theorem 3.23. *The* $\mathbf{\Sigma}_0^B$*-TreeRec axiom scheme is provable in* $\mathbf{VNC}^1$.

Proof. Given a, ψ and φ, the idea is to construct a (large) treelike circuit (b, G) and inputs I so that from $Fval(b, G, I)$ we can extract Y (using $\mathbf{\Sigma}_0^B$-**COMP**) that satisfies (3.24).

Notice the "gates" $\varphi(x)[p,q]$ in (3.24) can be any of the sixteen Boolean functions in two variables p,q. We will (uniformly) construct binary treelike $\wedge$-$\vee$ circuits of constant depth that compute $\varphi(x)[p,q]$.

Let

$$\beta_1,\ldots,\beta_8,\beta_9 \equiv \neg\beta_1,\ldots,\beta_{16} \equiv \neg\beta_8$$

be the sixteen Boolean functions in two variables p,q. Each β_i can be computed by a binary treelike and-or circuit of depth 2 with inputs among 0, 1, p, q, $\neg p$, $\neg q$. For $1 \le i \le 16$, let X_i be defined by

$$X_i(x) \leftrightarrow (x < a \wedge \varphi(x)[p,q] \leftrightarrow \beta_i(p,q))$$

Then,

$$\varphi(x)[p,q] \leftrightarrow \bigvee_{i=1}^{16}(X_i(x) \wedge \beta_i(p,q))$$

Consequently, $\varphi(x)[p,q]$ can be computed by a binary and-or tree T_x of depth 7 whose inputs are $0,1,p,\neg p,q,\neg q,X_i(x)$. Similarly, $\neg\varphi(x)[p,q]$ is computed by a binary and-or tree T'_x having the same depth and set of inputs. Our large tree G has one copy of T_1, and in general for each copy of T_x or T'_x, there are multiple copies of $T_{2x},T_{2x+1},T'_{2x},T'_{2x+1}$ that supply the inputs $Y(2x),Y(2x+1),\neg Y(2x),\neg Y(2x+1)$, and other trivial treelike circuits that provide inputs 0, 1, $X_i(x)$ $(1 \le i \le 16)$.

Finally, I is defined as follows: $I(x) \leftrightarrow (x < a \wedge \psi(x))$. □

3.4.1 $\mathbf{VTC}^0 \subseteq \mathbf{VNC}^1$

To show that $\mathbf{VNC}^1$ extends $\mathbf{VTC}^0$ it suffices to show that the axiom *NUMONES* is provable in $\mathbf{VNC}^1$. In other words, we need to formalize in $\mathbf{VNC}^1$ the construction of $\mathbf{NC}^1$ circuits that compute *numones* and prove (in $\mathbf{VNC}^1$) the correctness of this construction. We formalize the construction by Buss [Bus87b].

The next two theorems show that $\mathbf{VNC}^1$ proves some generalizations of $\mathbf{\Sigma}_0^B$-*TreeRec*. They are useful in formalizing the construction of the counting circuits.

Theorem 3.24. *Suppose that* $2 \le k \in \mathbb{N}$*, and* $\psi(x)$ *and* $\varphi(x)[p_0, \ldots, p_{k-1}]$ *are* $\mathbf{\Sigma}_0^B$ *formulas. Then* $\mathbf{VNC}^1$ *proves*

$$\exists Y,\ \forall x < ka,\ a \le x \supset Y(x) \leftrightarrow \psi(x) \wedge \\ \forall x < a,\ Y(x) \leftrightarrow \varphi(x)[Y(kx), \ldots, Y(kx+k-1)] \quad (3.25)$$

Proof. We prove for the case $k = 4$; similar arguments work for other cases.

Using Theorem 3.23 we will define a', ψ', φ' so that from Y' that satisfies the $\mathbf{\Sigma}_0^B$-*TreeRec* axiom (3.24) for a', ψ' and φ' we can obtain Y that satisfies (3.25) above.

Intuitively, consider Y in (3.25) as a forest of three trees whose nodes are labeled with $Y(x)$, $x < |Y|$. Then Y has branching factor of 4 (since $k = 4$), and the three trees are rooted at $Y(1)$, $Y(2)$ and $Y(3)$. So it suffices to simulate each layer in Y by two layers in the binary tree Y'. (See Figure 3.2.)

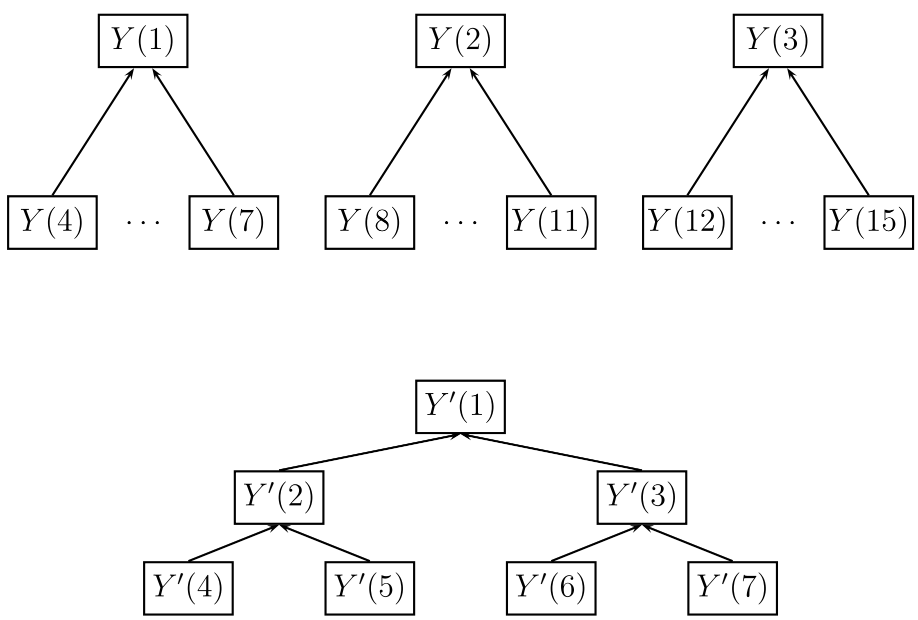

Figure 3.2: The forest Y in Theorem 3.24 when $k = 4$. Trees rooted at $Y(1)$, $Y(2)$ and $Y(3)$ are simulated by the sub-trees $Y'(4)$, $Y'(5)$ and $Y'(6)$, respectively.

We will define an injective map f so that $Y(x) \leftrightarrow Y'(f(x))$. Since the trees rooted at $Y(1)$, $Y(2)$ and $Y(3)$ are disjoint, f is defined so that these trees are the images of disjoint subtrees in the tree Y'. Here we take

$$f(1) = 4, f(2) = 5, f(3) = 6$$
$$f(4^m + y) = 4^{m+1} + y \qquad \text{for } \ 0 \le y < 3 \cdot 4^m$$

(Note that f has a $\mathbf{\Delta}_0$ graph (Example 2.6), so it is provably total in $\mathbf{I\Delta}_0$ and hence also in $\mathbf{V}^0$.)

Now we need ψ' such that

$$\psi'(f(x)) \leftrightarrow \psi(x) \qquad \text{for } a \le x < 4a$$

Define ψ' by

$$\psi'(4^{m+1} + y) \leftrightarrow \psi(4^m + y) \qquad \text{for } y < 3 \cdot 4^m \text{ and } a \le 4^m + y < 4a$$

To obtain φ', write $\varphi(x)[p_0, p_1, p_2, p_3]$ in the form

$$\varphi_1(x)[\varphi_2(x)[p_0, p_1], \varphi_3(x)[p_2, p_3]]$$

where φ_i is $\mathbf{\Sigma}_0^B$ with at most 2 Boolean variables, for $1 \le i \le 3$. Define φ' so that

$$\varphi'(4^{m+1} + y)[p, q] \leftrightarrow \varphi_1(4^m + y)[p, q] \qquad \text{for } y < 3 \cdot 4^m$$
$$\varphi'(2 \cdot 4^{m+1} + 2y)[p, q] \leftrightarrow \varphi_2(4^m + y)[p, q] \qquad \text{for } y < 3 \cdot 4^m/2$$
$$\varphi'(2 \cdot 4^{m+1} + 2y + 1)[p, q] \leftrightarrow \varphi_3(4^m + y)[p, q] \qquad \text{for } y < 3 \cdot 4^m/2$$

Finally, let $a' = f(a)$. Let Y' satisfies (3.24) for a', ψ' and φ', and let Y be such that

$$Y(x) \leftrightarrow Y'(f(x))$$

It is easy to verify that Y satisfies (3.25). □

The next theorem shows that in $\mathbf{VNC}^1$ we can evaluate multiple inter-connected Boolean circuits with logarithmic depth and constant fan-in.

Theorem 3.25. *Suppose that* $1 \le m, \ell \in \mathbb{N}$, $\psi_i(x,y)$ *and* $\varphi_i(x,y)[p_1, q_1, \ldots, p_{m\ell}, q_{m\ell}]$ *are* $\mathbf{\Sigma}_0^B$ *formulas for* $1 \le i \le m$, *where* $\vec{p}, \vec{q}$ *are the Boolean variables. Then* $\mathbf{VNC}^1$ *proves the existence of* $Z_1, \ldots, Z_m$ *such that*

$$\forall z < c \forall x < a \bigwedge_{i=1}^{m} [(Z_i^{[z]}(x+a) \leftrightarrow \psi_i(z,x)) \wedge 0 < x \supset$$
$$(Z_i^{[z]}(x) \leftrightarrow \varphi_i(z,x)[Z_1^{[z]}(2x), Z_1^{[z]}(2x+1), \ldots, Z_m^{[z+\ell-1]}(2x), Z_m^{[z+\ell-1]}(2x+1)]))]$$

Proof. The idea is to construct a constant k, a number a' and $\mathbf{\Sigma}_0^B$ formulas $\psi'(c,x)$ and $\varphi'(c,x)[p_0, \ldots, p_{k-1}]$ so that from Y that satisfies (3.25) (for k, a', ψ' and φ') we can obtain $Z_1, \ldots, Z_m$.

Consider, for example, $m = 2, \ell = 2$. W.l.o.g., assume that $c \ge 1$. The following (overlapping) subtrees

$$Z_1^{[0]}, Z_2^{[0]}, \ldots, Z_1^{[c-1]}, Z_2^{[c-1]} \tag{3.26}$$

have branching factor 8 (i.e., $2m\ell$). So let $k = 8$ (i.e., $k = 2m\ell$). We will construct Y (with branching factor 8) so that the disjoint subtrees rooted at

$$Y(c), \ldots, Y(3c-1) \tag{3.27}$$

are exactly the subtrees listed in (3.26).

We will define an 1-1, into map $s : \{1,2\} \times \mathbb{N}^2 \to \mathbb{N}$ so that

$$Z_i^{[z]}(x) \leftrightarrow Y(s(i,z,x))$$

For the root level of the trees in (3.26) we need

$$s(1,0,1) = c,\ s(2,0,1) = c+1,\ s(1,1,1) = c+2,\ s(2,1,1) = c+3,\ \ldots$$

For other levels we need: If $s(i,z,x) = y$, then

$$s(1,z,2x) = 8y,\ s(1,z,2x+1) = 8y+1,\ \ldots,\ s(2,z+1,2x+1) = 8y+7$$

To define s, we define partial, onto maps $f, g : \mathbb{N} \to \mathbb{N}$ and $h : \mathbb{N} \to \{1,2\}$ so that

$$s(h(y), g(y), f(y)) = y$$

In other words,

$$Y(y) \leftrightarrow Z_{h(y)}^{[g(y)]}(f(y))$$

For example, for $0 \leq z < 2c$:

$$f(c+z) = 1, \qquad g(c+z) = \lfloor z/2 \rfloor, \qquad h(c+z) = 1 + (z \mod 2)$$

In general, we need to define f, g, h only for values of x of the form $8^r c + z$ for $0 \leq z < 2 \cdot 8^r c$. The definitions of f, g, h at $8^r c + z$ are straightforward using the base 8 notation for z, where $0 \leq z < 2 \cdot 8^r c$.

Once f, g, h are defined, the formula ψ' and φ' are defined by

$$\psi'(c, x) \leftrightarrow \psi_{h(x)}(g(x), f(x)) \qquad \text{and} \qquad \varphi'(c, x)[\ldots] \leftrightarrow \varphi_{h(x)}(g(x), f(x))[\ldots]$$

(where $\ldots$ is the list of $2m\ell$ Boolean variables). □

Theorem 3.26. $\mathbf{VTC}^0 \subseteq \mathbf{VNC}^1$.

Proof. $numones(n, X)$ can be computed using the divide-and-conquer technique: let c_i $(1 \leq i < 2n)$ be such that

$$c_{i+n} = X(i) \qquad \text{for } 0 \leq i < n$$

$$c_i = c_{2i} + c_{2i+1} \qquad \text{for } 1 \leq i < n$$

Then $numones(n, X) = c_1$. (See Figure 3.3 for an example.) The next theorem shows

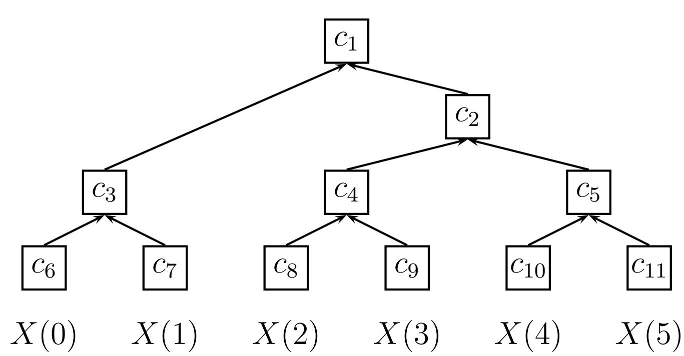

Figure 3.3: Computing $numones(6, X)$ in $\mathbf{NC}^1$

that we can formalize the same computation, but the "counters" are strings $Z^{[i]}$ instead of numbers c_i. This is more general since converting a number into its binary representation can be done in $\mathbf{V}^0$.

Finally, the fact that $\mathbf{VNC}^1$ proves the correctness of the construction is shown in Theorem 3.28. □

Theorem 3.27. $\mathbf{VNC}^1$ *proves*

$$\exists Z \forall x < a,\ Z^{[a+x]} = I^{[x]} \wedge x > 0 \supset Z^{[x]} = Z^{[2x]} + Z^{[2x+1]}$$

Proof. We compute $Z^{[i]}$ as in Figure 3.3 where now the nodes contain $Z^{[x]}$ instead of c_x. Note that if for each $x < a$ we simply construct an $\mathbf{AC}^0$ circuit that performs string addition to compute $Z^{[x]}$ $(= Z^{[2x]} + Z^{[2x+1]})$, then we will end up with an $\mathbf{AC}^1$ circuit.

Here we use the fact that

$$X + Y + T = G(X, Y, T) + H(X, Y, T) \tag{3.28}$$

where $G(X, Y, T)$ is the string of bit-wise sums, and $G(X, Y, T)$ is the string of carries:

$$\begin{aligned} G(X, Y, T)(z) &\leftrightarrow X(z) \oplus Y(z) \oplus T(z) \\ H(X, Y, T)(0) &\leftrightarrow \perp \\ H(X, Y, T)(z+1) &\leftrightarrow ((X(z) \wedge Y(z)) \vee (X(z) \wedge T(z)) \vee (Y(z) \wedge T(z))) \end{aligned}$$

It is straightforward to show that $\mathbf{V}^0(G, H)$ proves the equation (3.28).

Thus, for each $Z^{[x]}$ we have a pair of strings $(S^{[x]}, C^{[x]})$ where $S^{[x]}$ is the string of bit-wise sums and $C^{[x]}$ is the string of carries for computing $Z^{[x]}$. For $1 \le x < 2a$, $Z^{[x]} = S^{[x]} + C^{[x]}$. For $a \le x < 2a$, $S^{[x]} = I^{[x]}, C^{[x]} = \varnothing$ ($\varnothing$ denotes the empty set), and for $1 \le x < a$ we will have:

$$S^{[x]} + C^{[x]} = S^{[2x]} + C^{[2x]} + S^{[2x+1]} + C^{[2x+1]}$$

$S^{[x]}$ and $C^{[x]}$ are computed as follows:

$$S^{[x]} = G(C^{[2x+1]}, U, V), \qquad C^{[x]} = H(C^{[2x+1]}, U, V)$$

where

$$U = G(S^{[2x]}, C^{[2x]}, S^{[2x+1]}), \qquad V = H(S^{[2x]}, C^{[2x]}, S^{[2x+1]})$$

In other words, let F_1, F_2 be the $\mathbf{AC}^0$ functions

$$F_1(X, Y, Z, W) = G(W, G(X, Y, Z), H(X, Y, Z))$$

$$F_2(X, Y, Z, W) = H(W, G(X, Y, Z), H(X, Y, Z))$$

Then

$$S^{[x]} = F_1(S^{[2x]}, C^{[2x]}, S^{[2x+1]}, C^{[2x+1]}), \qquad C^{[x]} = F_2(S^{[2x]}, C^{[2x]}, S^{[2x+1]}, C^{[2x+1]})$$

We need to prove in $\mathbf{VNC}^1$ the existence of S and C such that

$$\forall x < a, S^{[x+a]} = I^{[x]} \wedge C^{[x+a]} = \varnothing \wedge 0 < x \supset$$
$$S^{[x]} = F_1(S^{[2x]}, C^{[2x]}, S^{[2x+1]}, C^{[2x+1]}) \wedge C^{[x]} = F_2(S^{[2x]}, C^{[2x]}, S^{[2x+1]}, C^{[2x+1]})$$

Notice that for each z, the bits $S^{[x]}(z), C^{[x]}(z)$ are computed from the bits

$$\{S^{[2x]}(y), S^{[2x+1]}(y), C^{[2x]}(y), C^{[2x+1]}(y) : z - 2 \leq y \leq z\}$$

(where we define $S^{[2x]}(y) \equiv \perp$ if $y < 0$, etc.). This is not in the form of the hypothesis of Theorem 3.25, but we can put it in the right form by "transposing" S and C. Formally, let S' and C' be such that

$$S'^{[y]}(x) \leftrightarrow S^{[x]}(y), \qquad C'^{[y]}(x) \leftrightarrow C^{[x]}(y)$$

Then $S'^{[z]}(x)$ and $C'^{[z]}(x)$ are computed from

$$\{S'^{[y]}(2x), S'^{[y]}(2x+1), C'^{[y]}(2x), C'^{[y]}(2x+1) : z - 2 \leq y \leq z\}$$

by some $\mathbf{\Sigma}_0^B$ formulas. By Theorem 3.25, $\mathbf{VNC}^1$ proves the existence of S' and C', and hence the existence of S and C. $\square$

Given a, I, let Z be constructed as above. Define $Sum(a, I)$ by

$$Sum(0, I) = \varnothing, \quad Sum(a, I) = Z^{[1]} \text{ for } a \geq 1$$

Then Sum is provably total in $\mathbf{VNC}^1$, so $\mathbf{VNC}^1(Sum)$ is a conservative extension of $\mathbf{VNC}^1$. The fact that $\mathbf{VNC}^1$ proves the correctness of the construction given above is shown in the next theorem.

Theorem 3.28. $\mathbf{VNC}^1(Sum) \vdash Sum(z, I) + I^{[z]} = Sum(z + 1, I)$.

Proof. Let $z_0 = 1, z_1, z_2, \ldots, z_d = z$ be the initial segment of the binary representation of z. Let Z be the string constructed for computing $Sum(z + 1, I)$ as in the proof of Theorem 3.27. Then the path from root $Z^{[1]}$ to leaf $Z^{[z]}$ (the rightmost path, see Figure 3.3) consists of the nodes

$$Z^{[z_0]}, Z^{[z_1]}, \ldots, Z^{[z_d]}$$

Let Z_0 be the string constructed for computing $Sum(z, I)$. It can be proved by (reverse) induction on i that

$$(Z^{[z_i]} = Z_0^{[z_i]} + I^{[z]}) \ \wedge \ \forall x < z(|x| = |z_i| \wedge x < z_i \supset Z^{[x]} = Z_0^{[x]}) \qquad \square$$

3.5 VNL

Suppose that (a, E) encode a directed graph G: the vertices of G are numbered $0, \ldots, (a-1)$, and for $x, y < a$, $E(x, y)$ holds if and only if there is a directed edge from x to y. Then the string function $Conn(a, E)$ is $\mathbf{AC}^0$ complete for $\mathbf{FNL}$, where $Conn(a, E)(z, x)$ holds iff there is a path in G from 0 to x of length at most z:

Proposition 3.29. *$Conn$ is $\mathbf{AC}^0$ many-one complete for $\mathbf{NL}$, where $Conn(a, E) = Y \leftrightarrow |Y| \leq \langle a, a\rangle \wedge \delta_{CONN}(a, E, Y)$, and*

$$\delta_{CONN}(a, E, Y) \equiv Y(0, 0) \wedge \forall x < a(x \neq 0 \supset \neg Y(0, x)) \wedge$$
$$\forall z < a \forall x < a, \ Y(z + 1, x) \leftrightarrow (Y(z, x) \vee \exists y < a, \ Y(z, y) \wedge E(y, x)). \quad (3.29)$$

Definition 3.30. **VNL** *is the theory over* $\mathcal{L}_A^2$ *that is axiomatized by* $\mathbf{V}^0$ *and the axiom* $CONN \equiv \forall a \forall E \exists Y \delta_{CONN}(a, E, Y)$,

To apply Theorem 3.8 for **VNL** we need the next lemma, which essentially shows that $Conn^\star$ is provably total in $\mathbf{VNL}(Row, seq)$. (Recall that $seq(u, X) = (X)^u$ is the u-th number of the sequence encoded by X.)

Lemma 3.31. $\mathbf{VNL}(Row, seq) \vdash \forall b \forall Z \forall E \exists Y \forall u < b\, \delta_{CONN}((Z)^u, E^{[u]}, Y^{[u]})$.

Proof. Given the graphs G_u encoded by $((Z)^u, E^{[u]})$ (for $0 \leq u < b$), the idea is to construct a (larger) graph G encoded by (a, E') so that from $Conn(a, E')$ we can define (using $\mathbf{\Sigma}_0^B$-**COMP**) $Conn((Z)^0, E^{[0]}), \ldots, Conn((Z)^{b-1}, E^{[b-1]})$. The graph G is obtained by introducing a common source node s with directed edges to the sources of G_u.

Formally, for each u, there is a copy of G_u in G with vertices

$$s_u = \langle u+1, 0\rangle, \langle u+1, 1\rangle, \ldots, \langle u+1, (Z)^u - 1\rangle$$

Let $a = \langle b, max\{(Z)^u\}\rangle$. E' has, in addition, edges $\langle 0, s_u\rangle$, for $0 \leq u < b$.

Now, there is a path of length z from 0 to x in G_u iff there is a path of length $z+1$ from 0 to $\langle u+1, x\rangle$ in G, i.e.,

$$Conn((Z)^u, E^{[u]})(z, x) \leftrightarrow Conn(a, E')(z+1, \langle u+1, x\rangle)$$

for $0 \leq x, z < (Z)^u$. □

3.6 VL

Let $SinglePath(a, E)$ be the function that when (a, E) encode a directed graph whose out-degree is exactly one gives the unique path from the source node 0 of length a. Then $SinglePath$ is complete for **L**. In the following formula, $(P)^z = x$ iff x has distance z to

0. The function $(P)^z$ can be eliminated using its defining axiom (Definition 2.20). Let

$$\delta_{SinglePath}(a, E, P) \equiv P \leftrightarrow [\exists x < a \neg \exists! y < a E(x, y) \supset P = \varnothing]$$
$$\wedge [\forall x < a \exists! y < a E(x, y) \supset (P)^0 = 0 \wedge \forall z < a E((P)^z, (P)^{z+1})]$$

Proposition 3.32. *SinglePath is* $\mathbf{AC}^0$ *many-one complete for* $\mathbf{L}$*, where* $SinglePath(a, E) = P \leftrightarrow |P| \leq \langle a, a\rangle \wedge \delta_{SinglePath}(a, E, P)$.

Definition 3.33 (**VL**)**.** *Let* $SinglePATH \equiv \forall a \forall E \exists P \leq \langle a, a\rangle, \delta_{SinglePath}(a, E, P)$. **VL** *is the theory over* $\mathcal{L}_A^2$ *that is axiomatized by* $\mathbf{V}^0$ *and* $SinglePATH$.

To apply Theorem 3.8 for **VL** and **L** we need

Lemma 3.34. $\mathbf{VL} \vdash \forall b \forall X \forall E \exists P \forall u < b, \delta_{SinglePath}((X)^u, E^{[u]}, P^{[u]})$.

Proof. Informally, given b graphs $G_u = (a, E^{[u]})$ (for $0 \leq u < b$) whose out-degree is exactly 1 we need to construct simultaneously in **VL** the paths $P^{[u]}$ that satisfy *SinglePATH* for $(a, E^{[u]})$, for $0 \leq u < b$.

We will construct a graph $G = (a', E')$ so that from $Q = SinglePath(a', E')$ we can define $P^{[0]}, \ldots, P^{[b-1]}$. In fact, Q will be just the concatenation of $P^{[u]}$, $0 \leq u < b$.

The nodes of G are triples $\langle u, z, x\rangle$ $(0 \leq u < b, 0 \leq z \leq a, 0 < x < a)$. Our aim is that if $P^{[u]}$ encodes the path $(0, x_1, \ldots, x_a)$, then Q has a sub-path:

$$\langle u, 0, 0\rangle, \langle u, 1, x_1\rangle, \ldots, \langle u, a, x_a\rangle$$

The set E' (of edges of G) consist of (for $0 \leq u < b$):

$$(\langle u, z, x\rangle, \langle u, z+1, y\rangle) \qquad \text{for } 0 \leq z, x, y < a \text{ and } (x, y) \in E^{[u]}$$
$$(\langle u, a, x\rangle, \langle u+1, 0, 0\rangle) \qquad \text{for } x < a$$

Let a' and $Q = SinglePath(a', E')$. We can prove by induction (on u and z) that the $(u(a+1)+z)$-th node in Q must be of the form $\langle u, z, x\rangle$:

$$(Q)^{u(a+1)+z} = \langle u, z, x\rangle \qquad \text{for some } x, 0 \leq x < a$$

Define P so that

$$(P^{[u]})^z = x \text{ iff } (Q)^{u(a+1)+z} = \langle u, z, x\rangle$$

It is straightforward that each $P^{[u]}$ satisfies *SinglePATH* for $(a, E^{[u]})$. $\square$

Zambella [Zam97] introduced the theory $\mathbf{\Sigma}_0^B$-**Rec** which is axiomatized essentially by $\mathbf{V}^0$ together with the following axiom scheme:

$$\forall w < b \forall x < a \exists y < a\, \varphi(w, x, y) \supset \exists Z, \forall w < b\, \varphi(w, (Z)^w, (Z)^{w+1}).$$

for all $\mathbf{\Sigma}_0^B$ formulas φ not involving Z. It is easy to show that **VL** is the same as $\mathbf{\Sigma}_0^B$-**Rec**. Now we prove:

Theorem 3.35. $\mathbf{VNC}^1 \subseteq \mathbf{VL}$.

Proof. It suffices to show that **VL** proves *MFV* (Definition 3.21), or equivalently, *Fval* (3.23) is provably total in **VL**.

Thus, given (a, G, I) (specifying a "balanced" formula and the truth values of its variables), for each inner node z of this balanced tree (where $1 \le z < a$) we construct a graph encoded by (a', E) so that the value of this node, $Fval(a, G, I)(z)$, can be obtained from $SinglePath(a', E)$. Then, since $SinglePath^\star$ is provably total in **VL**, all nodes in (a, G, I) can be evaluated at once.

The graph (a', E) describes a depth-first traversal in the circuit (a, G) starting from node z. Each vertex is a (potential) state of the traversal. There is a source (vertex 0), and each other vertex is of the form

$$\langle x, \mathsf{d}, 0\rangle \quad \text{or} \quad \langle x, \mathsf{u}, v\rangle, \qquad \text{where } z \le x < 2a, v \in \{0, 1\}$$

(here $\mathsf{d} = 1, \mathsf{u} = 2$ indicate the direction of the traversal). A vertex $\langle x, \mathsf{d}, 0\rangle$ corresponds to the state when the depth-first traversal visits the gate numbered x for the first time (so in general it will go "down"). Similarly, a state $\langle x, \mathsf{u}, v\rangle$ is when the search visits gate x the second time (thus the direction is "up"); by this time the truth value of the gate is known, and v carries this truth value.

The edges of this graph represent the transition between the states of the search. The search starts at the root, and when visiting a gate x for the first time, it will travel down along the left-most branch from x. Thus we have the following edge:

$$(0, \langle z, \mathsf{d}, 0\rangle), \qquad \text{and} \quad (\langle x, \mathsf{d}, 0\rangle, \langle 2x, \mathsf{d}, 0\rangle) \text{ for } z \leq x < a$$

And here are the transitions when the algorithm reaches the input gates:

$$(\langle x+a, \mathsf{d}, 0\rangle, \langle x+a, \mathsf{u}, 0\rangle) \qquad \text{if } \neg I(x),\ 0 \leq x < a$$

$$(\langle x+a, \mathsf{d}, 0\rangle, \langle x+a, \mathsf{u}, 1\rangle) \qquad \text{if } I(x),\ 0 \leq x < a$$

For an $\vee$–gate x (i.e., if $\neg G(x)$, where $1 \leq x < a$), if the left child ($2x$) is $\top$ then the search can ignore the right child ($2x+1$). We have the following edges:

$$\text{either child is } \top: \qquad (\langle 2x, \mathsf{u}, 1\rangle, \langle x, \mathsf{u}, 1\rangle) \text{ and } (\langle 2x+1, \mathsf{u}, 1\rangle, \langle x, \mathsf{u}, 1\rangle)$$

$$\text{the left child is } \bot: \qquad (\langle 2x, \mathsf{u}, 0\rangle, \langle 2x+1, \mathsf{d}, 0\rangle) \ \ \text{(go the the right child)}$$

$$\text{the right child is } \bot: \qquad (\langle 2x+1, \mathsf{u}, 0\rangle, \langle x, \mathsf{u}, 0\rangle) \ \ \text{(value of gate } x \text{ must be } \bot)$$

The transitions for an $\wedge$–edge are similar.

Let $a' = \langle 2a-1, 2, 1\rangle$. It is easy to see that (a', E) encodes a graph of out-degree ≤ 1. To make the out-degree *exactly* 1 we can create an extra vertex and connect all vertices with out-degree 0 to it. The value of node z in (a, G, I) is determined by whether $\langle z, \mathsf{u}, 1\rangle$ is reachable from 0:

$$Fval(a, G, I)(z) \leftrightarrow \exists w\, (SinglePath(a', E))^w = \langle z, \mathsf{u}, 1\rangle$$

To prove the correctness of the construction, let P_z be the path in the graph constructed for computing $Fval(a, G, I)(z)$ $(1 \leq z < a)$. Let Y be defined by

$$|Y| \leq 2a \wedge \forall z < a\ ((Y(a+z) \leftrightarrow I(z)) \wedge (z \neq 0 \supset Y(z) \leftrightarrow \exists w (P_z)^w = \langle z, \mathsf{u}, 1\rangle))$$

We show that Y satisfies $\delta_{MFV}(a, G, I, Y)$ (3.22). It suffices to show that

$$0 < z < a \supset \ (Y(z) \leftrightarrow [(G(z) \wedge Y(2z) \wedge Y(2z+1)) \vee (\neg G(z) \wedge (Y(2z) \vee Y(2z+1)))])$$

This can be proved by reverse induction on the length of z. $\square$

3.7 VP

We use the fact that evaluating a monotone Boolean circuit, where the gates are numbered $0, 1, 2, \ldots, (a-1)$ and inputs of a gate x come only from gates y, where $y < x$, is complete for **P**. Suppose that (a, G, E) code a monotone circuit, where

- $G(0)$ and $G(1)$ are constants 0 and 1 respectively,
- $G(x)$ specifies gate x for $2 \leq x < a$ ($G(x)$ holds iff gate x is an $\wedge$ gate), and
- for $0 \leq y < x, 2 \leq x < a$, $E(y, x)$ states that the output of gate y is connected to an input of gate x.

In the formula δ_{MCV} below (MCV stands for Monotone Circuit Value) Y evaluates the circuit: $Y(x)$ holds iff the output of gate x is 1.

$$\begin{aligned}\delta_{MCV}(a, G, E, Y) \;\equiv\; &\neg Y(0) \wedge Y(1) \wedge \;\forall x < a,\; 2 \leq x \;\supset \\ &Y(x) \leftrightarrow [(G(x) \wedge \forall y < x(E(y, x) \supset Y(y))) \vee (\neg G(x) \wedge \exists y < x(E(y, x) \wedge Y(y)))]\end{aligned}$$

Proposition 3.36. *Mcv is* $\mathbf{AC}^0$*-many-one complete for* **P***, where* $Mcv(a, G, E) = Y \;\leftrightarrow\; |Y| \leq a \wedge \delta_{MCV}(a, G, E, Y)$.

Definition 3.37. **VP** *is the theory over* $\mathcal{L}_A^2$ *and is axiomatized by the axioms of* $\mathbf{V}^0$ *and* $MCV \equiv \forall a \forall G \forall E \exists Y \delta_{MCV}(a, G, E, Y)$.

Notice that because *Mcv* is $\mathbf{AC}^0$-many-one complete for **P**, proving directly that **VP** can $\mathbf{\Sigma}_1^1$-define all functions in **FP** is easier than the proof of the general case for **VC** in Section 3.2: For each polytime function F we can describe a circuit (a, G, E) and from Y that satisfies $\delta_{MCV}(a, G, E, Y)$ we can extract the value of F.

Of course the results in Section 3.2 also imply that the provably total functions of **VP** are precisely **FP**. We will need to show that

$$\mathbf{VP} \vdash \forall b \forall X \forall G \forall E \exists Y \forall u < b,\; \delta_{MCV}((X)^u, G^{[u]}, E^{[u]}, Y^{[u]})$$

This is straightforward and we leave the details to the reader.

3.7.1 $\mathbf{VP} = \mathbf{TV}^0$

To define $\mathbf{TV}^0$ we need the empty set $\varnothing$ and the successor function for strings: $S(X)$ is the set whose binary representation when interpreted as a natural number is one plus that of X. (Both functions are $\mathbf{AC}^0$.) The theory $\mathbf{TV}^0$ [Coo05] has vocabulary $\mathcal{L}_A^2 \cup \{\varnothing, S\}$ and extends $\mathbf{V}^0$ by $\mathbf{\Sigma}_0^B$-**SIND**, the string induction axioms for $\mathbf{\Sigma}_0^B$ formulas. In general, Φ-**SIND** is the set of axioms of the form

$$[\varphi(\varnothing) \wedge \forall X(\varphi(X) \supset \varphi(S(X))] \supset \varphi(Y)$$

for $\varphi(X)$ in Φ that may have free variables other than X.

Write $X^{<z}$ for $Cut(z, X)$ (see (3.7) on page 29), and define

$$\varphi^{rec}(y, X) \equiv \forall i < y(X(i) \leftrightarrow \varphi(i, X^{<i}))$$

The bit recursion scheme Φ-**BIT-REC** is the set of axioms of the form $\exists X \varphi^{rec}(y, X)$ where $\varphi(i, X)$ is in Φ and φ may have free variables other than X. The next theorem is proved by Cook [Coo05].

Theorem 3.38. $\mathbf{TV}^0(Cut)$ *is equivalent to* $\mathbf{V}^0(\varnothing, S) + \mathbf{\Sigma}_0^B$-**BIT-REC**.

We will use the above theorem to prove the next theorem:

Theorem 3.39. $\mathbf{TV}^0$ *is a conservative extension of* $\mathbf{VP}$.

Proof. The axiom MCV is a special case of $\mathbf{\Sigma}_0^B$-**BIT-REC**, so by Theorem 3.38 MCV is provable in $\mathbf{TV}^0(Cut)$, and hence in $\mathbf{TV}^0$. Therefore $\mathbf{VP} \subseteq \mathbf{TV}^0$.

For the conservativity, it suffices to show that $\mathbf{VP}(Cut) \vdash \mathbf{\Sigma}_0^B$-**BIT-REC**. Thus for each $\mathbf{\Sigma}_0^B$-formula $\varphi(\vec{w}, y, X, \vec{W})$ we must show

$$\mathbf{VP}(Cut) \vdash \exists X \forall z < y,\ X(z) \leftrightarrow \varphi(\vec{w}, z, X^{<z}, \vec{W}) \tag{3.30}$$

We will show that $\mathbf{VP}$ proves the existence of a monotone circuit C that computes X. It is easier to describe C by its sub-circuits.

Our circuit C will compute the bits of X in the order of $X(0), X(1), \ldots$. Because C is monotone, we will compute explicitly both $X(z)$ and $\neg X(z)$ for $0 \leq z < y$ by a sub-circuit C_z. Since the outputs of $C_{z'}$ are fed to C_z for $z' < z$, we need to make sure that the gates in C_{z+1} have larger indices than those in C_z. Thus the gates in C_z have indices $\langle (z+1)n, i\rangle$ for $1 \leq i \leq m$ where m is specified below and n is sufficiently large so that $\langle (z+1)n, 1\rangle > \langle zn, m\rangle$ for $z \leq y$. The constants 0,1 and inputs $\vec{w}$ and $\vec{W}$ are given by the gates $0, 1, \langle 0, 1\rangle, \ldots, \langle 0, m\rangle$. Here $\vec{w}$ are presented in unary while $\vec{W}$ are given in binary; we also need $\neg W_j(t)$ (see below). In short, we need $m = \mathcal{O}(\sum w_i + \sum |W_j|)$ and m be larger than the maximum size of all C_z.

The construction of C_z below will guarantee that

$$\text{gates } \langle (z+1)n, m\rangle \text{ and } \langle (z+1)n, m-1\rangle \text{ evaluate } X(z) \text{ and } \neg X(z) \text{, respectively} \tag{3.31}$$

Let Y be the string that evaluates C (Y exists by the axiom MCV). Then the string X in (3.30) is defined by

$$X(z) \leftrightarrow Y((z+1)n, m) \tag{3.32}$$

We will need to prove that such X satisfies (3.30). The proof is by induction on z, and is clear from our construction of C_z given below.

In constructing C_z we may assume that string equality $Y = Z$ has been removed from φ by using the $\mathbf{V}^0$ axiom **SE** and the equality axioms. Further we can use De Morgan's laws to push negations in so that in both φ and $\neg\varphi$ negations appear only in front of atomic formulas. We proceed to construct the sub-circuits C_z by structural induction on the resulting formulas.

For the base case we consider the possible literals

$$s = t, \qquad s \neq t, \qquad s \leq t, \qquad t < s, \qquad Z(t), \qquad \neg Z(t) \tag{3.33}$$

The values of all variables except $|X|$ making up each term t are precomputed from the data $\vec{w}, z, \vec{W}$, so $t = t(|X|)$ is known as a polynomial in $|X|$ before constructing C_z. In general, the value v of a term t is represented in unary notation as a sequence T_t of b gates in C_z (for some precomputed upper bound b on t) whose output $T_t(i)$ satisfy

$$T_t(i) \leftrightarrow i = v \qquad \text{for } 0 \leq i < b$$

In case t is $|X|$, this sequence computes the following formulas:

$$T_{|X|}(i) \equiv X(i-1) \wedge \bigwedge_{j=i}^{z-1} \neg X(j)$$

where the first term $X(i-1)$ is omitted if $i = 0$. For example, for $i \geq 1$ gate $T_{|X|}(i)$ is an $\wedge$-gate with inputs from gates $\langle in, m\rangle$ and $\langle (j+1)n, m-1\rangle$ for $i \leq j \leq z-1$ (see (3.31)).

The sum $s+t$ or product st of two terms is easily computed from s and t; for example

$$T_{st}(i) \equiv \bigvee_{i=jk} (T_s(j) \wedge T_t(k))$$

Using these ideas sub-circuits C_z for the first four literals in (3.33) are easily constructed. Now consider the cases $Z(t)$ and $\neg Z(t)$. When Z is X: $X(i)$ and $\neg X(i)$ are outputs of gates $\langle (i+1)n, m\rangle$ and $\langle (i+1)n, m-1\rangle$, respectively. We can simplify the cases in which Z is a parameter variable W by preprocessing φ so that any occurrence of the form $W(t)$, where t contains $|X|$, is replaced by $\exists x \leq s(x = t \wedge W(x))$, where s is a term not involving $|X|$ which is an upper bound for t (and similarly for $\neg W(t)$). Thus for literals $W(t)$ and $\neg W(t)$ we may assume that t is a constant known "at compile time" and hence $W(t)$ and $\neg W(t)$ are outputs of appropriate gates $\langle 0, j\rangle$.

For the induction step, the cases where φ is $\varphi_1 \wedge \varphi_2$ and φ is $\varphi_1 \vee \varphi_2$ are easy. So it remains to consider the bounded quantifier cases, say

$$\varphi(z, X) \equiv \exists x \leq t\psi(x, z, X) \tag{3.34}$$

We may assume the bounding term t in (3.34) does not contain $|X|$ by replacing t by an upper bound s for t, and adding the conjunct $x \leq t$. Hence the value of t is known at compile time. By the induction hypothesis, $\mathbf{V}^0$ proves the existence of sub-circuits for $\psi(x, z, X)$. A circuit for $\exists x \le t\psi(x, z, X)$ can be constructed by placing circuits for each of $\psi(0, z, X), \psi(1, z, X), \ldots, \psi(t, z, X)$ side by side so that these formulas are evaluated in parallel. Then φ can be computed by a single $\vee$ gate from the outputs of these circuits. The circuit for $\neg\varphi$ is constructed using the equivalence $\neg\varphi \leftrightarrow \forall x \le t\neg\psi(x, z, X)$ and following the case $\forall x \le t$, which is handled similarly. This completes the description of the sub-circuits C_z. □

3.8 $\mathbf{VAC}^k$ and $\mathbf{VNC}^k$

Recall $\mathbf{AC}^k$ and $\mathbf{NC}^k$ from Definition 2.1.

Consider encoding a layered, monotone Boolean circuit C with $(d + 1)$ layers and n unbounded fan-in ($\wedge$ or $\vee$) gates on each layer. We need to specify the type (either $\wedge$ or $\vee$) of each gate, and the wires between the gates. Suppose that layer 0 contains the inputs which are specified by a string variable I of length $|I| \le n$. To encode the gates on other layers, there is a string variable G such that for $1 \le z \le d$, $G(z, x)$ holds if and only if gate x on layer z is an $\wedge$-gate (otherwise it is an $\vee$-gate). Also, the wires of C are encoded by a 3-dimensional array E: $\langle z, x, y\rangle \in E$ iff the output of gate x on layer z is connected to the input of gate y on layer $z + 1$.

The following algorithm computes the outputs of C using $(d + 1)$ loops: in loop z it identifies all gates on layer z which output 1. It starts by singling out the input gates with the value 1. Then in each subsequent loop $(z + 1)$ the algorithm identifies the following gates on layer $(z + 1)$:

- $\vee$–gates that have at least one input which is identified in loop z;
- $\wedge$–gates all of whose inputs are identified in loop z.

The formula $\delta_{LMCV}(n, d, E, G, I, Y)$ below formalizes this algorithm (LMCV stands for Layered Monotone Circuit Value). The 2–dimensional array Y stores the result of computation: For $1 \leq z \leq d$, row $Y^{[z]}$ contains the gates on layer z that output 1.

$$\delta_{LMCV}(n, d, E, G, I, Y) \equiv \forall x < n \forall z < d,\ (Y(0, x) \leftrightarrow I(x)) \wedge$$
$$[Y(z+1, x) \leftrightarrow (G(z+1, x) \wedge \forall u < n,\ E(z, u, x) \supset Y(z, u)) \vee$$
$$(\neg G(z+1, x) \wedge \exists u < n,\ E(z, u, x) \wedge Y(z, u))] \quad (3.35)$$

For $\mathbf{NC}^k$ we need the following formula which states that the circuit with underlying graph (n, d, E) has fan-in 2:

$$\mathit{Fanin2}(n, d, E) \equiv \forall z < d \forall x < n \exists u_1, u_2 < n \forall v < n,\ E(z, v, x) \supset v = u_1 \vee v = u_2$$

Recall (Example 2.18) that the function $\log(x) = \lfloor \log_2(x) \rfloor$ (or $|x|$) can be defined by a $\mathbf{\Sigma}_0^B$ formula. Let

$$\mathit{Lmcv}_k(n, E, G, I) = Y \leftrightarrow |Y| \leq \langle n, |n|^k \rangle \wedge \delta_{LMCV}(n, |n|^k, E, G, I, Y)$$
$$\mathit{Lmcv}_{k,2}(n, E, G, I) = Y \leftrightarrow (\neg \mathit{Fanin2}(n, d, E) \wedge Y = \varnothing) \vee$$
$$(\mathit{Fanin2}(n, d, E) \wedge |Y| \leq \langle n, |n|^k \rangle \wedge \delta_{LMCV}(n, |n|^k, E, G, I, Y))$$

Proposition 3.40. *For $k \geq 1$, Lmcv_k is $\mathbf{AC}^0$ many-one complete for $\mathbf{AC}^k$. For $k \geq 2$, $\mathit{Lmcv}_{k,2}$ is $\mathbf{AC}^0$ many-one complete for $\mathbf{NC}^k$.*

Proof Sketch. First, it is easy to see that every function in uniform $\mathbf{AC}^k$ (resp. $\mathbf{NC}^k$) is $\mathbf{AC}^0$ many-one reducible to Lmcv_k (resp. $\mathit{Lmcv}_{k,2}$). It remains to show that the Lmcv functions belong to the respective classes.

We show that Lmcv_1 is in $\mathbf{AC}^1$. The argument for Lmcv_k in general is similar. Consider a tuple (n, d, E, G, I) that encodes an unbounded fan-in circuit C of depth $d \leq c \log(n)$ for some $c \in \mathbb{N}$, and I encodes the inputs to C. For each $z \leq c \log(n)$ and $x \leq n$ we construct a constant-depth sub-circuit $K_{z,x}$ that computes the output of gate $\langle z, x \rangle$ (gate numbered x on layer z) in C. The inputs to $K_{z,x}$ are the bits of E (that specify the inputs to gate $\langle z, x \rangle$) and the output of other gates $K_{z-1,y}$. In particular, $K_{z,x}$

computes the following formula (recall that $G(z,x)$ holds iff gate $\langle z,x\rangle$ is an $\wedge$ gate):

$$(G(z,x)\wedge\bigwedge_{y<n}(E(z-1,y,x)\supset K(z-1,y)))\vee(\neg G(z,x)\wedge\bigvee_{y<n}(E(z-1,y,x)\wedge K(z-1,y)))$$

Our $\mathbf{AC}^1$ circuit that computes $Lmcv_1$ is obtained by stack the sub-circuits $K_{z,x}$ together. To make sure that it has depth $\log(m)$ where m is the length of the encoding of (n,d,E,G,I), we require that m is at least n^c whenever $(c-1)\log(n) < d \le c\log(n)$.

Now we show that $Lmcv_{2,2}$ is in $\mathbf{NC}^2$ (the argument for $Lmcv_{k,2}$ where $k > 2$ is similar). Suppose that (n,d,E,G,I) encodes a circuit C of of fan-in 2 and depth $d \le c\log(n)$ for some $c \in \mathbb{N}$, and I encodes the inputs to C. We use a $\log\log(n)$-depth unbounded fan-in circuit K that computes whether there is a path in E from a gate $\langle z,y\rangle$ to $\langle z',x\rangle$ for any $z < z' \le d$, $z' \le z+\log(n)$ and $x,y<n$

Using circuit K we can evaluate each $\log(n)$-depth sub-circuit of C rooted at gate $\langle z,x\rangle$ by a sub-circuit $K_{z,x}$ of depth $\mathcal{O}(\log(n)$. Our $\mathbf{NC}^2$ circuit computing $Lmcv_{2,2}$ is obtained by stacking the sub-circuits $K_{i\log(n),x}$ together (on top of K), for $i \le c$. Note that K can be simulated by a bounded fan-in circuit of depth $\mathcal{O}(\log(n))$. Again, we can make sure that the resulting circuit has depth $(\log(m))^2$, where m is the length of the encoding of (n,d,E,G,I), by requiring that $m \ge n^{c'}$ whenever $(c-1)\log(n) < d \le c\log(n)$ for some c' depending on c. □

Note that we do not know whether $Lmcv_{1,2}$ is in $\mathbf{NC}^1$.

Definition 3.41 ($\mathbf{VAC}^k$ **and** $\mathbf{VNC}^k$)**.** *For $k \ge 1$, the theory $\mathbf{VAC}^k$ has vocabulary $\mathcal{L}_A^2$ and is axiomatized by $\mathbf{V}^0$ and the axiom $\forall n\forall E\forall G\forall I\exists Y\, \delta_{LMCV}(n,|n|^k,E,G,I,Y)$. For $k \ge 2$, $\mathbf{VNC}^k$ has vocabulary $\mathcal{L}_A^2$ and is axiomatized by $\mathbf{V}^0$ and the axiom*

$\forall n\forall E\forall G\forall I(Fanin2(n,|n|^k,E)\supset\exists Y\, \delta_{LMCV}(n,|n|^k,E,G,I,Y))$.

It is straightforward to show that the functions $Lmcv_k^\star$ (resp. $Lmcv_{k,2}^\star$, for $k \ge 2$) is provably total in $\mathbf{VAC}^k$ (resp. $\mathbf{VNC}^k$, for $k \ge 2$). Thus these theories are instances of $\mathbf{VC}$ discussed in Section 3.2.

Chapter 4

Some Function Algebras

In this chapter we introduce the bounded number recursion (BNR) operation and use it to characterize a number of function classes inside $\mathbf{FL}$. Essentially each of these classes is the closure of the empty set of functions under $\mathbf{AC}^0$ reduction together with some (limited version of) BNR. This operation is defined in Section 4.1 where we prove the characterizations of $\mathbf{FL}$, $\mathbf{FTC}^0$ and $\mathbf{FAC}^0(2)$. The characterization of $\mathbf{FAC}^0(6)$ is more technical and is presented in Section 4.2. (The characterization of $\mathbf{FNC}^1$ is yet more complicated and follows from the result in Chapter 5.) Finally we show in Section 4.3 that the use of $\mathbf{AC}^0$ reduction in these characterizations can be replaced by only two operations: composition and a newly defined operation called *string comprehension*.

As mentioned in Section 1.1.2, the algebras for $\mathbf{FAC}^0(2)$ and $\mathbf{FAC}^0(6)$ are two-sorted version of the algebras given in [CT95, PW85]; here we carry out the proofs in more detail. Also, the algebra for $\mathbf{FL}$ is two-sorted version of Lind's characterization [Per05]. The algebra for $\mathbf{FTC}^0$ is new.

These algebras can be used to obtained universal theories that are equivalent to $\overline{\mathbf{VC}}$ introduced in Chapter 3.2. In fact, in the next chapter we use the algebra for $\mathbf{FNC}^1$ to develop the theory $\mathbf{VALV}$ and prove that $\mathbf{VALV}$ is a conservative extension of $\mathbf{VNC}^1$. In effect, our result there shows that $\mathbf{VALV}$ is equivalent to $\overline{\mathbf{VNC}}^1$.

Notation In this chapter, $\varnothing$ denotes the empty set of functions (as oppose to the empty-set element of the second sort used in the previous chapter).

4.1 Bounded Number Recursion

Definition 4.1 (Bounded Number Recursion)**.** *For a number term $t(y, \vec{x}, \vec{X})$ and number functions $g(\vec{x}, \vec{X})$ and $h(y, z, \vec{x}, \vec{X})$, we say that a number function $f(y, \vec{x}, \vec{X})$ is obtained by t-*bounded number recursion *(t-BNR) from g and h if $f < t$ and*

$$f(0, \vec{x}, \vec{X}) = g(\vec{x}, \vec{X}) \tag{4.1}$$

$$f(y + 1, \vec{x}, \vec{X}) = h(y, f(y, \vec{x}, \vec{X}), \vec{x}, \vec{X}) \tag{4.2}$$

If t is a polynomial in $\vec{x}, |\vec{X}|$ then we also say that f is obtained from g, h by polynomial-bounded number recursion *(pBNR).*

Theorem 4.2. **FL** *is precisely the closure of $\varnothing$ under $\mathbf{AC}^0$ reduction and pBNR.*

Proof. First, it is straightforward to show that *SinglePath* (Definition 3.33 on page 50) can be obtained from $\mathbf{FAC}^0$ by pBNR. For the other direction, we prove by induction on the number of the applications of the number recursion operation. For the induction step, suppose that f is obtained from **FL** functions g, h by t-BNR, for some polynomial $t(\vec{x}, |\vec{X}|)$. Consider the graph $(\langle t, t\rangle, E)$ where $E(\langle y, u\rangle, \langle y + 1, v\rangle)$ iff $v = h(u, \vec{x}, \vec{X})$, for $u, v, y < t$. Then $f(y) = z$ iff $\langle y, z\rangle$ is reachable from $\langle 0, g\rangle$. Hence f is $\mathbf{AC}^0$ reducible to $\{g, h, \mathit{SinglePath}\}$. □

Theorem 4.3. $\mathbf{FAC}^0(2)$ *is precisely the closure $\varnothing$ under $\mathbf{AC}^0$ reduction and 2-BNR.*

Proof. For one direction, it is easy to show that the function mod_2 (Proposition 3.19 on page 38) can be obtained from $\mathbf{FAC}^0$ using 2-BNR. We prove the other direction by induction on the number of applications of the 2-BNR operation. For the induction step, suppose that f is obtained from $\mathbf{FAC}^0(2)$ functions g, h by 2-BNR as in Definition 4.1. For $y \geq 1$, let (we drop mention of $\vec{x}, \vec{X}$)

$$z = max(\{0\} \cup \{u < y \ : \ h(u,0) = h(u,1)\})$$

$$n = mod_2(y, \{u \ : \ z < u < y \wedge h(u,0) \neq 0\})$$

$$v = \begin{cases} g & \text{if } z = 0 \\ h(z,0) & \text{otherwise} \end{cases}$$

Then $f(y) = 0$ iff either (i) $v = 0$ and $n = 0$, or (ii) $v = 1$ and $n = 1$. In other words, f can be obtained from g, h and mod_2 by $\mathbf{AC}^0$ reduction. □

The characterization of $\mathbf{FTC}^0$ is stated using the following operation which is a special instance of polynomial-bounded number recursion (take $h(y,z) = y + z$).

Definition 4.4 (Number Summation)**.** *For a number function $f(y, \vec{x}, \vec{X})$, the function $\mathsf{sum}_f(y, \vec{x}, \vec{X})$ below is said to be defined from f* by number summation*, or just* summation*:*

$$\mathsf{sum}_f(y, \vec{x}, \vec{X}) = \sum_{z=0}^{y} f(z, \vec{x}, \vec{X})$$

Theorem 4.5. *A function is in $\mathbf{FTC}^0$ iff it can be obtained from $\varnothing$ by $\mathbf{AC}^0$ reduction and number summation.*

Proof. The ($\Longrightarrow$) direction follows from the fact that (where we write 0 for $\perp$ and 1 for $\top$):

$$numones(x, X) = \sum_{y=0}^{x} X(y)$$

We prove the other direction by induction on the number of applications of the summation operation. For the induction step it suffices to show that sum_f can be obtained from f and $numones$ using $\mathbf{AC}^0$ reduction. Define a string W that contains the right number of bits:

$$W(xa + v) \leftrightarrow x \leq y, v < f(x)$$

for some $a > max(\{f(x) \ : \ x < y\})$. Then it is easy to verify that $\mathsf{sum}_f(y) = numones((y+1)a, W)$. $\square$

4.2 Number Recursion for Permutations

Definition 4.6. *For $2 \le k \in \mathbb{N}$, we say that a function $h(x)$ is a k-*permutation *(or just* permutation*) if on domain $\{0, \dots, k-1\}$, the range of h is $\{0, \dots, k-1\}$. ${}^k k$ denotes the set of all functions $\{0, \dots, k-1\} \to \{0, \dots, k-1\}$, and $S_k \subseteq {}^k k$ the set of all k-permutations.*

We now show that the (general) k-BNR can be simulated by $\mathbf{AC}^0$ reduction and BNR for k-permutations (that is, k-BNR as in Definition 4.1 but $g(\vec{x}, \vec{X}) \le k-1$ and $h_{y,\vec{x},\vec{X}}(z) = h(y, z, \vec{x}, \vec{X})$ is a k-permutation). In the following discussion we will often drop mentions of $\vec{x}, \vec{X}$. First, we show that if for all y, $h_y(z) = h(y,z)$ is not a k-permutation, then k-BNR using h can be replaced by $(k-1)$-BNR.

Lemma 4.7. *Let be $h(y, z, \vec{x}, \vec{X})$ be a function such that for all y, $h_{y,\vec{x},\vec{X}} \notin S_k$, where $h_{y,\vec{x},\vec{X}}(z) = h(y, z, \vec{x}, \vec{X})$. Suppose that $k \ge 2$ and that $f(y, \vec{x}, \vec{X})$ is obtained from $g(\vec{x}, \vec{X})$ and $h(y, z, \vec{x}, \vec{X})$ by k-BNR. Then f can be obtained from g, h by $\mathbf{AC}^0$ reduction and $(k-1)$-BNR.*

Proof. Intuitively, since h_y are not k-permutation, we need the values of h_y only on a $(k-1)$-element subset of $\{0, 1, \dots, k-1\}$. So define $\ell(y)$ to be the least element in $\{0, 1, \dots, k-1\}$ that can be discarded from the domain of h_y without affecting the computation of f:

$$\ell(0) = \min\{w \ : \ w \neq g\} \wedge \ell(y+1) = \min\{w \ : \ w \notin h_y(\{0, 1, \dots, k-1\})\}$$

Define $h'_y \in {}^{(k-1)}(k-1)$ and bijection $r_y(z)$ (where $[k] = \{0, 1, \dots, k-1\}$):

$$[k] \setminus \{\ell(y)\} \xrightarrow{r_y} [k-1] \xrightarrow{h'_y} [k-1] \xrightarrow{r_{y+1}^{-1}} [k] \setminus \{\ell(y+1)\}$$

so that on domain $[k] \setminus \{\ell(y)\}$, $r_{y+1}^{-1} \circ h'_y \circ r_y = h_y$. Thus

$$r_y(z) = \begin{cases} z & \text{if } z < \ell(y) \\ z-1 & \text{if } \ell(y) < z < k \end{cases} \qquad \text{and} \qquad h'_y = r_{y+1} \circ h_y \circ r_y^{-1}$$

Let f' be obtained from g and h' by $(k-1)$-BNR, then it is easy to see that $f(y) = r_y^{-1}(f'(y))$. □

Now we show that if h_0 is not a k-permutation, then k-BNR using h can be simulated by $(k-1)$-BNR and number recursion using k-permutation:

Lemma 4.8. *Let $2 \le k \in \mathbb{N}$ and $g(\vec{x}, \vec{X}), h(y, z, \vec{x}, \vec{X})$ be functions such that $h_{0,\vec{x},\vec{X}}(z) \notin S_k$, where $h_{0,\vec{x},\vec{X}}(z) = h(0, z, \vec{x}, \vec{X})$. Suppose that f is obtained from g, h by k-BNR. Then f can also be obtained from g and h by $\mathbf{AC}^0$ reduction, $(k-1)$-BNR and number recursion using k-permutations.*

Proof. Since $h_0(z) = h(0, z)$ is not a k-permutation, for each $y \ge 0$ we need the values of h_y on only a $(k-1)$-elements subset of $\{0, 1, \dots, k-1\}$. The issue is to uniformly identify these subsets, then we can use Lemma 4.7 above. First, we identify a redundant number $\ell(y) \le k-1$ that can be removed from the domain of h_{y+1} without affecting the computation of f.

Let $m(y) = max\{u \le y : h_u \notin S_k\}$. Then $0 \le m(y) \le y$ for $y \ge 0$. Consider the case where $m(y) = y$ (i.e., h_y is not a k-permutation). Define

$$\ell(y) = \min\{w \le k-1 : \neg\exists z < k \; h_y(z) = w\}$$

Now suppose that $m(y) < y$, then h_u are k-permutations, for $m(y) < u \le y$. The redundant value in the domain of $h_{m(y)+1}$ (which exists because h_y is not a k-permutation) propagates through $h_{m(y)+2}, \dots, h_y$. These redundant values can be computed by number recursion using k-permutations as follows. Let $\ell'(u)$ be obtained by number recursion using k-permutation:

$$\ell'(0) = \min\{w \le k-1 \;:\; \forall z < k \; h_{m(y)}(z) \ne w\} \wedge \forall u \ge 0,\; \ell'(u+1) = h(u, \ell'(u))$$

Now $\ell(y) = \ell'(y \dot{-} m(y))$ can be safely removed from the range of h_y.

Now for $z < k$ let

$$h'(y,z) = \begin{cases} h(y,z) & \text{if } h(y,z) \neq \ell(y) \\ k & \text{otherwise} \end{cases}$$

Then $h'_y \notin S_k$ (for all y), and it can be shown that f is obtained from g and h' by k-BNR. By Lemma 4.7, f can also be obtained from g and h' by $\mathbf{AC}^0$ reduction and $(k-1)$-BNR. □

Theorem 4.9. *Suppose that $1 \leq k \in \mathbb{N}$ and f is obtained from g and h using k-BNR. Then f can be obtained from g and h by $\mathbf{AC}^0$ reduction and k-BNR using k-permutations.*

Proof. We prove by induction on k. The base case ($k = 1$) is trivially true.

For the induction step, assume that the theorem is true for $(k-1)$, we prove it for k. Suppose that for some y, $h_y(z) = h(y,z)$ is not a k-permutation. The idea is to identify the first point m where h_m is not a k-permutation:

$$m = \min(\{u < y \ : \ h_u \notin S_k\} \cup \{y\})$$

Then h_x is a k-permutation for $x < m$, and for $x \geq m$ we can use Lemma 4.8 above and the induction hypothesis. □

Now we show that $\mathbf{FAC}^0(6)$ is closed under 3-BNR and 4-BNR. Essentially, the proofs are based on the solvability of the groups associated with these operations (i.e., S_3, S_4).

Theorem 4.10. $\mathbf{FAC}^0(6)$ *is closed under 3-BNR.*

Proof. By Theorem 4.9, it suffices to consider 3-BNR for 3-permutations. Suppose that g and $h(x,z)$ are in $\mathbf{FAC}^0(6)$, $g < 3$ and $h_x(z) = h(x,z) \in S_3$ for all x. Let f be obtained from g and h using 3-BNR. We show that f is also in $\mathbf{FAC}^0(6)$.

Note that

$$f(x) = h_{x-1} \circ h_{x-2} \circ \ldots \circ h_0(g)$$

Let A_3 be the normal subgroup of S_3 which consists of the even permutations, and e be the identity of S_3. Then $S_3 = \{e, (0\ 1)\} \times A_3$, i.e., every element in S_3 is the product $\gamma \circ \sigma$ where $\gamma \in \{e, (0\ 1)\}$ and $\sigma \in A_3$. In particular,

$$h_u = (0\ 1)^{\epsilon_u} \circ \sigma_u \qquad \text{where } \epsilon_u \in \{0, 1\} \text{ and } \sigma_u \in A_3.$$

For $u < x$ let

$$\delta_u = (0\ 1)^{\epsilon_{x-1}} \circ \ldots \circ (0\ 1)^{\epsilon_u} = (0\ 1)^{(\epsilon_{x-1}+\ldots+\epsilon_u) \mod 2}$$

Then δ_u can be computed in $\mathbf{V}^0(2)$, and it is easy to see that

$$f(x) = \left(\prod_{u=x-1}^{u=0} (\delta_u \circ \sigma_u \circ \delta_u^{-1}) \right) \circ \delta_0(g) \tag{4.3}$$

We compute $\eta_u = \delta_u \circ \sigma_u \circ \delta_u^{-1}$ simultaneously for all $u < x$. Here $\eta_u \in A_3$, and therefore is of the form

$$(0\ 1\ 2)^{\lambda_u} \qquad \text{where } \lambda_u \in \{0, 1, 2\}.$$

As a result, $\prod_{u=x-1}^{u=0} \eta_u$ can be computed in $\mathbf{FAC}^0(3)$ by computing $(\lambda_{x-1} + \ldots + \lambda_0)$ mod 3. This shows that $f(x)$ can be computed in $\mathbf{FAC}^0(6)$. □

Theorem 4.11. $\mathbf{FAC}^0(6)$ *is closed under 4-BNR.*

Proof. Again, by Theorem 4.9 it suffices to show that $\mathbf{FAC}^0(6)$ is closed under 4-BNR for 4-permutations. Let g and $h(x, z)$ be in $\mathbf{FAC}^0(6)$, $g < 4$ and $h_x(z) = h(x, z) \in S_4$ for all x. Suppose that f is defined from g and h using 4-BNR.

As in the proof of Theorem 4.10, we need to compute (4.3) but now σ_u are in A_4, the normal subgroup of S_4 that consists of all even permutations. As before, $\eta_u = \delta_u \circ \sigma_u \circ \delta_u^{-1}$ can be computed simultaneously for all $u < x$, but here $\eta_u \in A_4$. Our next step is to compute

$$\prod_{u=x-1}^{u=0} \eta_u, \qquad (\text{for } \eta_u \in A_4) \tag{4.4}$$

Using the same idea as in the proof of Theorem 4.10, i.e., using the fact that A_4 contains a normal subgroup $V = \{e, (0\ 1)(2\ 3), (0\ 2)(1\ 3), (0\ 3)(1\ 2)\}$ (the Klein group). In particular, $A_4 = \{e, (0\ 1\ 2), (0\ 2\ 1)\} \times V$.

We repeat the steps in the proof of Theorem 4.10 and write $\eta_u = (0\ 1\ 2)^{\epsilon'_u} \circ \sigma'_u$, where $\epsilon'_v \in \{0, 1, 2\}$ and $\sigma'_u \in V$. Also, for $u < x$ let

$$\delta'_u = (0\ 1\ 2)^{\epsilon'_{x-1}} \circ \ldots \circ (0\ 1\ 2)^{\epsilon'_u} = (0\ 1\ 2)^{(\epsilon'_{x-1}+\ldots+\epsilon'_u) \mod 3}$$

Thus δ'_u can be computed in $\mathbf{FAC}^0(6)$ by computing $(\epsilon'_{x-1} + \ldots + \epsilon'_u) \mod 3$. Now (4.4) can be rewritten as (see (4.3))

$$\left(\prod_{u=x-1}^{u=0} (\delta'_u \circ \sigma'_u \circ (\delta'_u)^{-1}) \right) \circ \delta'_0$$

Here $\eta'_u = \delta'_u \circ \sigma'_u \circ (\delta'_u)^{-1}$ are in V, so the above product can be computed in $\mathbf{FAC}^0(2)$ using the fact that V is Abelian and its members have order 2. □

Corollary 4.12. $\mathbf{FAC}^0(6)$ *is the closure* $\varnothing$ *under* $\mathbf{AC}^0$ *reduction and 4-BNR.*

Proof. For one direction, it is straightforward to show that mod_6 can be obtained from $\mathbf{AC}^0$ functions by 3-bounded number recursion. The other direction follows from Theorem 4.11 above. □

4.3 The String Comprehension Operation

In many cases (such as all previous results in this chapter), $\mathbf{AC}^0$ reduction is equivalent to the combination of composition and the following operation:

Definition 4.13 (String Comprehension)**.** *For a number function* $f(x)$*, the* string comprehension of f *is the string function*

$$F(y) = \{f(x) : x \leq y\}$$

Note that if f is polynomially bounded, then so is F.

For example, consider the $\mathbf{\Sigma}_0^B$ formula $\delta_{parity}(X, Y)$ (3.20) on page 38. As a function of X, $Y = F(|X|, X)$, where F is the string comprehension of

$$f(x, X) = \begin{cases} x & \text{if } x > 0 \text{ and the number of 1 in } X(0), \ldots, X(x-1) \text{ is odd} \\ |X| + 1 & \text{otherwise} \end{cases}$$

Theorem 4.14. *Suppose that $\mathcal{L}$ is a class of polynomially bounded functions that includes* $\mathbf{FAC}^0$. *Then a function is* $\mathbf{AC}^0$*-reducible to $\mathcal{L}$ iff it can be obtained from $\mathcal{L}$ by finitely many applications of composition and string comprehension.*

Proof. For the IF direction, it suffices to prove that a function obtained from input functions by either of the operations composition or string comprehension is $\mathbf{\Sigma}_0^B$-definable from the input functions.

For composition, suppose

$$F(\vec{x}, \vec{X}) = G(h_1(\vec{x}, \vec{X}), \dots, h_k(\vec{x}, \vec{X}), H_1(\vec{x}, \vec{X}), \dots, H_m(\vec{x}, \vec{X}))$$

where G and $h_1, \dots, h_k$, $H_1, \dots, H_m$ are polynomially bounded. Then F is also polynomially bounded, and its bit graph $F(\vec{x}, \vec{X})(z)$ is represented by the open formula

$$G(h_1(\vec{x}, \vec{X}), \dots, h_k(\vec{x}, \vec{X}), H_1(\vec{x}, \vec{X}), \dots, H_m(\vec{x}, \vec{X}))(z)$$

(A similar argument works for a number function f.)

For string comprehension, suppose that $f(x)$ is a polynomially bounded number function. As noted before, the string comprehension $F(y)$ of f is also polynomially bounded, and it has bit graph

$$F(y)(z) \leftrightarrow z < t \wedge \exists x \leq y \; z = f(x)$$

where t is the bounding term for F. Hence F is also $\mathbf{\Sigma}_0^B$-definable from f.

For the ONLY IF direction, it suffices to show that if $\mathcal{L} \supseteq \mathbf{FAC}^0$ and F (or f) is $\mathbf{\Sigma}_0^B$-definable from $\mathcal{L}$, then F (resp. f) can be obtained from $\mathcal{L}$ by composition and string comprehension.

Claim : If $\mathcal{L} \supseteq \mathbf{FAC}^0$ and $\varphi(\vec{z}, \vec{X})$ is a $\mathbf{\Sigma}_0^B(\mathcal{L})$ formula, then the characteristic function c_φ defined by

$$c_\varphi(\vec{z}, \vec{Z}) = \begin{cases} 1 & \text{if } \varphi(\vec{z}, \vec{Z}) \\ 0 & \text{otherwise} \end{cases}$$

can be obtained from $\mathcal{L}$ by composition.

The claim holds because $c_\psi(\vec{x}, \vec{X})$ is in $\mathbf{FAC}^0$ for every $\mathbf{\Sigma}_0^B(\mathcal{L}_A^2)$-formula ψ, and (by structural induction on φ) it is clear that for every $\mathbf{\Sigma}_0^B(\mathcal{L})$-formula $\varphi(\vec{z}, \vec{Z})$ there is a $\mathbf{\Sigma}_0^B(\mathcal{L}_A^2)$-formula $\psi(\vec{x}, \vec{X})$ such that

$$\varphi(\vec{z}, \vec{Z}) \leftrightarrow \psi(\vec{s}, \vec{T})$$

for some $\mathcal{L}$-terms $\vec{s}$ and $\vec{T}$. Hence $c_\varphi(\vec{z}, \vec{Z}) = c_\psi(\vec{s}, \vec{T})$.

Now suppose that F is $\mathbf{\Sigma}_0^B$-definable from $\mathcal{L}$, so for some $\mathcal{L}_A^2$ term $t(\vec{z}, \vec{X})$ and $\mathbf{\Sigma}_0^B(\mathcal{L})$ formula $\varphi(x, \vec{z}, \vec{X})$:

$$F(\vec{z}, \vec{X})(x) \leftrightarrow x < t \wedge \varphi(x, \vec{z}, \vec{X})$$

Define the number function f by cases as follows:

$$f(x, \vec{z}, \vec{X}) = \begin{cases} x & \text{if } \varphi(x, \vec{z}, \vec{X}) \\ t & \text{if } \neg\varphi(x, \vec{z}, \vec{X}) \end{cases}$$

Then by the claim, f can be obtained from $\mathcal{L}$ by composition:

$$f(x, \vec{z}, \vec{X}) = g(x, c_\varphi, t, c_{\neg\varphi})$$

where g is the $\mathbf{FAC}^0$ function: $g(x, y, z, w) = x \cdot y + z \cdot w$. Now

$$F(\vec{z}, \vec{X}) = Cut(t, G(t, \vec{z}, \vec{X}))$$

where $G(y, \vec{z}, \vec{X})$ is the string comprehension of $f(x, \vec{z}, \vec{X})$, and Cut (see (3.7) on page 29) is the $\mathbf{FAC}^0$ function defined by

$$Cut(x, X)(z) \leftrightarrow z < x \wedge X(z)$$

It remains to show that if a number function f is $\mathbf{\Sigma}_0^B$-definable from $\mathcal{L}$ then f can be obtained from $\mathcal{L}$ by composition and string comprehension. Suppose f satisfies

$$y = f(\vec{z}, \vec{X}) \leftrightarrow y < t \wedge \varphi(y, \vec{z}, \vec{X})$$

where $t = t(\vec{z}, \vec{X})$ is a $\mathcal{L}_A^2$ term and φ is a $\mathbf{\Sigma}_0^B(\mathcal{L})$ formula. Use the claim to define $c_\varphi(y, \vec{z}, \vec{X})$ by composition from $\mathcal{L}$, and define g by

$$g(x, \vec{z}, \vec{X}) = x \cdot c_\varphi(x, \vec{z}, \vec{X})$$

Then

$$f(\vec{z}, \vec{X}) = |G(t, \vec{z}, \vec{X})| \dot{-} 1$$

where $G(y, \vec{z}, \vec{X})$ is the string comprehension of $g(x, \vec{z}, \vec{X})$. □

Chapter 5

$\mathbf{VNC}^1 \overset{\text{RSUV}}{\simeq} \mathbf{QALV}$

The equivalence between a single-sorted theory and a two-sorted theory are known as their RSUV isomorphism [Tak93, Raz93, Kra90]. We briefly define this notion in Section 5.1, for more details see [CN06]. The RSUV isomorphism between $\mathbf{VNC}^1$ and $\mathbf{QALV}$ is proved as follows. We first introduce a two-sorted theory called $\mathbf{VALV}$ (Section 5.2) which is easily shown to be RSUV isomorphic to $\mathbf{QALV}$. Then the major task is to show that $\mathbf{VALV}$ is a conservative extension of $\mathbf{VNC}^1$. To show that $\mathbf{VALV}$ extends $\mathbf{VNC}^1$, we need to formalize and prove the correctness of Barrington's reduction from the Boolean Sentence Value Problem to the word problem for S_5. This is carried out in Section 5.3. The fact that $\mathbf{VALV}$ is conservative over $\mathbf{VNC}^1$ follows from the results proved in Chapter 3.

5.1 RSUV Isomorphism

Essentially, to show that a single-sorted theory $\mathcal{T}_1$ is RSUV isomorphic to a two-sorted theory $\mathcal{T}_2$ (i.e., $\mathcal{T}_1 \overset{\text{RSUV}}{\simeq} \mathcal{T}_2$) we need to (a) construct from each model $\mathcal{M}$ of $\mathcal{T}_1$ a model $\mathcal{M}^\sharp$ of $\mathcal{T}_2$ whose second sort universe is the universe M of $\mathcal{M}$, and whose first sort universe is the subset $\log(M) = \{|u| \mid u \in M\}$; and (b) construct from each model $\mathcal{N}$ of $\mathcal{T}_2$ a model $\mathcal{N}^\flat$ of $\mathcal{T}_1$ whose universe is the second sort universe of $\mathcal{N}$. These constructions have the property that $\mathcal{M}$ and $(\mathcal{M}^\sharp)^\flat$ are isomorphic, and so are $\mathcal{N}$ and $(\mathcal{N}^\flat)^\sharp$.

These semantic mappings between models are associated with syntactic translations of formulas between the languages of $\mathcal{T}_1$ and $\mathcal{T}_2$. In particular, each two-sorted formula φ is translated into a single-sorted formula $\varphi^\flat$ such that for any model $\mathcal{M}$ of $\mathcal{T}_1$:

$$\mathcal{M}^\sharp \models \forall\varphi \text{ if and only if } \mathcal{M} \models \forall\varphi^\flat$$

and each single-sorted formula ψ is translated into a two-sorted formula $\psi^\sharp$ so that for any model $\mathcal{N}$ of $\mathcal{T}_2$:

$$\mathcal{N}^\flat \models \forall\psi \text{ if and only if } \mathcal{N} \models \forall\psi^\sharp$$

For example, the single sorted formula $x \le y$ is translated into $X \le_2 Y$, where

$$\begin{aligned} X \le_2 Y \;\equiv\; & X =_2 Y \vee |X| < |Y| \vee \\ & |X| = |Y| \wedge \exists x < |X|(Y(x) \wedge \neg X(x) \wedge \forall y < |X|, x < y \supset (X(y) \leftrightarrow Y(y))) \end{aligned} \tag{5.1}$$

It turns out that the hard work in proving RSUV isomorphism is often in interpreting certain functions in the appropriate structures, e.g., interpreting the multiplication function in $\mathbf{VTC}^0$ [Ngu04, NC05]. In the case of **QALV** and $\mathbf{VNC}^1$, a difficulty is in interpreting the function *Fval* in a model for **QALV**.

5.2 The Theory VALV

VALV is defined in style of $\overline{\mathbf{VNC}}^1$ (an instance of $\overline{\mathbf{VC}}$, Definition 3.9), but using the 5-bounded number recursion operation (Definition 4.1) instead of the function *Fval* (Definition 3.22). Suppose that $f_{g,h}(y, \vec{x}, \vec{X})$ is defined from $g(\vec{x}, \vec{X})$ and $h(y, z, \vec{x}, \vec{X})$ by 5-BNR. Then $f_{g,h}$ has the following defining axiom (we drop mention of $\vec{x}, \vec{X}$, and write f for $f_{g,h}$):

$$(g < 5 \wedge f(0) = g) \vee (g \ge 5 \wedge f(0) = 0) \tag{5.2}$$

$$(h(y, f(y)) < 5 \wedge f(y+1) = h(y, f(y))) \vee (h(y, f(y)) \ge 5 \wedge f(y+1) = 0) \tag{5.3}$$

Definition 5.1. $\mathcal{L}_{\mathbf{FALV}}$ *is the smallest set that satisfies*

1) $\mathcal{L}_{\mathbf{FALV}}$ *includes* $\mathcal{L}_A^2 \cup \{pd, f_{\mathbf{SE}}\}$.
2) *For each open formula* $\varphi(z, \vec{x}, \vec{X})$ *over* $\mathcal{L}_{\mathbf{FALV}}$ *and term* $t = t(\vec{x}, \vec{X})$ *of* $\mathcal{L}_A^2$ *there is a string function* $F_{\varphi,t}$ *and a number function* $f_{\varphi,t}$ *in* $\mathcal{L}_{\mathbf{FALV}}$.
3) *For any two functions* g, h *of* $\mathcal{L}_{\mathbf{FALV}}$*, there is a number function* $f_{g,h}$ *in* $\mathcal{L}_{\mathbf{FALV}}$.

Definition 5.2. **VALV** *is the theory over* $\mathcal{L}_{\mathbf{FALV}}$ *with the following set of axioms:* **B1**-**B11**, **L1**, **L2** *(Figure 2.1),* (2.8), (2.9), (2.10), (2.11) *for each function* $F_{\varphi,t}$, (2.12) *for each function* $f_{\varphi,t}$*, and* (5.2), (5.3) *for each function* $f_{g,h}$ *of* $\mathcal{L}_{\mathbf{FALV}}$.

Theorem 5.3. **VALV** *proves* $\mathbf{\Sigma}_0^B(\mathcal{L}_{\mathbf{FALV}})$-**COMP** *and* $\mathbf{\Sigma}_0^B(\mathcal{L}_{\mathbf{FALV}})$-**IND**.

Proof. The fact that **VALV** proves $\mathbf{\Sigma}_0^B(\mathcal{L}_{\mathbf{FALV}})$-**COMP** can be proved as for Lemma 3.10. The fact that **VALV** proves $\mathbf{\Sigma}_0^B(\mathcal{L}_{\mathbf{FALV}})$-**IND** now follows from Theorem 2.14. □

5.2.1 QALV

The single-sorted theory $\mathbf{ALV}'$ [Clo93] is an equational theory whose axioms include the defining axioms for some basic $\mathbf{AC}^0$ functions and the functions defined inductively by composition, Concatenation Recursion on Notation (CRN) and k-Bounded Recursion on Notation (k-BRN). **QALV** is a (single-sorted) first-order theory whose non-logical symbols are those of $\mathbf{ALV}'$ and whose axioms are the universal closure of the axioms of $\mathbf{ALV}'$ (together with some basic axioms, see [Coo98]).

Let $s_0(x) = 2x, s_1(x) = 2x + 1$. Suppose that $h_0(x), h_1(x) \leq 1$. Then $f(x)$ is defined by CRN from g, h_0 and h_1 by CRN if (here f, g, h_0, h_1 might have other parameters):

$$f(0) = g, \qquad f(1) = s_{h_1(x)}(g), \qquad \text{and} \qquad f(s_i(x)) = s_{h_i(x)}(f(x)) \text{ for } x > 0 \tag{5.4}$$

Intuitively, if $g > 0$, then $|f(x)| = |g| + |x|$; otherwise

$$|f(x)| = |x| \dot{-} |\min\{z : h_0(z) > 0 \vee h_1(z) > 0\}|$$

In addition, the bits of $f(x)$ are computed in parallel from the bits of g and x using $h_i(x)$. So this operation corresponds to taking the $\mathbf{AC}^0$-closure (i.e., defining $f_{\varphi,t}$ and $F_{\varphi,t}$ in Definition 5.2).

k-BRN can be seen as the single-sorted version of our $(k+1)$-BNR (see the next section): a function f is defined by k-BRN from g, h_0 and h_1 provided that $f(x) \leq k$ for all x, and

$$f(0) = g, \qquad f(1) = h_1(0, g), \qquad \text{and} \qquad f(s_i(x)) = h_i(x, f(x)) \;\; \text{for} \;\; x > 0 \tag{5.5}$$

Theorem 5.4. **VALV** *and* **QALV** *are RSUV isomorphic.*

In the next section we outline a proof of this theorem.

5.2.2 $\mathbf{QALV} \overset{\text{RSUV}}{\simeq} \mathbf{VALV}$

We refer to [Tak93, Raz93, Kra90, CN06] for back-and-forth translations between single-sorted and two-sorted theories. The translations of initial functions of **QALV** and the functions in $\mathcal{L}_A^2$ are straightforward, and we outline here only the translations of functions that are obtained by BRN, CRN and composition (for functions in **QALV**) and $\mathbf{AC}^0$-reduction (i.e., $F_{\varphi,t}$ and $f_{\varphi,t}$) and BNR (for functions in **VALV**). (Note that **VALV** is defined using 5-BNR while **QALV** is defined using k-BRN for all $k \in \mathbb{N}$. The fact that **VALV** extends $\mathbf{VNC}^1$ (Section 5.3) shows that 5-BNR simulates k-BNR for all $k > 5$, because it is easy to show that if f is obtained from g and h by k-BNR where g, h are provably total in $\mathbf{VNC}^1$, then f is also provably total in $\mathbf{VNC}^1$.)

First we show how to interpret functions of **QALV** in **VALV**. Suppose that $f(x)$ is obtained from g and $h_i(x, z)$ using BRN as in (5.5). Using the terminologies of Section 5.1, the functions $g, h_i(x, z)$ are translated into string functions $g^\sharp, h_i^\sharp(X, Z)$ in the two-sorted setting. These functions have values bounded by $k^\sharp$ which is the set: $k^\sharp = \{i : Bit(i, k)\}$, where $Bit(i, k)$ holds iff the i-th least significant bit of k is 1; for example, $5^\sharp = \{2, 0\}$. Here we compare two strings X, Y using $\leq_2$ defined in (5.1). We will briefly show how to obtain the translation $f^\sharp(X)$ of $f(x)$ from $g^\sharp$ and $h_i^\sharp(X,Z)$ using $(k+1)$-BNR and $\mathbf{AC}^0$ reduction.

Because $g^\sharp$ and $h_i^\sharp(X,Z)$ are bounded by a constant string, we can treat them as number functions. Indeed, define number functions g' and $h_i'(X,z)$ so that $g' = g$ and $h_i'(x^\sharp, z) = h_i(x,z)$. Then g' and $h_i'(X,z)$ can be obtained from $g^\sharp$ and $h_i^\sharp(X,Z)$ by $\mathbf{AC}^0$ reduction. Let the number function $h''(i,X,z)$ be defined as follows (for $0 \le i \le |X|-1$):

$$h''(i,X,z) = \begin{cases} h_1'(\mathit{Trim}(i,X), z) & \text{if } X(|X| \dot{-} i \dot{-} 1) \\ h_0'(\mathit{Trim}(i,X), z) & \text{otherwise} \end{cases}$$

where $\mathit{Trim}(i,X) = \{z \ : \ z + (|X| \dot{-} i) \in X\}$ is the substring of the i most significant bits of X (for $0 \le i \le |X|$).

Define by $(k+1)$-BNR a number function $\ell(i,X)$ as follows

$$\ell(0,X) = g', \qquad \ell(i+1,X) = h''(i,X,\ell(i,X)) \ \text{ for } 0 \le i \le |X|-1.$$

Then $f^\sharp(\mathit{Trim}(i,X)) = (\ell(i,X))^\sharp$, so $f^\sharp(X) = (\ell(|X|,X))^\sharp$ (here $(\ell(i,X))^\sharp$ denotes the set $\{j \ : \ \mathit{Bit}(j,\ell(i,X))\}$).

Next, suppose that $f(x)$ is obtained from g and $h_i(x)$ by CRN as in (5.4). It is easy to define the bits of the string functions $f^\sharp(X)$ using $g^\sharp$ and $H_i(z,X) = h_i^\sharp(\mathit{Trim}(z,X))$. Finally, suppose that f is obtained by composition, i.e., $f = g(h_1, h_2, \ldots, h_k)$. Then $f^\sharp$ is also obtained from $g^\sharp, h_1^\sharp, h_2^\sharp, \ldots, h_k^\sharp$ by composition, and it is clear that **FALV** is closed under composition.

For the other direction, first, suppose that $f(y)$ is obtained from g and $h(y,z)$ by 5-BNR (we omit the parameters $\vec{x}, \vec{X}$). The functions g and h are translated into $g^\flat$ and $h^\flat(y,z)$, respectively. We show that $f^\flat(y)$ can be obtained from $g^\flat$ and $h^\flat$ using 4-BRN; here we only need to define $f^\flat(y)$ for $y \le |a|$, for some a. Thus $f^\flat(y) = f'(|y|)$, where $f'(y)$ is obtained from g and $h'(y,z) = h(|y|,z)$ as follows:

$$f'(0) = g, \qquad f'(s_0(y)) = f'(s_1(y)) = h'(y, f'(y))$$

Now suppose that the translations $\varphi^\flat(z,\vec{x},\vec{y})$ and $t^\flat(\vec{x},\vec{y})$ have been obtained, for a $\mathbf{\Sigma}_0^B(\mathcal{L}_{\mathbf{FALV}})$ formula $\varphi(z,\vec{x},\vec{X})$ and an $\mathcal{L}_A^2$ term $t(\vec{x},\vec{X})$. We show how to obtain $F^\flat_{\varphi,t}(\vec{x},\vec{y})$ and $f^\flat_{\varphi,t}(\vec{x},\vec{y})$.

The function f is said to be obtained from g and h by sharply bounded minimization if

$$f(x) = \begin{cases} i & \text{if } i < |g(x)| \wedge h(i,x) = 0 \wedge \forall j < i, h(j,x) \neq 0 \\ |g(x)| & \text{otherwise} \end{cases}$$

It can be shown (using only CRN and composition and some initial functions of **QALV**) that the functions in **QALV** are closed under taking sharply bounded minimization (see the remark after (5.4) and also [Clo93, Lemma 6]). This shows that $f^\flat_{\varphi,t}(\vec{x},\vec{y})$ can be obtained from the functions in $\varphi^\flat(z,\vec{x},\vec{y})$ and initial functions of **QALV** by CRN and composition.

The characteristic function $c_\varphi(\vec{x})$ of a formula $\varphi(\vec{x})$ is defined as follows

$$c(\vec{x}) = \begin{cases} 1 & \text{if } \varphi(\vec{x}) \\ 0 & \text{otherwise} \end{cases}$$

Using sharply bounded minimization, it can be shown that the characteristic function c_φ for any sharply bounded formula φ of **QALV** is also in **QALV**. Hence $c_{\varphi^\flat}(z,\vec{x},\vec{y})$ is in **QALV**. We leave it to the reader to verify that the function $F^\flat_{\varphi,t}(\vec{x},\vec{y})$ can be obtained from $c_{\varphi^\flat}(z,\vec{x},\vec{y})$ and $t^\flat(\vec{x},\vec{y})$ using composition, CRN and initial functions of **QALV**.

5.3 VALV is Equivalent to VNC^1

The fact that $\mathbf{QALV} \overset{\text{RSUV}}{\simeq} \mathbf{VNC}^1$ follows from Theorems 5.4 and 5.5.

Theorem 5.5. **VALV** *is a conservative extension of* $\mathbf{VNC}^1$.

Proof of Conservativity. The language $\mathcal{L}_{\mathbf{FALV}}$ can be seen as being constructed in stages from $\mathcal{L}_0 = \mathcal{L}_{\mathbf{FAC}^0}$ using (2) and (3) in Definition 5.1. We will apply Corollary 3.17

for $\mathcal{T}_i = \mathbf{VNC}^1(\mathcal{L}_i) + \mathbf{\Sigma}_0^B(\mathcal{L}_i)$-$\mathbf{COMP}$. Note that $\mathcal{T}_0 = \mathbf{VNC}^1 + \overline{\mathbf{V}}^0$ is a conservative extension of $\mathbf{VNC}^1$, and the hypothesis of Corollary 3.17 applies to $\mathcal{T}_0$ (i.e., $\langle \mathcal{T}_0, \mathcal{L}_0, \mathcal{L}_A^2 \rangle$ satisfies (3.19)). Therefore it suffices to show that for each new function F (or f) in $\mathcal{L}_{n+1}$, F (or f) is provably total in $\mathcal{T}_n$ and (3.14) (resp. (3.15)) holds.

The case where the new function in $\mathcal{L}_{n+1}$ is of the form $F_{\varphi,t}$ or $f_{\varphi,t}$ follows from Lemma 3.14. So suppose that the new function f in $\mathcal{L}_{n+1}$ is of the form $f_{g,h}$ where $g, h \in \mathcal{L}_n$. Write $h_y(z)$ for $h(y,z)$. Then

$$f(y) = h_{y-1} \circ \ldots \circ h_1 \circ h_0(g)$$

The composition $h_{y-1} \circ \ldots \circ h_1 \circ h_0$ is computed by a balanced binary tree with leaves labelled by $h_0, h_1, \ldots, h_{y-1}$. The tree is depicted in Figure 5.1 and is constructed using Theorem 3.25.

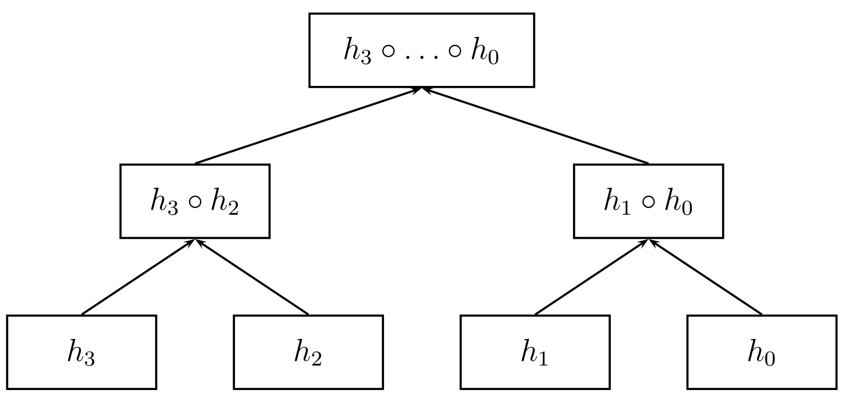

Figure 5.1: Computing f by a binary tree.

The fact that the new function f defined as above satisfies the defining axioms (5.2) and (5.3) of $f_{g,h}$ is proved in $\mathcal{T}_n$ by proving by induction (on the height of the subtree) that each subtree with leaves $h_i, h_{i+1}, \ldots, h_j$ computes the composition $h_i \circ h_{i+1} \circ \ldots \circ h_j$.

To show that $f^\star$ is also provably total in $\mathcal{T}_n$, we need polynomially many trees similar to the binary tree we used for computing f above. Again, the existence of these trees is provable in $\mathbf{VNC}^1$ using Theorem 3.25. The fact that $\mathcal{T}_n(f, f^\star)$ proves (3.11) follows from the construction of the circuits. $\square$

In the remainder of this chapter we show that **VALV** extends $\mathbf{VNC}^1$. First, it follows from Theorem 5.3 that **VALV** extends $\mathbf{V}^0$, so it remains to show that **VALV** proves MFV (Definition 3.21). To prove the existence of Y in MFV, we formalize Barrington's proof [Bar89] that the Boolean Sentence Value Problem is reducible to the word problem for S_5. Given a Boolean sentence (represented as a balanced binary tree), we construct a S_5 word whose value determines the truth value of the sentence. Once this has been done, the string Y can be obtained by $\mathbf{\Sigma}_0^B$-**COMP**.

First we outline the reduction.

5.3.1 The Reduction to the Word Problem for S_5

Notation Let $\sigma_1 = (12345), \sigma_2 = (13542)$ and $\sigma = \sigma_1^{-1} \circ \sigma_2^{-1} \circ \sigma_1 \circ \sigma_2 = (12534)$. Also, let e be the identity in S_5.

Consider a tree-like circuit T of depth $\log(a)$, with inputs $I(0), \ldots, I(a-1)$. The goal is to (uniformly) construct for each gate x in T a sequence P_x of permutations in S_5:

$$P_x = p_{x,0}, p_{x,1}, \ldots, p_{x,k-1}$$

where k depends on x (see below), so that

$$\circ P_x = p_{x,0} \circ p_{x,1} \circ \ldots \circ p_{x,k-1} = \begin{cases} \sigma & \text{if } T(x) = 1 \\ e & \text{if } T(x) = 0 \end{cases} \tag{5.6}$$

($\circ$ is the composition operator). Here P_x has length $k = k(x) = 4^h$, where h is the height (i.e., longest distance to a leaf) of the gate x in T.

The sequence P_x is defined inductively based on the height of gate x. We use the fact that σ in Notation above is a nontrivial commutator of S_5. Consider the following cases:

Case I: Gate x of T is an input gate. Then $k(x) = 1$, and

$$p_{x,0} = \begin{cases} \sigma & \text{if } I(x) = 1 \\ e & \text{if } I(x) = 0 \end{cases}$$

Case II: Gate x is an $\wedge$-gate with inputs from gates y, z. Then P_x is of the form

$$P_x = P'_y, P'_z, P''_y, P''_z$$

where P'_y, P''_y, P'_z, P''_z are obtained from P_y, P_z (see below) so that

$$\circ P'_y = \begin{cases} \sigma_1^{-1} & \text{if } T(y) = 1 \\ e & \text{if } T(y) = 0 \end{cases} \qquad \circ P''_y = \begin{cases} \sigma_1 & \text{if } T(y) = 1 \\ e & \text{if } T(y) = 0 \end{cases}$$

$$\circ P'_z = \begin{cases} \sigma_2^{-1} & \text{if } T(z) = 1 \\ e & \text{if } T(z) = 0 \end{cases} \qquad \circ P''_z = \begin{cases} \sigma_2 & \text{if } T(z) = 1 \\ e & \text{if } T(z) = 0 \end{cases}$$

Notation $\theta_1 = (14532), \theta_2 = (13425), \eta_1 = (13254), \eta_2 = (12543)$. Note that

$$\theta_i \circ \sigma \circ \theta_i^{-1} = \sigma_i, \qquad \eta_i \circ \sigma \circ \eta_i^{-1} = \sigma_i^{-1}$$

The sequences P'_y, P''_y both have length $k(y)$, and P'_z, P''_z both have length $k(z)$. They are obtained from P_y and P_z as follows:

$$p'_{y,i} = \eta_1 \circ p_{y,i} \circ \eta_1^{-1}, \qquad p''_{y,i} = \theta_1 \circ p_{y,i} \circ \theta_1^{-1} \qquad (0 \le i \le k(y) - 1) \tag{5.7}$$

$$p'_{z,i} = \eta_2 \circ p_{z,i} \circ \eta_2^{-1}, \qquad p''_{z,i} = \theta_2 \circ p_{z,i} \circ \theta_2^{-1} \qquad (0 \le i \le k(z) - 1) \tag{5.8}$$

Case III: Gate x of T is an $\vee$-gate with inputs from gates y and z. Essentially, this case reduces to the previous case using the identity:

$$A \vee B \Leftrightarrow \neg(\neg A \wedge \neg B)$$

So first we will construct sequences Q'_y, Q''_y and Q'_z, Q''_z so that the sequence

$$Q = Q'_y, Q'_z, Q''_y, Q''_z$$

satisfies

$$\circ Q = \begin{cases} e & \text{if } T(x) = 1 \\ \sigma^{-1} & \text{if } T(x) = 0 \end{cases}$$

Then the sequence P_x is defined to be the same as Q except for the last permutation q is replaced by $q \circ \sigma$. It is easy to verify that P satisfies (5.6).

Note that $\sigma^{-1} = \sigma_2^{-1} \circ \sigma_1^{-1} \circ \sigma_2 \circ \sigma_1$. We want the sequences Q'_y, Q''_y and Q'_z, Q''_z so that

$$\circ Q'_y = \begin{cases} e & \text{if } T(y) = 1 \\ \sigma_2^{-1} & \text{if } T(y) = 0 \end{cases} \qquad \circ\, Q''_y = \begin{cases} e & \text{if } T(y) = 1 \\ \sigma_2 & \text{if } T(y) = 0 \end{cases}$$

$$\circ Q'_z = \begin{cases} e & \text{if } T(z) = 1 \\ \sigma_1^{-1} & \text{if } T(z) = 0 \end{cases} \qquad \circ\, Q''_z = \begin{cases} e & \text{if } T(z) = 1 \\ \sigma_1 & \text{if } T(z) = 0 \end{cases}$$

The elements of Q'_y, Q''_y, Q'_z, Q''_z are defined as follows:

$$q'_{y,i} = \theta_2 \circ p_{y,i} \circ \theta_2^{-1}, \qquad q''_{y,i} = \eta_2 \circ p_{y,i} \circ \eta_2^{-1} \qquad (\text{for } 0 \le i \le k(y) - 2) \tag{5.9}$$

$$q'_{z,i} = \theta_1 \circ p_{z,i} \circ \theta_1^{-1}, \qquad q''_{z,i} = \eta_1 \circ p_{z,i} \circ \eta_1^{-1} \qquad (\text{for } 0 \le i \le k(z) - 2) \tag{5.10}$$

$$q'_{y,k(y)-1} = \theta_2 \circ p_{y,k(y)-1} \circ \theta_2^{-1} \circ \sigma_2^{-1}, \qquad q''_{y,k(y)-1} = \eta_2 \circ p_{y,k(y)-1} \circ \eta_2^{-1} \circ \sigma_2 \tag{5.11}$$

$$q'_{z,k(z)-1} = \theta_1 \circ p_{z,k(z)-1} \circ \theta_1^{-1} \circ \sigma_1^{-1}, \qquad q''_{z,k(z)-1} = \eta_1 \circ p_{z,k(z)-1} \circ \eta_1^{-1} \circ \sigma_1 \tag{5.12}$$

5.3.2 Nonsolvability of S_5

Before formalizing the above reduction, we analyze how the fact that S_5 is nonsolvable is used. (In general, Barrington shows that the word problem for any nonsolvable group is complete for $\mathbf{NC}^1$.)

What is needed is the existence of distinct elements σ_1, σ_2 of the group S_5 so that they both are conjugates of their commutator σ. We will show that this implies the nonsolvability of S_5.

Lemma 5.6. *Suppose that G is a group that contains two elements σ_1, σ_2 with the property that that σ_i is a conjugate of $\sigma = \sigma_1^{-1} \circ \sigma_2^{-1} \circ \sigma_1 \circ \sigma_2$, for $i = 1, 2$. Then G is nonsolvable.*

Proof. Let $H = \langle \sigma_1, \sigma_2 \rangle$ (the group generated by σ_1 and σ_2). We show that H is a nonsolvable group. It follows that G is nonsolvable, since G contains a nonsolvable subgroup.

Let K be the commutator subgroup of H. Then consider the quotient map $q : H \to H/K$. Since $\sigma \in K$, $q(\sigma) = 1$. Also, since σ_i are conjugates of σ, $q(\sigma_i) = 1$, for $i = 1, 2$. Thus $H = K$, and hence H is nonsolvable. $\square$

5.3.3 Formalizing the Proof of Barrington's Theorem

For simplicity, assume $a = 2^d$ for $d \geq 1$. Consider a gate x of height $h \geq 0$, then $2^{d-h} \leq x < 2^{d-h+1}$, and the S_5-word P_x has length 4^h. We will show how to compute the i-th permutation $p_{x,i}$ in P_x, for $0 \leq i < 4^h$.

Let $y = 2x, z = 2x + 1$ (the outputs of gates y, z are connected to the inputs of gate x). For $i < 4^h$, write i in base 4:

$$i = i_{h-1} \dots i_0, \text{ where } 0 \leq i_r \leq 3. \tag{5.13}$$

Bit i_{h-1} states which of the "quarters" P'_y, P'_z, P''_y, P''_z (or Q'_y, Q'_z, Q''_y, Q''_z) that $p_{x,i}$ comes from. For example, suppose that gate x is an $\wedge$-gate. Then for $i < 4^{h-1}$ (i.e, $i_{h-1} = 0$), using (5.7) we have

$$p_{x,i} = \eta_1 \circ p_{y,i'} \circ \eta_1^{-1}, \qquad \text{where } i' = i_{h-2} \dots i_0 \text{ (base 4)}$$

In other words, $p_{x,i}$ is defined from $p_{2x+(i_{h-1} \mod 2),i'}$ using (5.7)–(5.8) and (5.9)–(5.12).

In general, the base 4 representation (5.13) of i fully describes a path from the input gate $2^h x + k$ to gate x, where k is the number with the binary representation

$$(i_{h-1} \mod 2)(i_{h-2} \mod 2) \dots (i_0 \mod 2) \tag{5.14}$$

The ℓ-th gate on this path is the gate numbered $2^{h-\ell} x + j$, where j is the number with binary representation

$$(i_{h-1} \mod 2) \dots (i_\ell \mod 2) \tag{5.15}$$

(when $\ell = h$, $j = 0$).

The sequence P_x can be seen as being constructed in h stages: In each stage we have a sequence of length 4^h whose i-th element (a 5-permutation) is obtained from the i-th element of the previous sequence. The ℓ-th sequence will be encoded by $f_{\ell,x,i}(u)$ for $0 \le i < 4^h$. Thus we will define a function $f(\ell, x, i, u)$ so that

$$p_{x,i}(u) = f(h, x, i, u) \qquad \text{for } i < 4^h$$

We will write $f(\ell, x, i, \cdot)$ for the permutation $f_{\ell,x,i}(u) = f(\ell, x, i, u)$.

Recall (Section 3.4) that the value of the leaf gate x is $I(x - a)$. First, for i as in (5.13),

$$f(0, x, i, \cdot) = \begin{cases} \sigma & \text{if } I(2^h x + k - a) = 1 \\ e & \text{otherwise} \end{cases}$$

where k is the number with binary representation (5.14).

Next, for $1 \le \ell \le h$, $f(\ell, x, i, \cdot)$ is defined from $f(\ell - 1, x, i, \cdot)$ by cases, depending on the type of gate $(2^{h-\ell}x + j)$, where j is the number with binary representation (5.15). (note that when $\ell = h$, $j = 0$).

For example, suppose that gate $(2^{h-\ell}x + j)$ is an $\wedge$-gate. Then (following (5.7) and (5.8)):

$$f(\ell, x, i, \cdot) = \begin{cases} \eta_1 \circ f(\ell - 1, x, i, \cdot) \circ \eta_1^{-1} & \text{if } i_{\ell-1} = 0 \\ \eta_2 \circ f(\ell - 1, x, i, \cdot) \circ \eta_2^{-1} & \text{if } i_{\ell-1} = 1 \\ \theta_1 \circ f(\ell - 1, x, i, \cdot) \circ \theta_1^{-1} & \text{if } i_{\ell-1} = 2 \\ \theta_1 \circ f(\ell - 1, x, i, \cdot) \circ \theta_1^{-1} & \text{if } i_{\ell-1} = 3 \end{cases}$$

Since $\theta_1, \theta_2, \eta_1, \eta_2$ are 5-permutations, it is clear that f is defined using 5-BNR.

Finally, the value of gate x is determined by the composition $p_{x,0} \circ \ldots \circ p_{x,k(x)-1}$, i.e.,

$$f(h, x, 0, \cdot) \circ f(h, x, 1, \cdot) \circ \ldots \circ f(h, x, 4^h - 1, \cdot) \tag{5.16}$$

To compute this, define $g(h, x, i, k, \cdot)$ using 5-BNR from f as follows:

$$g(h, x, i, 0, \cdot) = f(h, x, i, \cdot)$$
$$g(h, x, i, j+1, \cdot) = g(h, x, i, j, \cdot) \circ f(h, x, i+j+1, \cdot)$$

Then

$$g(h, x, i, j, \cdot) = f(h, x, i, \cdot) \circ \ldots \circ f(h, x, i+j, \cdot) \tag{5.17}$$

Hence, (5.16) is just $g(h, x, 0, 4^h - 1, \cdot)$. As a result, $T(x)$ is 1 if and only if $g(h, x, 0, 4^h - 1, \cdot) = \sigma$.

Our definitions of f, g above show that they are in $\mathcal{L}_{\mathbf{FALV}}$.

Since **VALV** extends $\mathbf{V}^0$ (see the discussion after the Proof of Conservativity on page 76), to show that **VALV** extends $\mathbf{VNC}^1$ it remains to prove the following theorem (recall MFV from Definition 3.21):

Theorem 5.7. $\mathbf{VALV} \vdash MFV$.

Proof. Let Y be defined (using $\mathbf{\Sigma}_0^B(\mathcal{L}_{\mathbf{FALV}})$-**COMP**) as:

$$|Y| \leq 2a \ \wedge \ \forall x < 2a, Y(x) \leftrightarrow g(h, x, 0, 4^h - 1, \cdot) = \sigma$$

We will show that $\mathbf{VALV} \vdash \delta_{MFV}(a, G, I, Y)$.

Following the formalization describe above, we will prove by induction on ℓ that the sequence constructed in stage ℓ works as expected. More precisely, using (5.17), we will prove by induction on ℓ that for

$$i = i_{h-\ell-1} \ldots i_0 \text{ (base 4) and } k = (i_{h-\ell-1} \mod 2) \ldots (i_0 \mod 2) (\text{ base 2}),$$

$$g(\ell, x, i4^\ell, 4^\ell - 1, \cdot) = \begin{cases} \sigma & \text{if } Y(2^{h-\ell}x + k) = 1 \\ e & \text{otherwise} \end{cases} \tag{5.18}$$

(When $\ell = h$, $k = i = 0$ and we have $g(h, x, 0, 4^h - 1, \cdot) = \sigma$ iff $Y(x)$.)

The base case is obvious from the definition of f and Y. For the induction step, suppose that (5.18) holds for $(\ell - 1)$, where $1 \le \ell \le h$. We prove (5.18) for ℓ.

Consider the case where gate $(2^{h-\ell}x + k)$ is an $\wedge$-gate. We need to verify that $g(\ell, x, i4^\ell, 4^{\ell-1} - 1, \cdot)$, $g(\ell, x, i4^\ell + 4^{\ell-1}, 4^{\ell-1} - 1, \cdot)$, $g(\ell, x, i4^\ell + 2 \times 4^{\ell-1}, 4^{\ell-1} - 1, \cdot)$ and $g(\ell, x, i4^\ell + 3 \times 4^{\ell-1}, 4^{\ell-1} - 1, \cdot)$ respectively compute the compositions of P'_y, P'_z, P''_y and P''_z as in **Case II** in Section 5.3.1 (see (5.7) and (5.8). This can be verified by proving by induction on $j < 4^{\ell-1} - 1$ that

$$g(\ell, x, i4^\ell, j + 1, \cdot) = \eta_1 \circ g(\ell, x, i4^\ell, j, \cdot) \circ \eta_1^{-1}$$

(and $g(\ell, x, i4^\ell + 4^{\ell-1}, j + 1, \cdot) = \eta_2 \circ g(\ell, x, i4^\ell + 4^{\ell-1}, j, \cdot) \circ \eta_2^{-1}$, etc.)

Other cases are handled similarly. $\square$

Chapter 6

Theories for Relativized Classes

In Section 6.1 we give new definitions of the relativizations of $\mathbf{NC}^k$, $\mathbf{L}$ and $\mathbf{NL}$ and show that they preserve the non-relativized inclusions. We also show that separating the relativizations of any two classes in $\mathbf{AC}^0(m), \mathbf{TC}^0, \mathbf{NC}^1, \mathbf{L}, \mathbf{NL}$ is as hard as separating the nonrelativized classes themselves. The relativized theories are given in Section 6.2. The materials of this chapter are from [ACN07].

6.1 Relativizing Subclasses of P

Recall the definitions of complexity classes from Section 2.2. To relativize a circuit class where the gates have unbounded fanin, we simply allow the circuit to have unbounded fanin oracle gates:

Definition 6.1 ($\mathbf{AC}^k(\alpha)$, $\mathbf{AC}^0(m,\alpha)$, $\mathbf{TC}^0(\alpha)$)**.** *For $k \geq 0$, the class $\mathbf{AC}^k(\alpha)$ (resp. $\mathbf{AC}^0(m,\alpha)$, $\mathbf{TC}^0(\alpha)$) is defined as uniform $\mathbf{AC}^k$ (resp. $\mathbf{AC}^0(m)$, $\mathbf{TC}^0$) except that unbounded fan-in α gates are allowed.*

Defining $\mathbf{NC}^k(\alpha)$ is more complicated. In [Coo85] the depth of an oracle gate with m inputs is defined to be $\log(m)$. A circuit is an $\mathbf{NC}^k(\alpha)$-circuit provided that it has polynomial size and the total depth of all gates along any path from the output gate to an input gate is $\mathcal{O}((\log n)^k)$. Note that if there is a mix of large and small oracle gates, the number of oracle gates may not be $\mathcal{O}((\log n)^{k-1})$. Here we restrict the definition further, requiring that the nested depth of oracle gates is $\mathcal{O}((\log n)^{k-1})$.

Definition 6.2 ($\mathbf{NC}^k(\alpha)$)**.** *For $k \geq 1$, a language is in $\mathbf{NC}^k(\alpha)$ if it is computable by a uniform family of $\mathbf{NC}^k(\alpha)$ circuits, i.e., $\mathbf{AC}^k(\alpha)$ circuits where the $\wedge$ and $\vee$ gates have fanin 2, and the nested depth of α gates is $\mathcal{O}((\log n)^{k-1})$.*

The following inclusions extend the inclusions of the nonrelativized classes:

$$\mathbf{AC}^0(\alpha) \subsetneq \mathbf{AC}^0(2,\alpha) \subsetneq \mathbf{AC}^0(6,\alpha) \subseteq \mathbf{TC}^0(\alpha) \subseteq \mathbf{NC}^1(\alpha) \subseteq \mathbf{AC}^1(\alpha) \subseteq \ldots$$

To define oracle logspace classes, we use a modification of Wilson's stack model [Wil88]. An advantage is that the relativized classes defined here are closed under $\mathbf{AC}^0$-reductions. This is not true for the non-stack model.

A Turing machine M with a stack of oracle tapes can write 0 or 1 onto the top oracle tape if it already contains some symbols, or it can start writing on an empty oracle tape. In the latter case, the new oracle tape will be at the top of the stack, and we say that M performs a *push* operation. The machine can also *pop* the stack, and its next action and state depend on $\alpha(Q)$, where Q is the content of the top oracle tape. Note that here the oracle tapes are write-only.

Instead of allowing an arbitrary number of oracle tapes, we modify Wilson's model by allowing only a stack of constant height. This places the relativized classes in the same order as the order of their unrelativized counterparts. In the definition of $\mathbf{NL}(\alpha)$, we also use the restriction [RST84] that the machine is deterministic when the oracle stack is non empty.

Definition 6.3 ($\mathbf{L}(\alpha)$, $\mathbf{NL}(\alpha)$)**.** *For a unary relation α on strings, $\mathbf{L}(\alpha)$ is the class of languages computable by logspace, polytime Turing machines using an α-oracle stack whose height is bounded by a constant. $\mathbf{NL}(\alpha)$ is defined as $\mathbf{L}(\alpha)$ but the Turing machines are allowed to be nondeterministic when the oracle stack is empty.*

Theorem 6.4. $\mathbf{NC}^1(\alpha) \subseteq \mathbf{L}(\alpha) \subseteq \mathbf{NL}(\alpha) \subseteq \mathbf{AC}^1(\alpha)$.

Proof. The second inclusion is immediate from the definitions, the first can be proved as in the standard proof of the fact that $\mathbf{NC}^1 \subseteq \mathbf{L}$ (see also [Wil88]). The last inclusion can actually be strengthened, as shown in the next theorem. □

Theorem 6.5. *Each language in* $\mathbf{NL}(\alpha)$ *can be computed by a uniform family of* $\mathbf{AC}^1(\alpha)$ *circuits whose nested depth of oracle gates is bounded by a constant.*

Proof. We proceed as in the proof of the fact that $\mathbf{NL} \subseteq \mathbf{AC}^1$. Let M be a nondeterministic logspace Turing machine with a constant-height stack of oracle tapes. Let h be the bound on the height of the oracle stack. There is a polynomial $p(n)$ so that for each input length n and oracle α, M has at most $p(n)$ possible configurations:

$$u_0 = START,\ u_1 = ACCEPT,\ u_2, \ldots, u_{p(n)-1} \tag{6.1}$$

(Here a configuration u_i encodes information about the state, the content of the work tape, the position of the input tape head and the input symbol being read, but no information about the oracle stack.)

Given an input of length n, consider the directed graph G with vertices (k, u_i) for $0 \leq k \leq h$, $0 \leq i < p(n)$, where the edge relation E is as follows: For u_j a next configuration of u_i,

(i) if M does not *push* or *pop* after u_i, then $((0, u_i), (0, u_j)) \in E$; if furthermore u_i codes a deterministic state, then $((k, u_i), (k, u_j)) \in E$, for $1 \leq k \leq h$;

(ii) if the next move of M after u_i is *push*, then $((k, u_i), (k+1, u_j)) \in E$ for $0 \leq k < h$;

(iii) otherwise, if the move of M after u_i is *pop*, then $((k, u_i), (k-1, u_j)) \in E$ for $1 \leq k \leq h$.

(Here k is a possible height of the stack when M has configuration u_i.)

Suppose that edge relation E has been computed, then the Reachability relation in G can be computed by an $\mathbf{AC}^1$ circuit. M accepts if and only if $(0, ACCEPT)$ is reachable from $(0, START)$. It remains to show that E can be computed by an $\mathbf{AC}^1(\alpha)$ circuit.

Let E_k denote the subgraph of E that contains the edges in (i,ii), and the edges $((\ell, u_i), (\ell - 1, u_j))$ as in (iii) where $k \leq \ell \leq h$. (Thus $E_1 = E$.) Also, let E_{h+1} denote the subgraph of E that contains only the edges as in (i,ii).

Note that E_{h+1} can be computed by an $\mathbf{AC}^0$ circuit. We show that E_k can be computed from E_{k+1} by an $\mathbf{AC}^1(\alpha)$ circuit whose oracle depth is one (for $1 \leq k \leq h$). This will complete our proof of the theorem.

The new edges in E_k are of the form $((k, u_i), (k - 1, u_j))$ where u_j is resulted from u_i by a *pop* operation. To check whether u_i, u_j satisfy this condition, we need to compute the oracle query on the current oracle tape that is asked when M moves from u_i to u_j. This query is computed by tracing back the computation of M, starting at u_i, until we hit the first configuration v where the oracle stack height is $k - 1$. More precisely, we compute the path in E_{k+1} of the form

$$(k - 1, v), (k, v_0), (k_1, v_1), \ldots, (k_t, v_t), (k, u_i)$$

where $k \leq k_1, \ldots, k_t \leq h$. This path can be computed by a deterministic logspace function, and hence an $\mathbf{AC}^1$ circuit.

Now, the oracle query Q asked at u_i can be extracted from the sequence

$$(v, v_0, v_1, \ldots, v_t)$$

by an $\mathbf{AC}^0$ circuit. Then, $((k, u_i), (k - 1, u_j)) \in E_k$ if and only if $\alpha(Q)$. $\square$

We now consider the relativization of the following classes:

$$\{\mathbf{AC}^0(m), \mathbf{TC}^0, \mathbf{NC}^1, \mathbf{L}, \mathbf{NL}\} \tag{6.2}$$

Recall that each class $\mathbf{C}$ in (6.2) is the $\mathbf{AC}^0$ closure of a function $F_{\mathbf{C}}$: $F_{\mathbf{TC}^0} = numones$,

$F_{\mathbf{AC}^0(m)} = mod_m$, $F_{\mathbf{NC}^1} = Fval$, $F_{\mathbf{L}} = SinglePath$, and $F_{\mathbf{NL}} = Conn$. (See Chapter 3, Propositions 3.2, 3.19, 3.22, 3.29, 3.32.)

Theorem 6.6. *For each class* $\mathbf{C}$ *in* (6.2), $\mathbf{C}(\alpha)$ *(resp.* $\mathbf{FC}(\alpha)$*) is the class of relations (resp. functions)* $\mathbf{AC}^0$*-reducible to* $\{F_{\mathbf{C}}, \alpha\}$.

Proof. For the classes $\mathbf{TC}^0(\alpha), \mathbf{AC}^0(m, \alpha), \mathbf{NC}^1(\alpha)$ this is immediate from the definitions involved. For the classes $\mathbf{L}(\alpha), \mathbf{NL}(\alpha)$ we show they are $\mathbf{AC}^0$-reducible to their corresponding path problem and α using ideas in the proof of Theorem 6.5. (The $\mathbf{AC}^1(\alpha)$ circuit that computes E_k from E_{k+1} can be replaced by an $\mathbf{AC}^0(\alpha)$ circuit with gates computing *Conn*.) Conversely, to show that a relation that is $\mathbf{AC}^0$-reducible to the path problem and α is in the corresponding class $\mathbf{L}(\alpha)$ or $\mathbf{NL}(\alpha)$, the Turing machine performs a depth-first search of the constant-depth reducing circuit. Each α query is answered using the constant-height oracle stack, and each path query is answered by simulating the log-space Turing machine that solves that query, where each input bit of the query must be recomputed each time it is needed in the computation. □

The following corollary generalizes results in [Wil89]:

Corollary 6.7. *For any* $\mathbf{C}_1, \mathbf{C}_2$ *in* (6.2), $\mathbf{C}_1 = \mathbf{C}_2$ *if and only if for all* α, $\mathbf{C}_1(\alpha) = \mathbf{C}_2(\alpha)$.

On the other hand, it is shown [ACN07] that there is an oracle α so that

$$\mathbf{NC}^1(\alpha) \subsetneq \mathbf{NC}^2(\alpha) \subsetneq \ldots \subsetneq \mathbf{P}(\alpha)$$

6.1.1 $\mathbf{L}(\alpha)$ Reducibility

The next lemma can be used to show that Immerman-Szelepcsényi Theorem and Savitch's Theorems relativize. Recall that **STCONN** is the problem of given (G, s, t), where s, t are two designated vertices of a directed graph G, decide whether there is a path from s to t.

Lemma 6.8. *A language is in* $\mathbf{NL}(\alpha)$ *iff it is many-one* $\mathbf{L}(\alpha)$ *reducible to* **STCONN**.

Proof. The IF direction is easy, so we prove the ONLY IF direction. Let $\mathcal{L}$ be a language in $\mathbf{NL}(\alpha)$ which is computed by M, an $\mathbf{NL}$ machine with a constant height oracle stack. The $\mathbf{L}(\alpha)$ transformation is as follows. Given an input string X to M, the graph G has polynomially many vertices in (6.1), which are the configurations of M on input X. The edges of G are

(i) (u_i, u_j) where u_j is a next configuration of u_i, and u_i does not write on an empty stack.

(ii) (u_i, u_j) where u_i writes on an empty stack, and u_j is the next time the stack is empty.

The edges in (i) can be computed in $\mathbf{AC}^0$, while the edges in (ii) can be computed in $\mathbf{L}(\alpha)$ (because M is deterministic when the oracle stack is non-empty). □

Corollary 6.9 (Relativized Immerman-Szelepcsényi Theorem)**.** $\mathbf{NL}(\alpha)$ *is closed under complementation.*

Proof. Any language in *co*-$\mathbf{NL}(\alpha)$ is $\mathbf{L}(\alpha)$ reducible to $\overline{\mathbf{STCONN}}$, which is $\mathbf{AC}^0$ reducible to $\mathbf{STCONN}$. So *co*-$\mathbf{NL}(\alpha) \subseteq \mathbf{NL}(\alpha)$. □

Let $\mathbf{L}^2(\alpha)$ denote the class of languages computable by a deterministic oracle Turing machine in $\mathcal{O}(\log^2)$ space and constant-height oracle stack.

Corollary 6.10 (Relativized Savitch's Theorem)**.** $\mathbf{NL}(\alpha) \subseteq \mathbf{L}^2(\alpha)$.

Proof. The corollary follows from Lemma 6.8 and the fact that the composition of a $\mathbf{L}(\alpha)$ function and a $(\log^2)$ space function (for $\mathbf{STCONN}$) is a $\mathbf{L}^2(\alpha)$ function. □

6.2 Relativizing the Theories

Recall the function *Row* from Definition 2.20.

Notation For a predicate α, let $\mathbf{\Sigma}_0^B(\alpha)$ denote the class of $\mathbf{\Sigma}_0^B$ formulas in $\mathcal{L}_A^2 \cup \{Row, \alpha\}$.

Definition 6.11. $\mathbf{V}^0(\alpha) = \mathbf{V}^0 + \mathbf{\Sigma}_0^B(\alpha)$-**COMP**. *For each class* $\mathbf{C}$ *in* (6.2), *the theory* $\mathbf{VC}(\alpha)$ *is defined as* $\mathbf{VC}$ *(Definitions 3.3, 3.20, 3.21, 3.30, 3.33) with* $\mathbf{\Sigma}_0^B$-**COMP** *replaced by* $\mathbf{\Sigma}_0^B(\alpha)$-**COMP**.

Notice that a natural relativized version of the additional axioms of $\mathbf{VC}$, such as *CONN* (Definition 3.30), are already provable in $\mathbf{VC}(\alpha)$. For example, let $CONN(\alpha)$ be the axiom scheme

$$\forall a \exists Y,\ Y(0,0) \wedge \forall x < a(x \neq 0 \supset \neg Y(0,x)) \wedge$$
$$\forall z < a \forall x < a,\ Y(z+1,x) \leftrightarrow (Y(z,x) \vee \exists y < a,\ Y(z,y) \wedge \varphi(y,x)).$$

where φ is a $\mathbf{\Sigma}_0^B(\alpha)$ formula. Then it is easy to use $\mathbf{\Sigma}_0^B(\alpha)$-**COMP** to show that $\mathbf{VNL}(\alpha) \vdash CONN(\alpha)$.

Theorem 6.12. *For a class* $\mathbf{C}$ *in* $\{\mathbf{AC}^0, \mathbf{AC}^0(m), \mathbf{TC}^0, \mathbf{NC}^1, \mathbf{L}, \mathbf{NL}\}$, *a function is in* $\mathbf{FC}(\alpha)$ *if and only if it is* $\mathbf{\Sigma}_1^1(\alpha)$ *definable in* $\mathbf{VC}(\alpha)$.

Proof. The theorem follows from Corollary 3.17 (for $\mathcal{L}' = \mathcal{L}_A^2 \cup \{Row, \alpha\}$) and the fact that for each class $\mathbf{C}$, the aggregate function $F_{\mathbf{C}}^\star$ (see $F_{\mathbf{C}}$ in Theorem 6.6, here $F_{\mathbf{AC}^0}$ is simply a constant function) is provably total in $\mathbf{VC}$. $\square$

Now we present the theories $\mathbf{VAC}^k(\alpha)$ (for $k \geq 1$) and $\mathbf{VNC}^k(\alpha)$ (for $k \geq 2$). We use the fact that the problem of evaluating uniform $\mathbf{AC}^k(\alpha)$ (or $\mathbf{NC}^k(\alpha)$) circuits is $\mathbf{AC}^0$-complete for the corresponding relativized class. We will give a defining axiom (see (3.5)) for the function *Ocv* that evaluates a given oracle circuit. (*Ocv* stands for oracle circuit value.)

Similar to the encoding of a monotone circuit (3.35), here an oracle circuit C is encoded by (w, d, E, G). The type (i.e., $\wedge$, $\vee$, $\neg$ or α) of gate x on layer z is specified by $(G)^{\langle z,x\rangle}$. Also, since the order of inputs to an oracle gate is important, the edge relation is now encoded (by E) as follows: $E(z,t,u,x)$ indicates that gate u on layer z is the t-th input to gate x on layer $z+1$. We need

$$Proper(w,d,E) \equiv \forall z < d \forall t,x,u_1,u_2 < w,\ (E(z,t,u_1,x) \land E(z,t,u_2,x)) \supset u_1 = u_2$$

In the following formula, $Q^{[z+1,x]}$ encodes the query to the oracle gate x on layer $z+1$:

$$\begin{aligned}
&\delta^{\alpha}_{OCV}(w,d,E,G,I,Q,Y) \equiv \forall z < d \forall x < w \\
&\quad [\forall t < w (Q^{[z+1,x]}(t) \leftrightarrow (\exists u < w,\ E(z,t,u,x) \land Y(z,u)))] \ \land \ [Y(0,x) \leftrightarrow I(x)] \land \\
&\qquad [Y(z+1,x) \leftrightarrow (((G)^{\langle z+1,x\rangle} = \text{“}\land\text{”} \land \forall t,u < w,\ E(z,t,u,x) \supset Y(z,u)) \lor \\
&\qquad\qquad ((G)^{\langle z+1,x\rangle} = \text{“}\lor\text{”} \land \exists t,u < w,\ E(z,t,u,x) \land Y(z,u)) \lor \\
&\qquad\qquad ((G)^{\langle z+1,x\rangle} = \text{“}\neg\text{”} \land \forall u < w,\ E(z,0,u,x) \supset \neg Y(z,u)) \lor \\
&\qquad\qquad\qquad ((G)^{\langle z+1,x\rangle} = \text{“}\alpha\text{”} \land \alpha(Q^{[z+1,x]}))))]
\end{aligned}$$

Definition 6.13 ($\mathbf{VAC}^k(\alpha)$)**.** *For $k \geq 1$, $\mathbf{VAC}^k(\alpha)$ is the theory over $\mathcal{L}^2_A \cup \{Row, \alpha\}$ and is axiomatized by $\mathbf{V}^0$ and the following axiom:*

$$\forall w,E,G,I(Proper(w,d,E) \supset \exists Q,Y \delta^{\alpha}_{OCV}(w,(\log w)^k,E,G,I,Q,Y))$$

To specify an $\mathbf{NC}^k(\alpha)$ circuit, we need to express the condition that $\land$ and $\lor$ gates have fanin 2. Here we use the following formula $Fanin2'(w,d,E,G)$:

$$\forall z < d \forall x < w((G)^{\langle z,x\rangle} \neq \text{“}\alpha\text{”} \supset \exists u_1,u_2 < w \forall t,v < w,\ E(z,t,v,x) \supset v = u_1 \lor v = u_2)$$

Moreover, the nested depth of oracle gates in circuit (w,d,E,G) needs to be bounded. The formula $ODepth(w,d,h,E,G,H)$ below states that this nested depth is bounded by h ($H(z,x,s)$ holds iff the nested depth of oracle gates in the subtree rooted at gate x on layer z is s):

$$\begin{gathered}
\forall z \leq d \forall x < w \exists! s \leq h\, H(z,x,s) \land \forall x < w H(0,x,0) \land \\
\forall z < d \forall x < w \exists m, m = max\{h : \exists t,u < w E(z,t,u,x) \land H(z,u,h)\} \land \\
[((G)^{\langle z+1,x\rangle} = \text{“}\alpha\text{”} \supset H(z+1,x,m+1)) \land ((G)^{\langle z+1,x\rangle} \neq \text{“}\alpha\text{”} \supset H(z+1,x,m))]
\end{gathered}$$

Definition 6.14 ($\mathbf{VNC}^k(\alpha)$)**.** *For $k \geq 2$, $\mathbf{VNC}^k(\alpha)$ is the theory over $\mathcal{L}_A^2 \cup \{Row, \alpha\}$ and is axiomatized by $\mathbf{V}^0$ and the axiom*

$$\forall w \forall E, G, I, H,\ [Proper(w, d, E) \wedge Fanin2'(w, |w|^k, E, G) \wedge$$
$$ODepth(w, d, |w|^{k-1}, E, G, H)] \ \supset\ \exists Q, Y\, \delta^{\alpha}_{OCV}(w, (\log w)^k, E, G, I, Q, Y)$$

The next theorem can be proved as Theorem 6.12.

Theorem 6.15. *For $k \geq 1$, the functions in $\mathbf{FAC}^k(\alpha)$ are precisely the provably total functions of $\mathbf{VAC}^k(\alpha)$. The same holds for $\mathbf{FNC}^k(\alpha)$ and $\mathbf{VNC}^k(\alpha)$, for $k \geq 2$.*

Chapter 7

The Discrete Jordan Curve Theorem

The Jordan Curve Theorem (JCT) asserts that a simple closed curve divides the plane into *exactly* two connected components. We consider the discrete version of the theorem where the curve lies on a grid graph. Thus a curve can be represented as a sequence of edges that form a cycle of distinct vertices, or a set of edges where each grid vertex has degree exactly 0 or 2. In the latter setting there may be multiple simple closed curves, so we can only show that there are *at least* two connected components.

In Section 7.1 we present a $\mathbf{V}^0(2)$-proof of the theorem in the second setting above. A $\mathbf{V}^0$-proof of the theorem in the first setting is given in Section 7.2. The reduction from the st-connectivity problem to JCT is shown in Section 7.3.

7.1 Input as a Set of Edges

We start by defining the notions of (grid) points and edges, and certain sets of edges which include closed curves, or connect grid points. All of these notions are definable by $\mathbf{\Sigma}_0^B$-formulas, and their basic properties can be proved in $\mathbf{V}^0$.

We assume a parameter n which bounds the x and y coordinates of points on the curve in question. Thus a *grid point* (or simply a *point*) p is a pair (x, y) which is encoded by the pairing function $\langle x, y\rangle$ (see (2.3) on page 20), where $0 \leq x, y \leq n$. The x and y

coordinates of a point p are denoted by $x(p)$ and $y(p)$ respectively. Thus if $p = \langle i, j \rangle$ then $x(p) = i$ and $y(p) = j$. An (undirected) *edge* is a pair (p_1, p_2) (represented by $\langle p_1, p_2 \rangle$) of adjacent points; i.e. either $|x(p_2) - x(p_1)| = 1$ and $y(p_2) = y(p_1)$, or $x(p_2) = x(p_1)$ and $|y(p_2) - y(p_2)| = 1$. For a horizontal edge e, we also write $y(e)$ for the (common) y-coordinate of its endpoints.

Let E be a set of edges (represented by a set of numbers representing those edges). The *E-degree* of a point p is the number of edges in E that are incident to p.

Definition 7.1. *A* curve *is a nonempty set E of edges such that the E-degree of every grid point is either 0 or 2. A set E of edges is said to* connect *two points p_1 and p_2 if the E-degrees of p_1 and p_2 are both 1 and the E-degrees of all other grid points are either 0 or 2. Two sets E_1 and E_2 of edges are said to* intersect *if there is a grid point whose E_i-degree is ≥ 1 for $i = 1, 2$.*

Note that a curve in the above sense is actually a collection of one or more disjoint closed curves. Also if E connects p_1 and p_2 then E consists of a path connecting p_1 and p_2 together with zero or more disjoint closed curves.

We also need to define the notion of two points being on different sides of a curve. We are able to consider only points which are "close" to the curve. It suffices to consider the case in which one point is above and one point is below an edge in E. (Note that the case in which one point is to the left and one point is to the right of E can be reduced to this case by rotating the $(n+1) \times (n+1)$ array of all grid points by 90 degrees.)

Definition 7.2. *Two points p_1, p_2 are said to be* on different sides *of E if (i) $x(p_1) = x(p_2) \wedge |y(p_1) - y(p_2)| = 2$, (ii) the E-degree of $p_i = 0$ for $i = 1, 2$, and (iii) the E-degree of p is 2, where p is the point with $x(p) = x(p_1)$ and $y(p) = \frac{1}{2}(y(p_1) + y(p_2))$. (See Figure 7.1.)*

Now we show that any set of edges that forms at least one simple curve must divide the plane into at least two connected components.

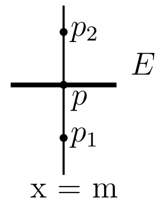

Figure 7.1: p_1, p_2 are on different sides of E.

Theorem 7.3 (Main Theorem for $\mathbf{V}^0(2)$)**.** *The theory* $\mathbf{V}^0(2)$ *proves the following: Suppose that* B *is a set of edges forming a curve,* p_1 *and* p_2 *are two points on different sides of* B*, and that* R *is a set of edges that connects* p_1 *and* p_2*. Then* B *and* R *intersect.*

7.1.1 The Proof of the Main Theorem for $\mathbf{V}^0(2)$

We will actually work with the conservative extension $\mathbf{V}^0(parity)$ of $\mathbf{V}^0(2)$ that is obtained from $\mathbf{V}^0(2)$ by adding the function *parity* and its defining axiom (3.21) on page 38. Note that $\mathbf{V}^0(parity)$ proves $\mathbf{\Sigma}_0^B(parity)$-**COMP** (see the proof of Theorem 3.10) and hence also $\mathbf{\Sigma}_0^B(parity)$-**IND** and $\mathbf{\Sigma}_0^B(parity)$-**MIN** (Theorem 2.14).

In the following discussion we also refer to the edges in B as "blue" edges, and the edges in R as "red" edges.

We argue in $\mathbf{V}^0(2)$, and prove the theorem by contradiction. Suppose to the contrary that B and R satisfy the hypotheses of the theorem, but do not intersect.

Notation A horizontal edge is said to be *on column* k (for $k \leq n-1$) if its endpoints have x-coordinates k and $k+1$.

Let $m = x(p_1) = x(p_2)$. W.l.o.g., assume that $2 \leq m \leq n-2$. Also, we may assume that the red path comes to both p_1 and p_2 from the left, i.e., the two red edges that are incident to p_1 and p_2 are both horizontal and on column $m-1$ (see Figure 7.2). (Note that if the red path does not come to both points from the left, we could fix this by effectively doubling the density of the points by doubling n to $2n$, replacing each edge in B or R by a double edge, and then extending each end of the new path by three (small) edges forming a "C" shape to end at points a distance 1 from the blue curve, approaching from the left.)

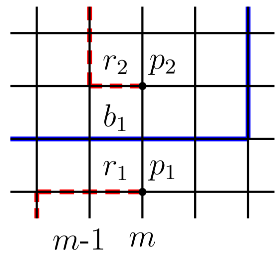

Figure 7.2: The red (dashed) path must cross the blue (undashed) curve.

We say that edge e_1 *lies below* edge e_2 if e_1 and e_2 are horizontal and in the same column and $y(e_1) < y(e_2)$. For each horizontal red edge r we consider the parity of the number of horizontal blue edges b that lie below r. The following notion is definable in $\mathbf{V}^0(2)$.

Notation An edge r is said to be an *odd edge* if it is red and horizontal and

$$parity(\{b : b \text{ is a horizontal blue edge that lies below } r\}) = 1$$

For example, it is easy to show in $\mathbf{V}^0(2)$ that exactly one of r_1, r_2 in Figure 7.2 is an odd edge.

For each $k \leq n-1$, define using $\mathbf{\Sigma}_0^B(parity)$-**COMP** the set

$$X_k = \{r : r \text{ is an odd edge in column } k\}$$

Lemma 7.4. *It is provable in* $\mathbf{V}^0(2)$ *that*

a) $parity(X_{m-1}) = 1 - parity(X_m)$.

b) *For* $0 \leq k \leq n-2$, $k \neq m$, $parity(X_k) = parity(X_{k+1})$.

Proof of the Main Theorem for $\mathbf{V}^0(2)$. We may assume that there are no edges in either B or R in columns 0 and $n-1$, so $parity(X_0) = parity(X_{n-1}) = 0$. On the other hand, it follows by $\mathbf{\Sigma}_0^B(\mathcal{L}_{\mathbf{FAC}^0}(2))$-**IND** using **b)** that $parity(X_0) = parity(X_{m-1})$ and $parity(X_m) = parity(X_{n-1})$, which contradicts **a)**. $\square$

Proof of Lemma 7.4. First we prove **b)**. For $k \leq n-1$ and $0 \leq j \leq n$, let $e_{k,j}$ be the horizontal edge on column k with y-coordinate j. Fix $k \leq n-2$. Define the ordered lists (see Figure 7.3)

$$L_0 = e_{k,0}, e_{k,1}, \ldots, e_{k,n}; \qquad L_{n+1} = e_{k+1,0}, e_{k+1,1}, \ldots, e_{k+1,n}$$

and for $1 \leq j \leq n$:

$$L_j = e_{k+1,0}, \ldots, e_{k+1,j-1}, \langle (k+1, j-1), (k+1, j) \rangle, e_{k,j}, \ldots, e_{k,n}$$

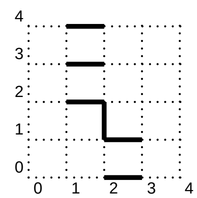

Figure 7.3: L_2 (for $n = 4, k = 1$).

A red edge r is said to be *odd in* L_j if $r \in L_j$, and

$$\mathit{parity}(\{b : b \text{ is a blue edge that precedes } r \text{ in } L_j\}) = 1$$

(In particular, X_k and X_{k+1} consist of odd edges in L_0 and L_{n+1}, respectively.) For $0 \leq j \leq n+1$, let

$$Y_j = \{r : r \text{ is an odd edge in } L_j\}$$

Thus $Y_0 = X_k$ and $Y_{n+1} = X_{k+1}$.

Claim: If $k \neq m-1$ then $\mathit{parity}(Y_j) = \mathit{parity}(Y_{j+1})$ for $j \leq n$.

This is because the symmetric difference of Y_j and Y_{j+1} has either no red edges, or two red edges with the same parity.

Thus by $\mathbf{\Sigma}_0^B(\mathcal{L}_{\mathbf{FAC}^0}(2))$-**IND** on j we have $\mathit{parity}(Y_0) = \mathit{parity}(Y_{n+1})$, and hence $\mathit{parity}(X_k) = \mathit{parity}(X_{k+1})$.

The proof of **a)** is similar. The only change here is that $\mathit{parity}(L_j)$ and $\mathit{parity}(L_{j+1})$ must differ for exactly one value of j: either $j = y(p_1)$ or $j = y(p_2)$ (because either r_1 is odd in $L_{y(p_1)}$ or r_2 is odd in $L_{y(p_2)}$, but not both). $\square$

7.2 Input as a Sequence of Edges

Now suppose that B is a sequence of edges

$$\langle q_0, q_1\rangle, \langle q_1, q_2\rangle, \ldots, \langle q_{t-2}, q_{t-1}\rangle, \langle q_{t-1}, q_0\rangle$$

that form a single closed curve (i.e, $t \geq 4$ and $q_0, \ldots, q_{t-1}$ are distinct). In this section we will show that the weak base theory $\mathbf{V}^0$ proves two theorems that together imply the Jordan Curve Theorem for grid graphs: The curve B divides the grid into exactly two connected regions. Theorem 7.5 is the analog of Theorem 7.3 (Main Theorem for $\mathbf{V}^0(2)$), and states that a sequence of edges forming a path connecting points p_1 and p_2 on different sides of the curve must intersect the curve. Theorem 7.13 states that any point p in the grid off the curve can be connected by a path (in a refined grid) that does not intersect the curve, and leads from p to one of the points p_1 or p_2.

There is no analog in Section 7.1 to the last theorem because in that setting it would be false: the definition of a *curve* as a set of edges allows multiple disjoint curves.

7.2.1 There are at Least Two Regions

Theorem 7.5 (Main Theorem for $\mathbf{V}^0$). *The theory $\mathbf{V}^0$ proves the following: Let B be a sequence of edges that form a closed curve, and let p_1, p_2 be any two points on different sides of B. Suppose that R is a sequence of edges that connect p_1, p_2. Then R and B intersect.*

(See Definition 7.1 to explain the notion of points p_1, p_2 being on different sides of a curve.)

We use the fact that the edges B can be directed (i.e., from q_i to q_{i+1}). This theorem follows easily from the Edge Alternation Theorem 7.7, which states that the horizontal edges on each column m of a closed curve must alternate between pointing right and pointing left.

Alternating edges and proof of the Main Theorem

The following notion of *alternating sets* is fundamental to the proof of the Main Theorem for $\mathbf{V}^0$. Two sets X and Y of numbers are said to alternate if their elements are interleaved, in the following sense.

Definition 7.6. *Two disjoint sets X, Y* alternate *if between every two elements of X there is an element of Y, and between every two elements of Y there is an element of X. These conditions are defined by the following $\mathbf{\Sigma}_0^B$ formulas:*

(i) $\forall x_1, x_2 \in X(x_1 < x_2 \supset \exists y \in Y, x_1 < y < x_2)$,

(ii) $\forall y_1, y_2 \in Y(y_1 < y_2 \supset \exists x \in X, y_1 < x < y_2)$

Theorem 7.7 (Edge Alternation Theorem)**.** *(Provable in $\mathbf{V}^0$) Let P be a sequence of edges that form a closed curve. For each column m, let A_m be the set of y-coordinates of left-pointing edges of P on the column, and let B_m be the set of y-coordinates of right-pointing edges of P on the column. Then A_m and B_m alternate.*

The proof of this theorem starts on page 106, after presenting necessary concepts and lemmas.

Proof of Theorem 7.5 from the Edge Alternation Theorem. The proof is by contradiction. Assume that R does not intersect B. We construct a sequence of edges P from B and R that form a closed curve, but that violate the Edge Alternation Theorem.

Without loss of generality, assume that p_1, p_2 and B, R are as in Figure 7.2. Also, suppose that the sequence R starts from p_1 and ends in p_2. We may assume that the edge b_1 is from right to left (otherwise reverse the curve). Assume that the point $\langle x(p_1)+1, y(p_1)\rangle$ is not on B or R. (This can be achieved by doubling the density of the grid.)

We merge B and R into a sequence of edges as in Figure 7.4. Let P be the resulting sequence of edges. Then P is a closed curve. However, the edges r_1 and b_1 have the same direction, and thus violate the Edge Alternation Theorem. $\square$

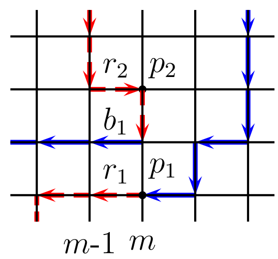

Figure 7.4: Merging the red (dashed) path and the blue (undashed) curve.

Bijections between alternating sets

Suppose that X and Y alternate and $f : X \to Y$ is a bijection from X to Y. Let $x_1, x_2 \in X$, $x_1 < x_2$, and suppose that neither $f(x_1)$ nor $f(x_2)$ lies between x_1 and x_2. Since the open interval (x_1, x_2) contains more elements of Y than X, it must contain an image $f(z)$ of some $z \in X$ where either $z < x_1$ or $z > x_2$.

The above property can be formalized and proved in $\mathbf{VTC}^0$, where f is given by its graph: a finite set of ordered pairs. However, it is not provable in $\mathbf{V}^0$, because it implies the surjective Pigeonhole Principle, which is not provable in $\mathbf{V}^0$ [CN06]. Nevertheless it is provable in $\mathbf{V}^0$ under the assumption that f satisfies the condition that connecting each x to its image $f(x)$ by an arc above the line $\mathbb{N}$ does not create any "crossings", i.e. (see Figure 7.5)

$$\text{the sets } \{z_1, f(z_1)\} \text{ and } \{z_2, f(z_2)\} \text{ are not alternating, for all } z_1, z_2 \in X,\ z_1 \neq z_2. \tag{7.1}$$

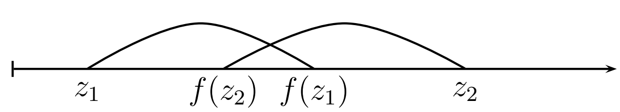

Figure 7.5: f violates (7.1)

We need the following result to prove the Edge Alternation Theorem.

Lemma 7.8 (Alternation Lemma). *(Provable in $\mathbf{V}^0$) Suppose that X and Y alternate* *and that f (given by a finite set of ordered pairs) is a bijection between X and Y that satisfies* (7.1)*. Let $x_1, x_2 \in X$ be such that $x_1 < x_2$ and neither $f(x_1)$ nor $f(x_2)$ is in the interval (x_1, x_2). Then,*

$$\exists z \in X, (z < x_1 \vee z > x_2) \wedge x_1 < f(z) < x_2 \tag{7.2}$$

Proof. We prove by contradiction, using the number minimization principle $\mathbf{\Sigma}_0^B$-**MIN**. Let x_1, x_2 be a counter example with the least difference $x_2 - x_1$.

Let $y_1 = max(\{y \in Y : y < x_2\})$. We have $x_1 < y_1 < x_2$. Let x_2' be the pre-image of y_1: $f(x_2') = y_1$. By our assumption that (7.2) is false, $x_1 < x_2' < x_2$. In addition, since $y_1 = max(\{y \in Y : y < x_2\})$ and X, Y alternate, we have $x_1 < x_2' < y_1$. (See Figure 7.6.)

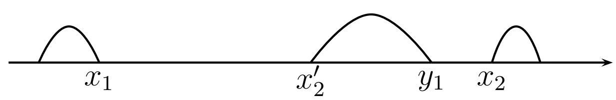

Figure 7.6: $f(x_1), f(x_2) \notin (x_1, x_2)$, and $f(x_2') = y_1$.

Now by (7.1), for all $z \in X$, $x_2' < z < y_1$ implies that $x_2' < f(z) < y_1$. Hence the pair x_1, x_2' is another counter example, and $x_2' - x_1 < x_2 - x_1$, contradicts our choice of x_1, x_2. □

Alternating endpoints of curve segments

From now on, P denotes a sequence of edges

$$\langle p_0, p_1 \rangle, \langle p_1, p_2 \rangle, \ldots, \langle p_{t-2}, p_{t-1} \rangle, \langle p_{t-1}, p_0 \rangle$$

that form a single closed curve (i.e, $t \geq 4$ and $p_0, \ldots, p_{t-1}$ are distinct).

For convenience, we assume that P has a point on the first vertical line ($x = 0$) and a point on the last vertical line ($x = n$). To avoid wrapping around the last index, we pick some vertical edge on the line ($x = n$) and define p_0 to be the forward end of this edge. In other words, the edge $\langle p_{t-1}, p_0 \rangle$ lies on the line ($x = n$).

It is easy to prove in $\mathbf{V}^0$ that for every m, $0 \le m \le n$, P must have a point on the vertical line $(x = m)$. For otherwise there is a largest $m < n$ such that the line $(x = m)$ has no point on P, and we obtain a contradiction by considering the edge $\langle p_{i-1}, p_i \rangle$, where i is the smallest number such that $x(p_i) \le m$.

For $a < b < t$, let $P_{[a,b]}$ be the oriented segment of P that contains the points $p_a, p_{a+1}, \ldots, p_b$, and let $P_{[a,a]} = \{p_a\}$. We are interested in the segments $P_{[a,b]}$ where $x(p_a) = x(p_b)$

The next Definition is useful in identifying segments of P that are "examined" as we scan the curve from left to right. See Figure 7.7 for examples.

Definition 7.9. *A segment $P_{[a,b]}$ is said to* stick *to the vertical line $(x = m)$ if $x(p_a) = x(p_b) = m$, and for $a < c < b$, $x(p_c) \le m$. A segment $P_{[a,b]}$ that sticks to $(x = m)$ is said to be* minimal *if $b - a > 1$, and for $a < c < b$ we have $x(p_c) < m$. Finally, $P_{[a,b]}$ is said to be* entirely on *$(x = m)$ if $x(p_c) = m$, for $a \le c \le b$.*

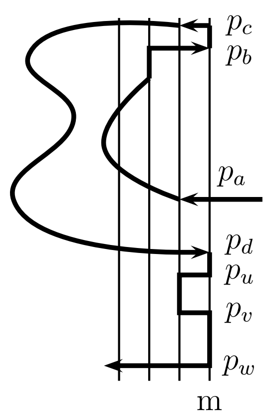

Figure 7.7: Segments that stick to $(x = m)$.

In Figure 7.7, the segments $P_{[a,b]}$, $P_{[a,c]}$, …, $P_{[u,w]}$, $P_{[v,w]}$ all stick to the vertical line $(x = m)$. Among these, $P_{[a,b]}, P_{[c,d]}$ and $P_{[u,v]}$ are minimal, while $P_{[b,c]}, P_{[d,u]}$ and $P_{[v,w]}$ are entirely on $(x = m)$.

Notice that minimal segments that stick to a vertical line $(x = m)$ are disjoint. Also, if $P_{[a,b]}$ is a minimal segment that sticks to $(x = m)$, then the first and the last edges of the segments must be horizontal edges in column $m-1$, i.e., $y(p_a) = y(p_{a+1})$ and $y(p_b) = y(p_{b-1})$. In fact, the left-pointing horizontal edges in column $m-1$ are precisely those of the form $\langle p_a, p_{a+1}\rangle$ for some minimal segment $P_{[a,b]}$ that sticks to the vertical line $(x = m)$, and the right-pointing horizontal edges in column $m-1$ are precisely those of the form $\langle p_{b-1}, p_b\rangle$ for some such minimal segment $P_{[a,b]}$.

These facts are provable in $\mathbf{V}^0$, and show that the Edge Alternation Theorem 7.7 is equivalent to the following lemma (see Figure 7.8). Here (and elsewhere) the assertion that two sets of points on a vertical line alternate means that the two corresponding sets of y-coordinates alternate.

Lemma 7.10 (Edge Alternation Lemma)**.** *(Provable in $\mathbf{V}^0$) Let $P_{[a_1,b_1]}, \ldots, P_{[a_k,b_k]}$ be all minimal segments that stick to the vertical line $(x = m)$. Then the sets $\{p_{a_1}, \ldots, p_{a_k}\}$ and $\{p_{b_1}, \ldots, p_{b_k}\}$ alternate.*

Note that although in $\mathbf{V}^0$ we can define the set of all segments $P_{[a_i,b_i]}$ in the lemma above, we are not able to define k, the total number of such segments. Thus the index k is used only for readability.

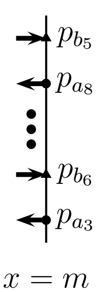

Figure 7.8: The end-edges of minimal segments that stick to $(x = m)$ alternate.

Before proving the Edge Alternation Lemma we give two important lemmas needed for the proof. The first states that the endpoints of two non-overlapping segments of P that stick to the same vertical line do not alternate on the vertical line.

Lemma 7.11 (Main Lemma)**.** *(Provable in $\mathbf{V}^0$) Suppose that $a < b < c < d$ and that* *the segments* $P_{[a,b]}$ *and* $P_{[c,d]}$ *both stick to* $(x = m)$. *Then the sets* $\{y(p_a), y(p_b)\}$ *and* $\{y(p_c), y(p_d)\}$ *do not alternate.*

Proof. We argue in $\mathbf{V}^0$ using induction on m. The base case $(m = 0)$ is straightforward: both $P_{[a,b]}$ and $P_{[c,d]}$ must be entirely on $(x = 0)$. For the induction step, suppose that the lemma is true for some $m \geq 0$. We prove it for $m + 1$ by contradiction.

Assume that there are disjoint segments $P_{[a,b]}$ and $P_{[c,d]}$ sticking to $(x = m + 1)$ that violate the lemma. Take such segments with smallest total length $(b - a) + (d - c)$. It is easy to check that both $P_{[a,b]}$ and $P_{[c,d]}$ must be minimal segments.

Now the segments $P_{[a+1,b-1]}$ and $P_{[c+1,d-1]}$ stick to the vertical line $(x = m)$, and their endpoints have the same y-coordinates as the endpoints of $P_{[a,b]}$ and $P_{[c,d]}$. Hence we get a contradiction from the induction hypothesis. □

From the Main Lemma we can prove an important special case of the Edge Alternation Lemma.

Lemma 7.12 (Provable in $\mathbf{V}^0$). *Let* $P_{[a,b]}$ *be a segment that sticks to* $(x = m)$, *and let* $P_{[a_1,b_1]}, \ldots, P_{[a_k,b_k]}$ *be all minimal subsegments of* $P_{[a,b]}$ *that stick to* $(x = m)$, *where* $a \leq a_1 < b_1 < \ldots < a_k < b_k \leq b$. *Then the sets* $\{p_{a_1}, \ldots, p_{a_k}\}$ *and* $\{p_{b_1}, \ldots, p_{b_k}\}$ *alternate.*

Proof. We show that between any two p_{a_i}'s there is a p_{b_j}. The reverse condition is proved similarly. Thus let $i \neq j$ be such that $y(p_{a_i}) < y(p_{a_j})$. We show that there is some ℓ such that $y(p_{a_i}) < y(p_{b_\ell}) < y(p_{a_j})$.

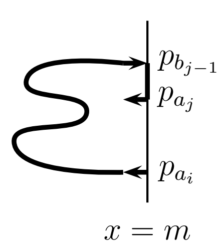

Figure 7.9: Proof of Lemma 7.12

Consider the case where $i < j$ (the other case is similar). Then the segment $P_{[b_{j-1},a_j]}$ is entirely on $(x = m)$. Now if $y(p_{b_{j-1}}) < y(p_{a_j})$, then $y(p_{a_i}) < y(p_{b_{j-1}})$, and we are done. So suppose that $y(p_{b_{j-1}}) > y(p_{a_j})$ (see Figure 7.9).

From the Main Lemma for the segments $P_{[a_i,b_{j-1}]}$ and $P_{[a_j,b_j]}$ it follows that $y(p_{a_i}) < y(p_{b_j}) < y(p_{b_{j-1}})$. Since $P_{[b_{j-1},a_j]}$ is entirely on $(x = m)$, it must be the case that $y(p_{a_i}) < y(p_{b_j}) < y(p_{a_j})$. □

Proof of the Edge Alternation Theorem

To prove Theorem 7.7 it suffices to prove the Edge Alternation Lemma 7.10. The proof relies on Lemma 7.12, the Main Lemma, and the Alternation Lemma 7.8.

Proof of Lemma 7.10. We argue in $\mathbf{V}^0$ and use downward induction on m. The base case, $m = n$, follows from Lemma 7.12, where the segment $P_{[a,b]}$ has $a = 0$ and $b = t-1$. (Recall our numbering convention that the edge $\langle p_{t-1}, p_0\rangle$ lies on the vertical line $(x = n)$.)

For the induction step, suppose that the conclusion is true for $m + 1$, we prove it for m by contradiction.

Let $\{P_{[a'_1,b'_1]}, \ldots, P_{[a'_k,b'_k]}\}$ be the definable set of all minimal segments that stick to the line $(x = m + 1)$. (k is not definable in $\mathbf{V}^0$, we use it only for readability.)

Notation Let $a_\ell = (a'_\ell + 1)$, $b_\ell = (b'_\ell - 1)$ and $A = \{y(p_{a_\ell})\}$, $B = \{y(p_{b_\ell})\}$.

Then, since $y(p_{a_\ell}) = y(p_{a'_\ell})$ and $y(p_{b_\ell}) = y(p_{b'_\ell})$, it follows from the induction hypothesis that A and B alternate. (Note that each $P_{[a_\ell,b_\ell]}$ sticks to $(x = m)$, but might not be minimal.)

Now suppose that there are horizontal P-edges e_1 and e_2 on column $m-1$ that violate the lemma, with $y(e_1) < y(e_2)$. Thus both e_1 and e_2 point in the same direction, and there is no horizontal P-edge e on column $(m - 1)$ with $y(e_1) < y(e) < y(e_2)$. We may assume that both e_1 and e_2 point to the left. The case in which they both point to the right can be argued by symmetry (or we could strengthen the induction hypothesis to apply to both of the curves P and the reverse of P).

Let the right endpoints of e_1 and e_2 be p_c and p_d, respectively. Thus $x(p_c) = x(p_d) = m$ and $y(p_c) < y(p_d)$.

Let $P_{[a_i,b_i]}$ be the segment of P containing p_c, and let $P_{[a_j,b_j]}$ be the segment of P containing p_d. Note that the segments $P_{[a_i,b_i]}$ and $P_{[a_j,b_j]}$ stick to $(x = m)$, but they are not necessarily minimal. It follows from Lemma 7.12 that $i \neq j$.

We may assume that p_{a_j} lies above p_c. This is because if p_{a_j} lies below p_c, then we claim that p_{a_i} lies below p_d (since otherwise the segments $P_{[a_i,c]}$ and $P_{[a_j,d]}$ would violate the Main Lemma). Thus the case p_{a_j} lies below p_c would follow by the case we consider, by interchanging the roles of a_i, c with a_j, d, and inverting the graph.

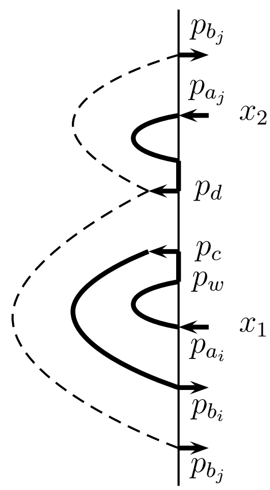

Figure 7.10: Case I: $y(p_{a_i}) < y(p_d)$

Case I: $y(p_{a_i}) < y(p_d)$ (See Figure 7.10)

We apply the Alternation Lemma 7.8 for the alternating sets A and B with the bijection $f(y(p_{a_\ell})) = y(p_{b_\ell})$ and $x_1 = y(p_{a_i})$ and $x_2 = y(p_{a_j})$. Note that f satisfies the non-arc-crossing condition (7.1) by the Main Lemma.

We claim that both $f(x_1)$ and $f(x_2)$ are outside the interval $[x_1, x_2]$. We show this for $f(x_1)$; the argument for $f(x_2)$ is similar. Thus we are to show that the point p_{b_i} does not lie on the vertical line $(x = m)$ between the points p_{a_i} and p_{a_j}.

First we show p_{b_i} does not lie between p_{a_i} and p_c. This is obvious if the segment $P_{[a_i,c]}$

lies entirely on $(x = m)$. Otherwise let $w < c$ be such that the segment $P_{[w,c]}$ lies entirely on $x = m$. (Note that $y(p_{a_i}) < y(p_w) < y(p_c)$, because there is no horizontal edge in column $m-1$ between p_c and p_d.) Then p_{b_i} does not lie between p_{a_i} and p_w by the Main Lemma applied to the segments $P_{[a_i,w]}$ and $P_{[c,b_i]}$.

Next, note that p_{b_i} does not lie between p_c and p_d, because there is no horizontal edge in column $m-1$ between these two points. Finally we claim that p_{b_i} does not lie between p_d and p_{a_j}. This is obvious if $a_j = d$, and otherwise use the Main Lemma applied to the segments $P_{[a_j,d]}$ and $P_{[a_i,b_i]}$.

This establishes the hypotheses for the Alternation Lemma. By that Lemma it follows that there must be some p_{a_ℓ} outside the vertical interval between p_{a_i} and p_{a_j} such that p_{b_ℓ} lies in that interval. But this is impossible, by applying the Main Lemma as above (for either $P_{[a_i,c]}$ and $P_{[a_\ell,b_\ell]}$ or $P_{[a_j,d]}$ and $P_{[a_\ell,b_\ell]}$). This contradiction shows that Case I is impossible.

Case II: $y(p_{a_i}) > y(p_d)$ (See Figure 7.11)

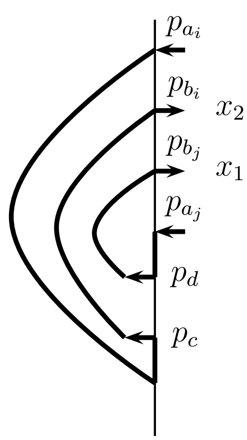

Figure 7.11: Case II: $y(p_{a_i}) > y(p_d)$

In this case we must have $y(p_{a_i}) > y(p_{a_j})$, by the Main Lemma applied to the segments $P_{[a_i,c]}$ and $P_{[a_j,d]}$. In fact, by repeated use of the Main Lemma we can show

$$y(p_{a_j}) < y(p_{b_j}) < y(p_{b_i}) < y(p_{a_i})$$

We get a contradiction by applying the Alternation Lemma, this time using the inverse bijection $f^{-1} : B \to A$, with $x_1 = y(p_{b_j})$ and $x_2 = y(p_{b_i})$. $\square$

7.2.2 There Are at Most Two Regions

Here we formalize and prove the idea that if P is a sequence of edges that form a closed curve, and p_1 and p_2 are points on opposite sides of P, then any point in the plane off P can be connected to either p_1 or p_2 by a path that does not intersect P. However this path must use points in a refined grid, in order not to get trapped in a region such as that depicted in Figure 7.12. Thus we triple the density of the points by tripling n to $3n$, and replace each edge in P by a triple of edges. We also assume that originally the curve P has no point on the border of the grid. (This assumption is different from our convention stated in Section 7.2.1.)

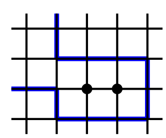

Figure 7.12: An "unwanted" region.

Let P' denote the resulting set of edges. (The new grid has size $(3n) \times (3n)$.)

Theorem 7.13. *The theory* $\mathbf{V}^0$ *proves the following: Let* P *be a sequence of edges that form a closed curve, and suppose that* P *has no point on the border of the grid. Let* P' *be the corresponding sequence of edges in the* $(3n) \times (3n)$ *grid, as above. Let* p_1, p_2 *be any two points on different sides of* P' *(Definition 7.2). Then any point* p *(on the new grid) can be connected to either* p_1 *or* p_2 *by a sequence of edges that does not intersect* P'*.*

Proof. Since edges in P' are directed it makes sense to speak of edges a distance 1 to the left of P' and a distance 1 to the right of P'. Thus, taking care when P' turns corners, it is straightforward to define (using $\mathbf{\Sigma}_0^B$-$\mathbf{COMP}$) two sequences Q_1, Q_2 of edges on either side of P', i.e., both Q_1 and Q_2 have distance 1 (on the new grid) to P'. Then p_1 and p_2

must lie on Q_1 or Q_2. By the Main Theorem for $\mathbf{V}^0$, p_1 and p_2 cannot be on the same Q_i. So assume w.l.o.g. that p_1 is on Q_1 and p_2 is on Q_2.

We describe informally a procedure that gives a sequence of edges connecting any point p to p_1 or p_2. First we compute (using the number minimization principle) the Manhattan distances from p to Q_1 and Q_2 ($d(p, Q_1)$ and $d(p, Q_2)$, respectively). Suppose w.l.o.g. that

$$d(p, Q_1) \leq d(p, Q_2)$$

Let q be a point on Q_1 so that $d(p, q) = d(p, Q_1)$. Then any shortest sequence of edges that connect p and q does not intersect P', because any sequence of edges starting at p that intersect P' must intersect either Q_1 or Q_2 before the first time the sequence intersects P'. Concatenate one such sequence and the sequence of edges on Q_1 that connect q and p_1, we have a sequence of edges that connects p and p_1 without intersecting P'. □

7.3 Proving the st-Connectivity Principle

The st-connectivity principle states that it is not possible to have a red path and a blue path of edges which connect diagonally opposite corners of the grid graph unless the paths intersect. Here we use results from the previous sections to show that the set-of-edges version of this is provable in $\mathbf{V}^0(2)$, and the sequence-of-edges version is provable in $\mathbf{V}^0$. As mentioned in Section 1.2.1, our results here strengthen earlier upper bounds in [Bus06] and [CR97].

Theorem 7.14 (Provable in $\mathbf{V}^0(2)$). *Suppose that B is a set of edges that connects $\langle 0, n-1\rangle$ and $\langle n-1, 0\rangle$, and R is a set of edges that connects $\langle 0, 0\rangle$ and $\langle n-1, n-1\rangle$. Then B and R intersect.*

Proof. We extend the grid (see Figure 7.13) and connect $\langle 0, n-1\rangle$ and $\langle n-1, 0\rangle$ by the

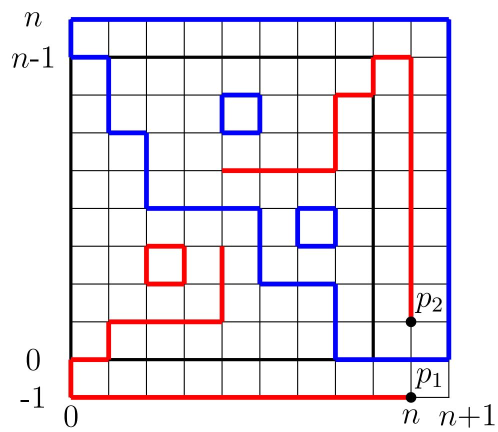

Figure 7.13: Reduction from *st*-Connectivity

edges

$$\{(\langle 0, n-1\rangle, \langle 0, n\rangle), (\langle n-1, 0\rangle, \langle n, 0\rangle), (\langle n, 0\rangle, \langle n+1, 0\rangle)\}\cup$$
$$\{(\langle i, n\rangle, \langle i+1, n\rangle) \ : \ 0 \leq i \leq n\} \cup \{(\langle n+1, j\rangle, \langle n+1, j+1\rangle) \ : \ 0 \leq j \leq n-1\}$$

The above edges together with B form a curve B'.

Similarly, connect the point $\langle 0, 0\rangle$ to $p_1 = \langle n, -1\rangle$ by

$$\{(\langle 0, 0\rangle, \langle 0, -1\rangle)\} \cup \{(\langle i, -1\rangle, \langle i+1, -1\rangle) \ : \ 0 \leq i \leq n-1\}$$

and connect $\langle n-1, n-1\rangle$ to $p_2 = \langle n, 1\rangle$ by the edges

$$\{(\langle n-1, n-1\rangle, \langle n, n-1\rangle)\} \cup \{(\langle n, i\rangle, \langle n, i+1\rangle) \ : \ 1 \leq i \leq n-2\}$$

These edges and the edges in R form a set R' that connects p_1 and p_2.

By Theorem 7.3, B' and R' intersect. As a result, B and R intersect. □

By the same proof, the next result follows from the Main Theorem for $\mathbf{V}^0$.

Theorem 7.15 (Provable in $\mathbf{V}^0$)**.** *Suppose that B is a sequence of edges connecting $\langle 0, n-1\rangle$ and $\langle n-1, 0\rangle$, and R is a sequence of edges connecting $\langle 0, 0\rangle$ and $\langle n-1, n-1\rangle$. Then B and R intersect.*

Chapter 8

Distribution of Prime Numbers

In this chapter we first give an outline of a proof of the lower bound for $\pi(n)$, the number of prime numbers $\leq n$ (Section 8.1). Then we define an approximation to the natural logarithm function $\ln(x)$ (Section 8.2). The $\mathbf{VTC}^0$ proof of the lower bound for $\pi(n)$ is given in Section 8.3. Proof of an upper bound for $\pi(n)$ are outlined and then formalized in Sections 8.4 and 8.5. Finally, Section 8.6 sketches $\mathbf{VTC}^0$ proofs of Bertrand's Postulate (that $\pi(2n) - \pi(n) \geq 1$ for all n) and of the fact that $\pi(2n) - \pi(n) = \Omega(n/\ln(n))$. The proofs outlined in Sections 8.1 and 8.4 can be found in many textbooks, such as [Sho07]. The proof in Section 8.6 is a slight modification of the proof from [Mos49].

Notice that the objects of interest in this chapter are numbers which we treat as objects of the number sort (as opposed to the string sort, for example when we discuss the *integer division* problem on pages 12 and 131). The function $\pi(n)$ mentioned above is a function on the number sort. It apparently cannot be defined by a $\mathbf{\Delta}_0$ formula, so here we use strings and a $\mathbf{\Sigma}_0^B$ formula to define it, and we need the axiom *NUMONES* of $\mathbf{VTC}^0$ to prove its totality and properties.

These theorems can be formalized and proved in $\mathbf{I\Delta}_0$ for $n \leq (\log(a))^c$, for some $c \in \mathbb{N}$ and for some a. Informally, this is because in $\mathbf{I\Delta}_0$ we can define the number of 1-bits in strings whose lengths are polylogarithms in a. The details are left to the reader.

CONVENTION: When p is used as the index in $\sum_p$ or $\prod_p$, then p ranges over prime numbers. Also, in the formalizations below, we will use the fact that that simple manipulations of summation (of numbers) can be carried out in $\mathbf{VTC}^0$ and its extensions. See for example the proof of the Pigeonhole Principle in [Ngu04, CN06]. For readability we will also write formulas using the function $x \dot{-} y$, here $x \dot{-} y = z \leftrightarrow ((x \leq y \wedge z = 0) \vee (y < x \wedge y + z = x))$, as abbreviations for the obvious equivalent formulas without this function. For example, suppose that $\mathbf{VTC}^0 \vdash t_i \geq s_i$ for $0 \leq i \leq n$, then it is provable in $\mathbf{VTC}^0$ that

$$\sum_{i=0}^{n} t_i - \sum_{i=0}^{n} s_i = \sum_{i=0}^{n} (t_i - s_i) \tag{8.1}$$

(Formally, we need to define (using $\mathbf{\Sigma}_0^B$-**COMP**) a string that encodes the sequence $\{t_i\}$ and then use *numones* to compute $\sum_{i=0}^{n} t_i$, etc., but we will omit these details here.)

Note that $\pi(n)$ is provably total in $\mathbf{VTC}^0$: Let $P(n) = \{p \leq n \mid p \text{ is a prime}\}$, then $P(n)$ is defined by $\mathbf{\Sigma}_0^B$-**COMP**, and (recall the function *numones* from Definition 3.1)

$$\pi(n) = numones(n+1, P(n))$$

Note also that rational numbers can be defined in $\mathbf{I\Delta}_0$ (see Section 8.2 below). Therefore our formulations of the Θ-, Ω-, $\mathcal{O}$-notations in $\mathbf{VTC}^0$ will use the rational constants. Recall that $\log(x) = \lfloor \log_2(x) \rfloor$ is definable in $\mathbf{V}^0$ (Example 2.18).

8.1 A Lower Bound Proof for $\pi(n)$

Note that $\pi(2n-1) = \pi(2n)$ for $n \geq 2$. So it suffices to give a lower bound for $\pi(2n)$. The idea is to compute an upper bound and a lower bound for $\frac{(2n)!}{n!n!}$; by comparing these bounds we can derive a lower bound for $\pi(2n)$.

First, for a prime $p < n$, the exponent of p in $n!$ is (see also Lemma 8.8)

$$\sum_{j=1}^{\infty} \left\lfloor \frac{n}{p^j} \right\rfloor \tag{8.2}$$

Hence, for a prime $p < 2n$, the exponent d of p in $\frac{(2n)!}{n!n!}$ is

$$d = \sum_{j=1}^{\infty} \left(\left\lfloor \frac{2n}{p^j} \right\rfloor - 2 \left\lfloor \frac{n}{p^j} \right\rfloor \right) \tag{8.3}$$

Since $\lfloor \frac{2n}{p^j} \rfloor - 2\lfloor \frac{n}{p^j} \rfloor \leq 1$ for $j \leq \frac{\ln(2n)}{\ln(p)}$ and $\lfloor \frac{2n}{p^j} \rfloor - 2\lfloor \frac{n}{p^j} \rfloor = 0$ for $j > \frac{\ln(2n)}{\ln(p)}$, it follows that $d \leq \frac{\ln(2n)}{\ln(p)}$. Therefore,

$$\frac{(2n)!}{n!n!} \leq \prod_{\text{prime } p<2n} p^{\frac{\ln(2n)}{\ln(p)}} = \prod_{\text{prime } p<2n} 2n = (2n)^{\pi(2n)} \tag{8.4}$$

On the other hand,

$$\frac{(2n)!}{n!n!} = \prod_{i=1}^{n} \frac{n+i}{i} \geq 2^n \tag{8.5}$$

Thus $2^n \leq (2n)^{\pi(2n)}$. So $\pi(2n) \geq \frac{\ln(2)}{2} \frac{2n}{\ln(2n)}$.

The value of $\frac{(2n)!}{n!n!}$ is a string, and we do not know how to compute it in $\mathbf{VTC}^0$. To formalize the proof above, we will therefore compute (approximately) the logarithm of $\frac{(2n)!}{n!n!}$ instead. The crude approximation provided by the $\mathbf{AC}^0$ function $\log(x) = \lfloor \log_2(x) \rfloor$ (Example 2.18) seems not sufficient for our purpose, so we will first compute a better approximation (to $\ln(x)$ instead of $\log_2(x)$): we will define $\ln(x, m)$, a rational-valued function that approximates $\ln(x)$ with an error at most $\frac{1}{m}$. Our results will stated using this function.

8.2 Approximating $\ln(x)$

We will approximate the natural logarithm function by rational numbers. Here we only need nonnegative numbers which can be defined in $\mathbf{I\Delta}_0$ by pairs $\langle x, y \rangle$. For readability we will write $\frac{x}{y}$ for $\langle x, y \rangle$. Equality, inequality, addition and multiplication for rational numbers are defined in the standard way, and these are preserved under the embedding $x \mapsto \frac{x}{1}$. For example, $=_{\mathbb{Q}}$ and $\leq_{\mathbb{Q}}$ are defined as:

$$\frac{x}{y} =_{\mathbb{Q}} \frac{x'}{y'} \equiv xy' = x'y, \qquad \text{and} \qquad \frac{x}{y} \leq_{\mathbb{Q}} \frac{x'}{y'} \equiv xy' \leq x'y$$

Then it can be shown that

$$\mathbf{I\Delta}_0 \vdash \lfloor x/y \rfloor \leq_{\mathbb{Q}} \frac{x}{y} <_{\mathbb{Q}} \lfloor x/y \rfloor + 1$$

(here $\lfloor x/y \rfloor$ is the $\mathbf{AC}^0$ function: $\lfloor x/y \rfloor = max\{z : zy \leq x\}$, and $r <_{\mathbb{Q}} s \equiv (r \leq_{\mathbb{Q}} s \wedge r \neq_{\mathbb{Q}} s)$). In the following discussion, we will simply omit the subscript $\mathbb{Q}$ from $=_{\mathbb{Q}}$, $\leq_{\mathbb{Q}}$, etc.; the exact meaning will be clear from the context.

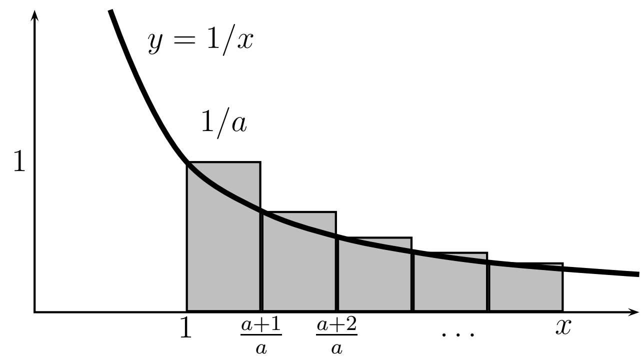

Figure 8.1: Defining $\ln(x, m)$: the shaded area is (8.6).

We will now define in $\mathbf{VTC}^0$ a function $\ln(x, m)$ which approximate $\ln(x)$ up to $1/m$, for $x \in \mathbb{N}$. Note that

$$\ln(x) = \int_1^x \frac{1}{y} dy$$

Our approximation will be roughly (the shaded area in Figure 8.1, here a is to be determined later):

$$\sum_{k=a}^{ax-1} \frac{1}{a} \frac{1}{k/a} = \sum_{k=a}^{ax-1} \frac{1}{k} \tag{8.6}$$

We will not compute this summation precisely (since we want to avoid computing the common denominator). Instead we approximate $\frac{1}{k}$ by $\frac{\lfloor b/k \rfloor}{b}$ for some b determined below. Thus

$$\ln(x, m) = \frac{\sum_{k=a}^{ax-1} \lfloor b/k \rfloor}{b} \tag{8.7}$$

The summation in (8.7) can be computed using the function *numones*.

Notice that (8.6) is an upper bound for $\ln(x)$ with an error (the total area of the shaded region above the line $xy = 1$) at most $1/a$, and (8.7) is a lower bound for (8.6) with an error at most ax/b. So to get an $1/m$-approximation to $\ln(x)$ it suffices to take $a = m, b = m^3$ (we will always have $m > x$).

Definition 8.1. $\ln(x, m)$ *is defined as in* (8.7) *with* $a = m, b = m^3$.

Lemma 8.2. a) $\mathbf{VTC}^0(\ln) \vdash x \le y \le m \supset \ln(x, m) \le \ln(y, m)$.

b) $\mathbf{VTC}^0(\ln) \vdash xy \le m \supset \ln(xy, m) \le \ln(x, m) + \ln(y, m) \le \ln(xy, m) + \frac{xy}{m}$

Proof. Part **a)** is obvious from definition. For part **b)** we have

$$\begin{aligned} \ln(xy, m) &= \frac{1}{b} \sum_{k=a}^{axy-1} \lfloor b/k \rfloor \\ &= \ln(x, m) + \frac{1}{b} \sum_{k=ax}^{axy-1} \lfloor b/k \rfloor \\ &= \ln(x, m) + \frac{1}{b} \sum_{k=a}^{ay-1} \sum_{j=0}^{x-1} \lfloor b/(kx + j) \rfloor \end{aligned} \tag{8.8}$$

Now using the facts (provable in $\mathbf{I\Delta}_0$) that $\lfloor b/(kx + j) \rfloor \le \lfloor b/(kx) \rfloor$ (for $0 \le j < x$) and $x \lfloor b/(kx) \rfloor \le \lfloor b/k \rfloor$ we have:

$$\ln(xy, m) \le \ln(x, m) + \frac{1}{b} \sum_{k=a}^{ay-1} x \lfloor b/(kx) \rfloor \le \ln(x, m) + \frac{1}{b} \sum_{k=a}^{ay-1} \lfloor b/k \rfloor = \ln(x, m) + \ln(y, m)$$

Also, from (8.8) and the facts (provable in $\mathbf{I\Delta}_0$) that $\lfloor b/(kx + j) \rfloor \ge \lfloor b/((k + 1)x) \rfloor$ (for $0 \le j < x$) and $x \lfloor b/((k + 1)x) \rfloor \ge \lfloor b/(k + 1) \rfloor - (x - 1)$, we have

$$\begin{aligned} \ln(xy, m) &\ge \ln(x, m) + \frac{1}{b} \sum_{k=a}^{ay-1} x \lfloor b/((k + 1)x) \rfloor \\ &\ge \ln(x, m) + \frac{1}{b} \sum_{k=a}^{ay-1} (\lfloor b/(k + 1) \rfloor - (x - 1)) \\ &= \ln(x, m) + \frac{1}{b} \left(\left(\sum_{k=a}^{ay-1} \lfloor b/k \rfloor \right) - ay(x - 1) - \lfloor b/a \rfloor + \lfloor b/(ay) \rfloor \right) \\ &= \ln(x, m) + \ln(y, m) - \frac{my(x - 1) + m^2 - \lfloor m^3/(my) \rfloor}{m^3} \quad \text{(recall } a = m, b = m^3\text{)} \\ &\ge \ln(x, m) + \ln(y, m) - \frac{xy}{m} \end{aligned}$$

∎

Lemma 8.3 (Provable in $\mathbf{VTC}^0(\ln)$). *For $n+2<m$:*

$$\ln(n,m)+\frac{1}{n+1}-\frac{1}{m^2}<\ln(n+1,m)\le\ln(n,m)+\frac{1}{n}$$

Proof. Consider the second inequality. By Definition 8.1 we have

$$\ln(n+1,m)-\ln(n,m)=\frac{\sum_{i=0}^{m-1}\lfloor m^3/(mn+i)\rfloor}{m^3}\le\frac{m\lfloor m^3/(mn)\rfloor}{m^3}=\frac{\lfloor m^2/n\rfloor}{m^2}\le\frac{1}{n}$$

Similarly, because $\lfloor\frac{m^3}{mn+i}\rfloor\ge\lfloor\frac{m^3}{m(n+1)}\rfloor>\frac{m^2}{n+1}-1$:

$$\ln(n+1,m)-\ln(n,m)>\frac{1}{m^3}m(\frac{m^2}{n+1}-1)=\frac{1}{n+1}-\frac{1}{m^2}$$ □

Lemma 8.4 (Provable in $\mathbf{VTC}^0(\ln)$). *For $m>n$:*

$$n\ln(n,m)-n+1\le\sum_{i=1}^{n}\ln(n,m)<n\ln(n,m)-n+\ln(n,m)+2$$

Proof. The first inequality is proved as follows,

$$\begin{aligned}n\ln(n,m)-\sum_{i=1}^{n}\ln(n,m)&=\sum_{i=1}^{n-1}i(\ln(i+1,m)-\ln(i,m))\\&\le\sum_{i=1}^{n-1}i\frac{1}{i}=n-1\qquad\text{(Lemma 8.3)}\end{aligned}$$

Similarly,

$$\begin{aligned}(n+1)\ln(n,m)-\sum_{i=1}^{n}\ln(n,m)&=\sum_{i=2}^{n}i(\ln(i,m)-\ln(i-1,m))\\&\ge\sum_{i=2}^{n}i(\frac{1}{i}-\frac{1}{m^2})\qquad\text{(Lemma 8.3)}\\&=n-1-\frac{n(n+1)-2}{2m^2}>n-2\qquad\text{(for } m>n)\end{aligned}$$ □

The following corollary relates $\ln(n,m)$ to $\log(n)=\lfloor\log_2(n)\rfloor$ (one less than the length of the binary representation of n, see Example 2.18).

Corollary 8.5 (Provable in $\mathbf{VTC}^0(\ln,\log)$). *For $2n\le m$,*

$$\log(n)\ln(2,m)-\frac{n}{m}\le\ln(n,m)\le(\log(n)+1)\ln(2,m)$$

Proof. Let $y = 2^{\log(n)}$ (y is definable in $\mathbf{I\Delta}_0$). Then $y \le n < 2y$. Lemma 8.2 **a)** shows that $\ln(y, m) \le \ln(n, m) \le \ln(2y, m)$. We can prove by induction on y using Lemma 8.2 **b)** that $\ln(2y, m) \le (\log(n) + 1)\ln(2, m)$ and $\ln(y, m) \ge \log(n)\ln(2, m) - \frac{y}{m}$. $\square$

The following lemma is used to calculate $\ln(2)$. Here we write $\|t_1 - t_2\| \le s$ as an abbreviation for $t_1 \le t_2 + s \wedge t_2 \le t_1 + s$.

Lemma 8.6 (Provable in $\mathbf{VTC}^0(\ln)$)**.** *For $x < m \wedge m \ge 3$,*

$$\|\ln(x, m) - \ln(x, 2m)\| < \frac{x}{4m}$$

Proof. From definition we have

$$\begin{aligned}\ln(x, 2m) - \ln(x, m) &= \frac{1}{8m^3}\sum_{k=2m}^{2mx-1}\lfloor 8m^3/k\rfloor - \frac{1}{m^3}\sum_{k=m}^{mx-1}\lfloor m^3/k\rfloor \\ &= \frac{1}{8m^3}\sum_{k=m}^{mx-1}(\lfloor 8m^3/2k\rfloor + \lfloor 8m^3/(2k+1)\rfloor - 8\lfloor m^3/k\rfloor)\end{aligned}$$

For $m \le k < mx$, let $\lfloor m^3/k\rfloor = q$, then it can be shown that

$$4q \le \lfloor 8m^3/2k\rfloor \le 4q + 3$$

$$4q - 2m \le \lfloor 8m^3/(2k+1)\rfloor \le 4q + 3$$

In other words, for $m \ge 3$ we have $\|\lfloor 8m^3/2k\rfloor + \lfloor 8m^3/(2k+1)\rfloor - 8\lfloor m^3/k\rfloor\| \le 2m$. Consequently, $\|\ln(x, 2m) - \ln(x, m)\| \le \frac{1}{8m^3}(mx - m)2m < \frac{x}{4m}$. $\square$

The lemma above can be used to approximate $\ln(2)$ using $\ln(2, m)$ where m is a power of 2. Here we give only a rough estimation. (Note that in particular we have $\frac{1}{2} < \ln(2, 2^{\log(m)}) < 1$ for $m \ge 8$).

Corollary 8.7 (Provable in $\mathbf{VTC}^0(\ln)$)**.** *For $8 \le m$,*

$$\frac{19}{32} < \ln(2, 2^{\log(m)}) < \frac{27}{32}$$

Proof. Lemma 8.6 can be used to show that $\|\ln(x, m) - \ln(x, 2^k m)\| < \frac{x}{2m}$. It follows that $\|\ln(2, m) - \ln(2, 2^k m)\| < \frac{2}{16}$. Also, we have $\ln(2, 8) = \frac{368}{512} = \frac{23}{32}$. $\square$

8.3 A Lower Bound Proof of $\pi(x)$ in $\mathbf{VTC}^0$

Throughout this section, let

$$s = \sum_{i=1}^{n} \ln(n+i, m) - \sum_{i=1}^{n} \ln(i, m) \qquad \text{for some } m > n^2 \tag{8.9}$$

Using the fact that $\ln(n+i, m) \geq \ln(i, m)$ (Lemma 8.2) we can prove in $\mathbf{VTC}^0(\ln)$ that

$$s = \sum_{i=1}^{n} (\ln(n+i, m) - \ln(i, m)) \tag{8.10}$$

(see (8.1)). Also,

$$s = \sum_{i=1}^{2n} \ln(i, m) - 2\sum_{i=1}^{n} \ln(i, m) \tag{8.11}$$

Using *NUMONES* and the fact that the relation $x^z = y$ is definable by a $\mathbf{\Delta}_0$ formula (Example 2.6), the following functions are provably total in $\mathbf{VTC}^0$ (in fact $\mathbf{ex}(p, n)$ is provably total in $\mathbf{I\Delta}_0$, here $\mathbf{ex}$ stands for exponent):

$$\mathbf{ex}(p, n) = max\{j \ : \ p^j | n\} \tag{8.12}$$

$$\mathbf{ex}(p, n!) = \sum_{i=1}^{n} \mathbf{ex}(p, i) \tag{8.13}$$

(These two functions have the same name, but the exact meaning will be clear from context.) Following (8.2) we have:

Lemma 8.8 (Provable in $\overline{\mathbf{VTC}}^0$)**.** $\mathbf{ex}(p, n!) = \sum_{j: p^j \leq n} \lfloor n/p^j \rfloor$.

Proof. The proof is by formalizing in $\mathbf{VTC}^0$ the standard proof by a counting argument: First we count the number of $i \leq n$ such that $p|i$ (there are $\lfloor n/p \rfloor$ of them), then we count the number of $i \leq n$ such that $p^2|i$ (there are $\lfloor n/p^2 \rfloor$ of them), etc. □

Prime factorization gives us:

Lemma 8.9 (Provable in $\overline{\mathbf{VTC}}^0$)**.**

$$\sum_{p|i} \mathbf{ex}(p, i) \ln(p, m) - \frac{i}{m} \leq \ln(i, m) \leq \sum_{p|i} \mathbf{ex}(p, i) \ln(p, m)$$

Proof. Each inequality can be proved by induction on i using Lemma 8.2 **b**. □

The next corollary follows easily:

Corollary 8.10 (Provable in $\overline{\mathbf{VTC}}^0$)**.**

$$\sum_{p\leq n}\mathbf{ex}(p,n!)\ln(p,m)-\frac{n(n+1)}{2m}\leq\sum_{i=1}^{n}\ln(i,m)\leq\sum_{p\leq n}\mathbf{ex}(p,n!)\ln(p,m)$$

Following (8.4) we prove (recall s from (8.9)):

Lemma 8.11 (Provable in $\mathbf{VTC}^0(\pi,\ln)$)**.** *For $m\geq 4n^2$, $s\leq\pi(2n)\ln(2n,m)+1$.*

Proof. Using (8.11) and from Corollary 8.10 and Lemma 8.8 above we have

$$\begin{aligned}
s &= \sum_{i=1}^{2n}\ln(i,m)-2\sum_{i=1}^{n}\ln(i,m)\\
&\leq \sum_{p\leq 2n}\mathbf{ex}(p,(2n)!)\ln(p,m)-2\sum_{p\leq n}\mathbf{ex}(p,n!)\ln(p,m)+\frac{n(n+1)}{m}\\
&= \sum_{p\leq 2n}\ln(p,m)(\mathbf{ex}(p,(2n)!)-2\mathbf{ex}(p,n!))+\frac{n(n+1)}{m}\\
&= \sum_{p\leq 2n}\ln(p,m)\sum_{p:p^j\leq 2n}(\lfloor 2n/p^j\rfloor-2\lfloor n/p^j\rfloor)+\frac{n(n+1)}{m}\\
&\leq \sum_{p\leq 2n}\ln(p,m)\cdot max\{j\ :\ p^j\leq 2n|\}+\frac{n(n+1)}{m}
\end{aligned}$$

The last inequality follows from the fact that $\mathbf{I\Delta}_0\vdash 0\leq\lfloor 2r\rfloor-2\lfloor r\rfloor\leq 1$ for rationals r.

Now we estimate $\ln(p,m)\cdot max\{j\ :\ p^j\leq 2n|\}$. Suppose that $p^j\leq 2n$, we can prove (by induction on j, using Lemma 8.2 **b**) that

$$j\ln(p,m)\leq\ln(p^j,m)+\frac{p^j}{m}$$

Therefore by Lemma 8.2 **a** we have $j\ln(p,m)\leq\ln(2n,m)+\frac{2n}{m}$. As a result,

$$\ln(p,m)\cdot max\{j\ :\ p^j\leq 2n|\}\leq\ln(2n,m)+\frac{2n}{m}$$

Hence $s\leq\pi(2n)\ln(2n,m)+\pi(2n)\frac{2n}{m}+\frac{n(n+1)}{m}<\pi(2n)\ln(2n,m)+1$ for $m>4n^2$. □

The next lemma is stronger than (8.5):

Lemma 8.12 (Provable in $\mathbf{VTC}^0(\pi, \ln)$). *For $m \geq 4n^2$ (recall s from (8.11)),*

$$s > 2n \ln(2, m) - 2\ln(2n, m) - 4 \tag{8.14}$$

Proof. By Lemma 8.4 we have

$$\begin{aligned}
s &> 2n \ln(2n, m) - 2n + 1 - 2(n \ln(n, m) - n + \ln(n, m) + 2) \\
&= 2n \ln(2n, m) - 2n \ln(n, m) - 2\ln(2n, m) - 3 \\
&> 2n(\ln(2, m) + \ln(n, m) - \frac{2n}{m}) - 2n \ln(n, m) - 2\ln(2n, m) - 3 \qquad \text{(Lemma 8.2)} \\
&\geq 2n \ln(2, m) - 2\ln(2n, m) - 4
\end{aligned}$$

□

Corollary 8.13 (Provable in $\mathbf{VTC}^0(\pi, \ln)$). *For $n^2 < m$,*

$$n \ln(2, m) < (\pi(n) + 2) \ln(n, m) + 5 \tag{8.15}$$

Proof. The case where $n = 2k$ follows from Lemmas 8.11 and 8.12.

Now consider the case where $n = 2k - 1$. Using Lemma 8.3 and the fact that $\pi(2k - 1) = \pi(2k)$ for $k \geq 2$:

$$\begin{aligned}
(\pi(2k-1) + 2)\ln(2k-1, m) + 5 &\geq (\pi(2k) + 2)(\ln(2k, m) - \frac{1}{2k-1}) + 5 \\
&> 2k \ln(2, m) - \frac{\pi(2k) + 2}{2k - 1} \qquad \text{(by the case } n = 2k\text{)} \\
&= (2k - 1)\ln(2, m) + \left(\ln(2, m) - \frac{\pi(2k) + 2}{2k - 1}\right)
\end{aligned}$$

Since $\ln(2, m) > \frac{19}{32}$ (Lemma 8.7) and $\pi(2k - 1) \leq k - 2$ (for $k \geq 8$), we have $\ln(2, m) - \frac{\pi(2k)+2}{2k-1} \geq 0$ for $k \geq 8$. As a result, the corollary holds when $k \geq 8$. The corollary can be verified for $n = 2k - 1$ and $k \leq 7$ directly. □

8.4 Outline of an Upper Bound Proof of $\pi(n)$

Recall that index p ranges over prime numbers. Chebyshev's function $\vartheta(n)$ defined below plays an important role:

$$\vartheta(x) = \sum_{p \le x} \ln(p) \tag{8.16}$$

Theorem 8.14. $\lim \frac{\pi(x)}{\vartheta(x)/\ln(x)} = 1$.

We will only need the fact that $\pi(x) = \mathcal{O}(\vartheta(x)/\ln(x))$.

Proof. First, since $\ln(p) \le \ln(x)$ for $p \le x$, we have $\vartheta(x) \le \pi(x)\ln(x)$, i.e., $\pi(x) \ge \vartheta(x)/\ln(x)$.

On the other hand, for any $\epsilon > 0$ we have

$$\vartheta(x) \ge \sum_{x^{1-\epsilon} < p \le x} \ln(p) \ge (1-\epsilon)\ln(x)(\pi(x) - \pi(x^{1-\epsilon})) \ge (1-\epsilon)\ln(x)(\pi(x) - x^{1-\epsilon})$$

So $\pi(x) \le x^{1-\epsilon} + \frac{\vartheta(x)}{(1-\epsilon)\ln(x)}$. Since $\pi(x) = \Omega(x/\ln(x))$ (Section 8.1), for sufficiently large x we have $x^{1-\epsilon} \le \epsilon\pi(x)$, and hence $\pi(x) \le \frac{\vartheta(x)}{(1-\epsilon)^2\ln(x)}$. □

Theorem 8.15. *For $n \ge 1$, $\vartheta(n) < 2n\ln(2)$.*

Proof. First, notice that

$$\frac{(2k+1)!}{k!(k+1)!} \le \frac{1}{2}2^{2k+1} = 2^{2k} \tag{8.17}$$

because $\frac{(2k+1)!}{k!(k+1)!}$ appears twice in the binomial expansion of 2^{2k+1}.

Also, all primes p where $k+1 < p \le 2k+1$ divide $\frac{(2k+1)!}{k!(k+1)!}$. Hence

$$\frac{(2k+1)!}{k!(k+1)!} \ge \prod_{k+1 < p \le 2k+1} p \tag{8.18}$$

Consequently,

$$\vartheta(2k+1) - \vartheta(k+1) = \sum_{k+1 < p \le 2k+1} \ln(p) \le \ln\frac{(2k+1)!}{k!(k+1)!} \le \ln(2^{2k}) = 2k\ln(2) \tag{8.19}$$

Now we prove the theorem by induction on n. The base cases ($n = 1$ and $n = 2$) are trivial. For the induction step, the case where n is even is also obvious, since then

$\vartheta(n) = \vartheta(n-1)$. So suppose that $n = 2k+1$. Using (8.19) and the induction hypothesis (for $n = k+1$) we have $\vartheta(2k+1) < 2k\ln(2) + 2(k+1)\ln(2) = 2(2k+1)\ln(2)$. $\square$

Corollary 8.16. *For every $\epsilon > 0$, there is a $n_0 \in \mathbb{N}$ so that for all $n \geq n_0$,*

$$\pi(n) \leq (1+\epsilon)\frac{2\ln(2)x}{\ln(x)}$$

Again, we have to avoid computing $\frac{(2k+1)!}{k!(k+1)!}$ in $\mathbf{VTC}^0$. In the formalization below, we will approximate $\vartheta(x)$ by $\vartheta(x,m)$ (using $\ln(x,m)$ defined in Section 8.2), and will show that $\vartheta(x,m) = \mathcal{O}(x)$ using a proof slightly different from the above proof.

8.5 An Upper Bound Proof of $\pi(x)$ in $\mathbf{VTC}^0$

Our version of Chebyshev' function is

$$\vartheta(x,m) = \sum_{p \leq x} \ln(p,m) \tag{8.20}$$

Note that $\vartheta(x,m)$ is provably total in $\mathbf{VTC}^0$. Following Theorem 8.15 we prove:

Theorem 8.17 (Provable in $\overline{\mathbf{VTC}}^0$)**.** *For $(n+1)^2 \leq m$*

$$\vartheta(n,m) \leq (2\ln(2,m))n + (\ln(n-1,m))^2 + 3\ln(n-1,m)$$

Proof. Using lemma 8.8 we can prove the following formalization of (8.18):

$$\sum_{k+1<p\leq 2k+1} \ln(p,m) \leq \sum_{i=1}^{2k+1} \ln(i,m) - \sum_{i=1}^{k} \ln(i,m) - \sum_{i=1}^{k+1} \ln(i,m)$$

The LHS is $\vartheta(2k+1,m) - \vartheta(k+1,m)$, so using Lemma 8.4 we have

$$\vartheta(2k+1,m) - \vartheta(k+1,m) < (2k+2)\ln(2k+1,m) - k\ln(k,m) - (k+1)\ln(k+1,m)$$

By Lemma 8.3,

$$\begin{aligned}
\text{RHS} &\le (2k+2)(\ln(2k,m)+\frac{1}{2k}) - k\ln(k,m) - (k+1)(\ln(k,m)+\frac{1}{k+1}-\frac{1}{m^2})\\
&\le (2k+2)(\ln(2,m)+\ln(k,m)) - (2k+1)\ln(k,m) + \frac{1}{k} + \frac{k+1}{m^2}\\
&\le 2k\ln(2,m) + \ln(k,m) + (2\ln(2,m) + \frac{1}{k} + \frac{k+1}{m^2})\\
&\le 2k\ln(2,m) + \ln(k,m) + 3
\end{aligned}$$

As in the proof of Theorem 8.15, the current theorem can be proved by strong induction on k. □

Now we prove the upper bound for $\pi(x)$ in $\mathbf{VTC}^0(\pi,\ln)$.

Corollary 8.18 (Provable in $\mathbf{VTC}^0(\pi,\ln)$)**.** *For $n \ge 2$ and $(n+1)^2 \le m$:*

$$\pi(n) \le 4\ln(2,m)\frac{n}{\ln(n,m)} + \lceil\sqrt{n}\rceil + \ln(n,m) + 3$$

Note that the RHS can be bounded by $c\frac{n}{\ln(n,m)}$ for some constant $c > 4\ln(2,m)$ when n is sufficiently large, but we leave the details to the reader.

Proof. We follow the proof of the fact that $\pi(x) = \mathcal{O}(\vartheta(x)/\ln(x))$ (Theorem 8.14). Let $\lceil\sqrt{x}\rceil = \min\{y \le x \ : \ y^2 \ge x\}$. Then by Lemma 8.2, for $x \le m$ we have

$$\ln(\lceil\sqrt{x}\rceil, m) \ge \frac{\ln(x,m)}{2}$$

Hence for $m \ge (n+1)^2$:

$$\vartheta(n,m) \ge \sum_{\lceil\sqrt{n}\rceil < p \le n} \ln(p,m) \ge \ln(\lceil\sqrt{n}\rceil, m)(\pi(n) - \pi(\lceil\sqrt{n}\rceil)) \ge \frac{\ln(n,m)}{2}(\pi(n) - \lceil\sqrt{n}\rceil)$$

Therefore $\pi(n) \le 2\frac{\vartheta(n)}{\ln(n,m)} + \lceil\sqrt{n}\rceil$. Consequently, the upper bound for $\pi(n)$ follows from Theorem 8.17 □

8.6 Bertrand's Postulate and a Lower Bound for $\pi(2n) - \pi(n)$

We will show that $\pi(2n)-\pi(n) \geq 1$ for all n, and $\pi(2n)-\pi(n) = \Omega(n/\ln(n))$. The upper bound for $\pi(n)$ proved in the previous section does not allow us to formalize directly the proof from [Mos49]. However, it suffices for a proof obtained by slightly modifying the proof from [Mos49]. First, the following lemma is easily proved using (8.3):

Lemma 8.19. *Suppose that p is a prime number where $\lceil\sqrt{2n}\rceil \leq p \leq \lfloor 2n/3 \rfloor$. Then p occurs in the prime factorization of $\frac{(2n)!}{n!n!}$ at most once, and p occurs exactly once in $\frac{(2n)!}{n!n!}$ if and only if for some $1 \leq c \in \mathbb{N}$:*

$$\left\lfloor \frac{n}{c+1} \right\rfloor + 1 \leq p \leq \left\lfloor \frac{2n}{2c+1} \right\rfloor$$

Theorem 8.20. *For $n > 31^2/2$, every prime number p where $\lfloor \frac{n}{c+1} \rfloor + 1 \leq p \leq \lfloor \frac{2n}{2c+1} \rfloor$ for some c, $1 \leq c \leq 14$, occurs exactly once in the prime factorization of $A = A_1 \cdot A_2 \cdot A_3$, where*

$$A_1 = \binom{\lfloor 2n/3 \rfloor}{\lfloor n/6 \rfloor}, \; A_2 = \binom{\lfloor 2n/5 \rfloor}{\lfloor n/15 \rfloor}, \; A_3 = \binom{\lfloor 2n/13 \rfloor}{\lfloor n/91 \rfloor} \tag{8.21}$$

Proof. The condition $n \geq 31^2/2$ guarantees that $n/15 \geq \lceil\sqrt{2n}\rceil$. The proof is by counting, using (8.2), the difference between the number of occurrences of a prime number p in the numerator and denominator of A. For example,

$$A_1 = \frac{(\lfloor 2n/3 \rfloor - \lfloor n/6 \rfloor + 1) \cdot (\lfloor 2n/3 \rfloor - \lfloor n/6 \rfloor + 2) \cdot \ldots \cdot \lfloor 2n/3 \rfloor}{1 \cdot 2 \cdot \ldots \cdot \lfloor n/6 \rfloor}$$

Hence, any prime number p where $\lfloor \frac{n}{2} \rfloor + 1 \leq p \leq \lfloor \frac{2n}{3} \rfloor$ occurs exactly once in A_1, because p appears in the numerator but no multiple of p appears in the denominator.

Similarly, any p where $\lfloor \frac{n}{c+1} \rfloor + 1 \leq p \leq \lfloor \frac{2n}{2c+1} \rfloor$, for $c \in \{3, 4, 5, 9\}$, occurs exactly once in A_1; and any p such that $\lfloor \frac{n}{c+1} \rfloor + 1 \leq p \leq \lfloor \frac{2n}{2c+1} \rfloor$, for $c \in \{2, 7, 8, 10, 11, 12, 13, 14\}$, occurs exactly once in A_2; and any p where $\lfloor \frac{n}{7} \rfloor + 1 \leq p \leq \lfloor \frac{2n}{13} \rfloor$ occurs exactly once in A_3. □

Corollary 8.21. *Suppose that $n \geq 31^2/2$, and let A be as in Theorem 8.20. Then*

$$\frac{(2n)!}{n!n!} \leq A \cdot (2n)^{\pi(2n/31)} \cdot \prod_{n<p<2n} p \tag{8.22}$$

Proof. The condition $n \geq 31^2/2$ ensures that $\lfloor 2n/31 \rfloor \geq \lceil \sqrt{2n} \rceil$. By Lemma 8.19, each prime p where $\lfloor 2n/31 \rfloor < p \leq n$ occurs at most once in $\frac{(2n)!}{n!n!}$, and the product of those that occur exactly once is bounded above by A by Theorem 8.20. In addition, for each of the $\pi(2n/31)$ primes $p \leq \lfloor 2n/31 \rfloor$, by (8.3) we know that the exponent d of p in $\frac{(2n)!}{n!n!}$ is at most $\frac{\ln(2n)}{\ln(p)}$, i.e., $p^d \leq 2n$. □

Corollary 8.22. $\pi(2n) - \pi(n) = \Omega(n/\ln(n))$ *and* $\pi(2n) - \pi(n) \geq 1$ *for all* $n \geq 1$.

Proof. Since $\frac{(2n)!}{n!n!}$ is the largest coefficient in the binomial expansion of 2^{2n}, we have

$$\frac{2^{2n}}{2n+1} \leq \frac{(2n)!}{n!n!}$$

Also, (recall A_i in (8.21)) $A_1 \leq 2^{2n/3}$, $A_2 \leq 2^{2n/5}$ and $A_3 \leq 2^{2n/13}$. Thus (8.22) gives us

$$\frac{2^{2n}}{2n+1} \leq 2^{\frac{2n}{3}+\frac{2n}{5}+\frac{2n}{13}} \cdot (2n)^{\pi(2n/31)} \cdot \prod_{n<p<2n} p$$

Hence

$$\ln(\prod_{n<p<2n} p) \geq \ln(2)(2n - \frac{2n}{3} - \frac{2n}{5} - \frac{2n}{13}) - \pi(\frac{2n}{31})\ln(2n) = \frac{152\ln(2)}{195}n - \pi(\frac{2n}{31})\ln(2n)$$

From the proof of Theorem 8.14, setting $\epsilon = 1/2$ and note that $x^{1/2} < \frac{1}{2}\pi(x)$ for $x \geq 4096$, (using $\pi(x) \geq \frac{\ln(2)}{2}\frac{x}{\ln(x)}$ by Section 8.1) we have $\pi(x) \leq \frac{4\vartheta(x)}{\ln(x)}$. Hence, from Theorem 8.15, $\pi(x) \leq 8\ln(2)\frac{x}{\ln(x)}$. In other words,

$$\pi(\frac{2n}{31}) \leq \frac{16\ln(2)}{31}\frac{n}{\ln(2n/31)}$$

Consequently,

$$\ln(\prod_{n<p<2n} p) \geq \frac{152\ln(2)}{195}n - \frac{16\ln(2)}{31}\frac{n}{\ln(2n/31)}\ln(2n) \tag{8.23}$$

Hence $\ln(2n^{\pi(2n)-\pi(n)}) \geq \ln(\prod_{n<p<2n} p) = \Omega(n)$. So $\pi(2n) - \pi(n) = \Omega(n/\ln(n))$.

In addition, the RHS of (8.23) is > 0 whenever $n \geq 12975$. It follows that $\pi(2n) - \pi(n) \geq 1$ for $n \geq 12975$. The fact that $\pi(2n) > \pi(n)$ for $n < 12975$ can be checked directly. □

8.6.1 Formalization in $\mathbf{VTC}^0$

Theorem 8.23. a) *It is provable in* $\mathbf{VTC}^0(\pi)$ *that* $\pi(2n) - \pi(n) \geq 1$ *for* $n \geq 1$.

b) *There are* $r, s \in \mathbb{N}, r > 0, s > 0, n_0 \in \mathbb{N}$ *so that*

$$\mathbf{VTC}^0(\pi, \ln) \vdash n \geq n_0 \supset \pi(2n) - \pi(n) \geq \frac{r}{s}\frac{n}{\ln(n)}$$

Proof Sketch. Notice that Corollary 8.21 can be formalized and proved in $\mathbf{VTC}^0(\pi, \ln)$ as in previous sections, i.e., the following is provable in $\mathbf{VTC}^0(\pi, \ln)$ (writing $\ln(x)$ for $\ln(x, m)$):

$$\begin{aligned}\sum_{i=1}^{2n} \ln(i) - 2\sum_{i=1}^{n} \ln(i) \leq \ln(2n)\pi(2n/31) + \sum_{n<p<2n} \ln(p) \\ + \sum_{c\in\{1,2,6\}} \left(\sum_{i=1}^{\lfloor 2n/(2c+1)\rfloor} \ln(i) - \sum_{i=1}^{\lfloor n/(c+1)\rfloor} \ln(i) - \sum_{i=1}^{\lfloor 2n/(2c+1)\rfloor - \lfloor n/(c+1)\rfloor} \ln(i) \right)\end{aligned} \tag{8.24}$$

Now by Lemma 8.18, for $n \geq 31$ we have

$$\pi(2n/31) \leq \frac{8\ln(2, m)}{31\ln(\lfloor 2n/31\rfloor)}n + \mathcal{O}(\lceil\sqrt{n}\rceil)$$

Next, it can be shown (using Lemmas 8.2 and 8.3) that for $n \geq 3$,

$$\sum_{i=1}^{\lfloor 2n/3\rfloor} \ln(i) - \sum_{i=1}^{\lfloor n/2\rfloor} \ln(i) - \sum_{i=1}^{\lfloor 2n/3)\rfloor - \lfloor n/2\rfloor} \ln(i) < (4\ln(4) - 3\ln(3))\frac{n}{6} + \mathcal{O}(\ln(n))$$

Similarly, for each c the summand on the second line of (8.24) is less than

$$((2c+2)\ln(2c+2) - (2c+1)\ln(2c+1))\frac{n}{(c+1)(2c+1)} + \mathcal{O}(\ln(n))$$

Hence, the RHS of (8.24) is at most

$$tn + \mathcal{O}(\ln(n)\lceil\sqrt{n}\rceil) + \sum_{n<p<2n} \ln(p)$$

where

$$t = \frac{4\ln(4) - 3\ln(3)}{6} + \frac{6\ln(6) - 5\ln(5)}{15} + \frac{14\ln(14) - 13\ln(13)}{91} + \frac{8\ln(2)\ln(2n)}{31\ln(\lfloor 2n/31\rfloor)}$$

The lower bound for the LHS of (8.24) is from Lemma 8.12. So we have

$$2n\ln(2) - 2\ln(2n) - 4 < tn + \mathcal{O}(\ln(n)\lceil\sqrt{n}\rceil) + \sum_{n<p<2n} \ln(p)$$

The conclusion follows from the fact that $t < 2\ln(2)$, $\ln(n)\lceil\sqrt{n}\rceil = o(n)$, and that $\log(n) = \Theta(\ln(n, m))$ for $m > n$ (Corollary 8.5). □

8.7 Comparison with Earlier Work

As this thesis is about to be submitted, we become aware of [CD94] and [Cor95]. In [CD94] the Prime Number Theorem has been formalized and proved in the theory $\mathbf{I\Delta}_0 +$ *exp*, where *exp* is the axiom $\forall x \forall z \exists y (y = x^z)$ (here $y = x^z$ is the $\mathbf{\Delta}_0$ formula defining the graph of the exponentiation function, see [Ben62, HP93, Bus98, CN06]). Indeed, it is remarked in [CD94] that much of their formalization can be done in the theory $\mathbf{I}\mathcal{E}^2$, or even an apparently weaker theory which we call $\mathbf{I\Delta}_0 +$ *counting* (see below). Here $\mathbf{I}\mathcal{E}^2$ is the theory that extends $\mathbf{I\Delta}_0$ by the defining axioms for functions of Grzegorczyk's class $\mathcal{E}^2$ (linear space) together with the induction axioms for bounded formulas containing these functions.

Let $\mathbf{I\Delta}_0 +$ *counting* be the extension of $\mathbf{I\Delta}_0$ defined as follows. For each $\mathbf{\Delta}_0$-formula $\varphi(x, \vec{y})$ we introduce a function $f_{\varphi(x,\vec{y})}(z, \vec{y})$ whose value is the cardinality of the set $\{x \,:\, x \le z \wedge \varphi(x, \vec{y})\}$. Now define $\mathbf{I\Delta}_0 +$ *counting* to be $\mathbf{I\Delta}_0$ together with the defining axioms (in the style of (3.1)–(3.3) on page 26) for the new functions and induction axioms on bounded formulas in the new language. It can be shown that $\mathbf{I\Delta}_0 +$ *counting* is equivalent to the number part of $\overline{\mathbf{VTC}}^0$ (Definition 3.6), and that $\mathbf{I\Delta}_0 +$ *counting* is a subtheory of $\mathbf{I}\mathcal{E}^2$. Similarly, it can be shown that $\mathbf{I}\mathcal{E}^2$ is equivalent to the number part of our theory $\overline{\mathbf{VL}}$ ($\mathbf{VL}$ is defined in Section 3.6, and for a general definition of $\overline{\mathbf{VC}}$ see Section 3.2.2). Now, both our formalizations in this chapter and the formalizations in [CD94] can be carried out in $\mathbf{I\Delta}_0 +$ *counting*.

In [Cor95] Bertrand's Postulate and a lower bound $\mathcal{O}(n/ln(n))$ for $\pi(2n)-\pi(n)$ have been formalized and proved in $\mathbf{I\Delta}_0(\pi, K)$, where

$$K(x) = \sum_{0<i\leq x} \log^*(i)$$

Here $\log^*(x)$ [Woo81] is the approximation, definable in $\mathbf{I\Delta}_0$, that approximates $\ln(x)$ within $1/(\log(m))^k$ for some constant $k \in \mathbb{N}$ and for some m. (Our approximation $\ln(x;m)$ approximates $\ln(x)$ upto $1/m$ but we need $\mathbf{VTC}^0$ to define it.) Since both K and π are definable in $\mathbf{I\Delta}_0 + counting$, the theory $\mathbf{I\Delta}_0(\pi, K)$ is a subtheory of $\mathbf{I\Delta}_0 + counting$. So the results of [Cor95] are stronger than our results here.

Chapter 9

Conclusion

A separation of any two classes in (1.1) would imply the separation of the corresponding theories **VC** given in Chapter 3. So if such separations are indeed the case, the latter is easier to prove and it might shed light on the former. Believing that an inclusion is proper, one might try first to separate the theories. (It is also consistent with our current knowledge that the theories are different but the classes coincide.) There is a hope that techniques from first-order logic and model theory are useful.

One topic that has not been discussed in this thesis is the Paris-Wilkie propositional translation of proofs in the theories into corresponding propositional proof systems. For example, $\mathbf{\Sigma}_0^B$ theorems of $\mathbf{TV}^0$ are translated into families of propositional tautologies having polynomial-size Extended Frege proofs, and Extended Frege is the strongest (up to polynomial simulation) propositional proof system whose soundness is provable in $\mathbf{TV}^0$. Similarly, for each theory **VC** discussed in Chapter 3 we can define a propositional proof system **C-Frege** which can be viewed as the nonuniform version of **VC**. The idea is to introduce some general connectives (e.g., the threshold connectives) that capture the complexity of the function $F_{\mathbf{C}}$ that is complete for **C** (e.g., *numones* for $\mathbf{TC}^0$). The details are being worked out.

The Bounded Reverse Mathematics program is to prove (the bounded versions of) mathematical theorems in (the weakest possible) theories of Bounded Arithmetic. A large number of interesting theorems, such as those from graph theory, can be listed here. Consider, for example, Hall's Theorem: Given a bipartite graph G with the bipartition (X, Y) of the nodes, Hall's Theorem states that, if for all subsets A of X,

$$\#N(A) \geq \#A$$

(where $\#A$ denotes the number of elements in the set A, and $N(A)$ denotes the set of neighbors of A, i.e., the set of vertices not in A which are adjacent to at least one vertex in A), then G has a perfect matching.

It is known that finding a perfect matching (if it exists) can be done in **RNC**. It might therefore be possible to prove the theorem in a theory that characterizes **RNC**. A plan for this direction is to develop a theory for **RNC**, and then try to prove Hall's Theorem in that theory.

An interesting question is whether it is possible to formalize in $\mathbf{VTC}^0$ the $\mathbf{TC}^0$ algorithm for integer division from [HAB02]. Here we have established in $\mathbf{VTC}^0$ facts about the distribution of prime numbers that are needed for the construction in [HAB02]. The next step is to see whether the relation

$$a^n \equiv b \mod p$$

can be formulated in $\mathbf{VTC}^0$. Although this relation is expressible by a $\mathbf{\Delta}_0$ formula (see [HAB02]), it is not clear how to prove in $\mathbf{VTC}^0$ that such formula is correct.

Bibliography

[ACN07] Klaus Aehlig, Stephen Cook, and Phuong Nguyen. Relativizing Small Complexity Classes and their Theories. In *16th EACSL Annual Conference on Computer Science and Logic*, pages 374–388, 2007.

[All91] Bill Allen. Arithmetizing uniform **NC**. *Annals of Pure and Applied Logic*, 53(1):1 – 50, 1991.

[Ara00] Toshiyasu Arai. Bounded arithmetic AID for Frege system. *Annals of Pure and Applied Logic*, 103:155–199, 2000.

[Bar89] David A. Barrington. Bounded-Width Polynomial-Size Branching Programs Recognizes Exactly Those Languages in $\mathbf{NC}^1$. *Journal of Computer and System Sciences*, 38:150–164, 1989.

[Ben62] James Bennett. *On Spectra*. PhD thesis, Princeton University, Departmentof Mathematics, 1962.

[BIS90] David A. Mix Barrington, Neil Immerman, and Howard Straubing. On Uniformity within $\mathbf{NC}^1$. *Journal of Computer and System Sciences*, 41:274–306, 1990.

[Bus86a] Jonathan Buss. Relativized Alternation. In *Proceedings, Structure in Complexity Theory Conference*. Springer-Verlag, 1986.

[Bus86b] Samuel Buss. *Bounded Arithmetic*. Bibliopolis, Naples, 1986.

[Bus87a] Samuel Buss. Polynomial Size Proofs of the Propositional Pigeonhole Principle. *Journal of Symbolic Logic*, 52:916–927, 1987.

[Bus87b] Samuel Buss. The Boolean formula value problem is in **Alogtime**. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, pages 123–131, 1987.

[Bus95] Samuel Buss. Relating the Bounded Arithmetic and Polynomial-Time Hierarchies. *Annals of Pure and Applied Logic*, 75:67–77, 1995.

[Bus98] Samuel Buss. First–Order Proof Theory of Arithmetic. In S. Buss, editor, *Handbook of Proof Theory*, pages 79–147. Elsevier, 1998.

[Bus06] Samuel Buss. Polynomial-size Frege and Resolution Proofs of st-Connectivity and Hex Tautologies. *Theorectical Computer Science*, 357:35–52, 2006.

[CD94] C. Cornaros and C. Dimitracopoulos. The Prime Number Theorem and Fragments of **PA**. *Archive for Mathematical Logic*, 33:265–281, 1994.

[CK02] Peter Clote and Evangelos Kranakis. *Boolean Functions and Computation Models*. Springer, 2002.

[CK03] Stephen Cook and Antonina Kolokolova. A Second-Order System for Polytime Reasoning Based on Grädel's Theorem. *Annals of Pure and Applied Logic*, pages 193–231, 2003.

[CK04] Stephen Cook and Antonina Kolokolova. A Second-Order Theory for **NL**. In *Logic in Computer Science (LICS)*, 2004.

[Clo90] Peter Clote. **Alogtime** and a Conjecture of S. A. Cook. In *Proceedings of IEEE Symposium on Logic in Computer Science*, 1990.

[Clo93] Peter Clote. On Polynomial Size Frege Proofs of Certain Combinatorial Principles. In Peter Clote and Jan Krajíček, editors, *Arithmetic, Proof Theory, and Computational Complexity*, pages 162–184. Oxford, 1993.

[CM05] Stephen Cook and Tsuyoshi Morioka. Quantified Propositional Calculus and a Second-Order Theory for $\mathbf{NC}^1$. *Archive for Mathematical Logic*, 44:711–749, 2005.

[CN06] Stephen Cook and Phuong Nguyen. Foundations of Proof Complexity: Bounded Arithmetic and Propositional Translations. Book in progress, 2006.

[Coo75] Stephen Cook. Feasibly Constructive Proofs and the Propositional Calculus. In *Proceedings of the 7th Annual ACM Symposium on the Theory of Computing*, pages 83–97, 1975.

[Coo85] Stephen Cook. A Taxonomy of Problems with Fast Parallel Algorithms. *Information and Control*, 64(1-3):2–21, 1985.

[Coo98] Stephen Cook. Relating the Provable Collapse of $\mathbf{P}$ to $\mathbf{NC}^1$ and the Power of Logical Theories. *DIMACS Series in Discrete Math. and Theoretical Computer Science*, 39, 1998.

[Coo02] Stephen Cook. Proof Complexity and Bounded Arithmetic. Course Notes for CSC 2429S. `http://www.cs.toronto.edu/~sacook/`, 2002.

[Coo05] Stephen Cook. Theories for Complexity Classes and Their Propositional Translations. In Jan Krajíček, editor, *Complexity of computations and proofs*, pages 175–227. Quaderni di Matematica, 2005.

[Coo07] Stephen Cook. Bounded Reverse Mathematics. Plenary Lecture for CiE 2007, 2007.

[Cor95] Ch. Cornaros. On Grzegorczyk Induction. *Annals of Pure and Applied Logic*, 74:1–21, 1995.

[CR97] Stephen Cook and Charles Rackoff. Unpublished research notes, 3 June, 1997.

[CT92] Peter Clote and Gaisi Takeuti. Bounded Arithmetic for **NC**, **Alogtime**, **L** and **NL**. *Annals of Pure and Applied Logic*, 56:73–117, 1992.

[CT95] Peter Clote and Gaisi Takeuti. First Order Bounded Arithmetic and Small Boolean Circuit Complexity Classes. In P. Clote and J. B. Remmel, editors, *Feasible Mathematics II*. Birkhäuser, 1995.

[D'A92] Paola D'Aquino. Local Behaviour of the Chebyshev Theorem in Models of $\mathbf{I\Delta}_0$. *Journal of Symbolic Logic*, 57:12–27, 1992.

[Grä92] Erich Grädel. Capturing Complexity Classes by Fragments of Second Order Logic. *Theorectical Computer Science*, 101:35–57, 1992.

[HAB02] William Hess, Eric Allender, and David A. Mix Barrington. Uniform Constant-Depth Threshold Circuits for Division and Iterated Multiplication. *Journal of Computer and System Sciences*, 65:695–716, 2002.

[Hal05] Thomas Hales. A Verified Proof of the Jordan Curve Theorem. Seminar Talk, Department of Mathematics, University of Toronto, 8 Dec, 2005.

[HP93] Petr Hájek and Pavel Pudlák. *Metamathematics of First-Order Arithmetic*. Springer–Verlag, 1993.

[Imm99] Neil Immerman. *Descriptive Complexity*. Springer, 1999.

[Joh96] Jan Johannsen. A Bounded Arithmetic Theory for Constant Depth Threshold Circuits. In Petr Hájek, editor, *GÖDEL '96. Springer Lecture Notes in Logic 6*, pages 224–234, 1996.

[Joh98] Jan Johannsen. Equational Calculi and Constant-Depth Propositional Proofs. In Paul Beame and Samuel Buss, editors, *Proof Complexity and Feasible Arithmetics*, volume 39, pages 149–162. AMS DIMACS Series, 1998.

[JP00] Jan Johannsen and Chris Pollett. On the Δ_1^b-Bit-Comprehension Rule. In Sam Buss, Petr Hájek and Pavel Pudlák, editor, *Logic Colloquium 98*, pages 262–279, 2000.

[Kol04] Antonina Kolokolova. *Systems of Bounded Arithmetic from Descriptive Complexity*. PhD thesis, University of Toronto, 2004.

[KPT91] Jan Krajíček, Pave Pudlák, and Gaisi Takeuti. Bounded Arithmetic and the Polynomial Hierarchy. *Annals of Pure and Applied Logic*, 52:143–153, 1991.

[Kra90] Jan Krajíček. Exponentiation and second-order bounded arithmetic. *Annals of Pure and Applied Logic*, 48:261–276, 1990.

[Kra95a] Jan Krajíček. *Bounded Arithmetic, Propositional Logic, and Complexity Theory*. Cambridge University Press, 1995.

[Kra95b] Jan Krajíček. On Frege and Extended Frege Proof Systems. In P. Clote and J. B. Remmel, editors, *Feasible Mathematics II*, pages 284–319. Birkhäuser, 1995.

[LL76] Richard Ladner and Nancy Lynch. Relativization of questions about log space computability. *Mathematical Systems Theory*, 10:19–32, 1976.

[Mos49] Leo Moser. A theorem on the distribution of primes. *American Mathematical Monthly*, 56(9):624–625, 1949.

[NC04] Phuong Nguyen and Stephen Cook. $\mathbf{VTC}^0$: A Second-Order Theory for $\mathbf{TC}^0$. In *Proc. 19th IEEE Symposium on Logic in Computer Science*, 2004.

[NC05] Phuong Nguyen and Stephen Cook. Theory for $\mathbf{TC}^0$ and Other Small Complexity Classes. *Logical Methods in Computer Science*, 2, 2005.

[NC07] Phuong Nguyen and Stephen Cook. The Complexity of Proving Discrete Jordan Curve Theorem. In *Proc. 22nd IEEE Symposium on Logic in Computer Science*, pages 245–254, 2007.

[Ngu04] Phuong Nguyen. VTC^0: A Second-Order Theory for TC^0. Master's thesis, University of Toronto, 2004. `http://www.cs.toronto.edu/~pnguyen/`.

[Ngu07] Phuong Nguyen. The Equivalence of Theories that Characterize **ALogTime**. (accepted to Archive for Mathematical Logic.) Available at `http://www.cs.toronto.edu/~pnguyen/`, 2007.

[Orp84] P. Orponen. General Nonrelativizability Results for Parallel Models of Computation. In *Proceedings, Winter School in Theoretical Computer Science*, pages 194–205. 1984.

[Par71] Rohit Parikh. Existence and feasibility in arithmetic. *Journal of Symbolic Logic*, 36(3):494–508, 1971.

[Per05] Steven Perron. $\mathbf{GL}^\star$: A Propositional Proof System For Logspace. Master's thesis, University of Toronto, 2005.

[Pit00] Francois Pitt. *A Quantifier-Free String Theory* **Alogtime** *Reasoning*. PhD thesis, University of Toronto, 2000.

[PW85] J. Paris and A. Wilkie. Counting Problems in Bounded Arithmetic. In A. Dold and B. Eckmann, editors, *Methods in Mathematical Logic*, pages 317–340. Springer–Verlag, 1985.

[Raz93] Alexander A. Razborov. An Equivalence between Second Order Bounded Domain Bounded Arithmetic and First Order Bounded Arithmetic. In Peter Clote and Jan Krajíček, editors, *Arithmetic, Proof Theory and Computational Complexity*, pages 247–277. Oxford, 1993.

[Raz95] Alexander A. Razborov. Bounded Arithmetic and Lower Bounds in Boolean Complexity. In P. Clote and J. B. Remmel, editors, *Feasible Mathematics II*, pages 344–386. Birkhäuser, 1995.

[RST84] Walter Ruzzo, Janos Simon, and Martin Tompa. Space-Bounded Hierarchies and Probabilistic Computations. *Journal of Computer and System Sciences*, 28(2):216–230, 1984.

[Sho07] Victor Shoup. *A Computational Introduction to Number Theory and Algebra.* Cambridge University Press, 2007.

[Sim77] Istvan Simon. *On some subrecursive reducibilities.* PhD thesis, Stanford University, 1977.

[Sim99] Stephen Simpson. *Subsystems of Second Order Arithmetic.* Springer, 1999.

[Tak93] Gaisi Takeuti. RSUV Isomorphism. In Peter Clote and Jan Krajíček, editors, *Arithmetic, Proof Theory and Computational Complexity*, pages 364–386. Oxford, 1993.

[Tho92] Carsten Thomassen. The Jordan-Schonflies Theorem and the Classification of Surfaces. *Amer. Math. Monthly*, 99(2):116–131, 1992.

[Wil88] Christopher Wilson. A Measure of Relativized Space Which Is Faithful with Respect to Depth. *Journal of Computer and System Sciences*, 36:303–312, 1988.

[Wil89] Christopher Wilson. Relativized **NC**. *Mathematical Systems Theory*, 20:13–29, 1989.

[Woo81] Alan Woods. *Some Problems in Logic and Number Theory and Their Connections*. PhD thesis, University of Manchester, 1981.

[Zam96] Domenico Zambella. Notes on Polynomially Bounded Arithmetic. *Journal of Symbolic Logic*, 61(3):942–966, 1996.

[Zam97] Domenico Zambella. End Extensions of Models of Linearly Bounded Arithmetic. *Annals of Pure and Applied Logic*, 88:263–277, 1997.

Index

A_3, 66

A_4, 66

$S(X)$, 54

S_3, 66

S_4, 66

S_5, 78, 80

$X \cdot Y$, 18

$\delta_{SinglePath}$, 50

δ_{parity}, 38, 67

$\delta_{\mathbf{MOD}_m}$, 38

$\pi(n)$, 112

$\varnothing$, 54, 61

$x \times y$, 2

$x^z = y$, 17

$\mathbf{\Delta}_1^b$-**CR**, 5

$\mathbf{\Pi}_i^B$, 16

$\mathbf{\Sigma}_0^B$-**Rec**, 51

$\mathbf{\Sigma}_i^B$, 16

$\mathbf{\Sigma}_0^B$, 16

$\mathbf{\Sigma}_0^B$-*TreeRec*, 40

δ_{CONN}, 48

δ_{LMCV}, 57

δ_{MCV}, 53

δ_{MFV}, 39

δ_{NUM}, 26

2-**BASIC**, 19

AC hierarchy, 16

- $\mathbf{AC}^0(2)$, 2, 7, 60, 61
- $\mathbf{AC}^0(6)$, 1, 7, 60, 65–67
- $\mathbf{AC}^0(m)$, 16
- $\mathbf{AC}^0(m)$, relativized, 85
- $\mathbf{AC}^0$, 2
- $\mathbf{AC}^k$, 57
- $\mathcal{L}_{\mathbf{FAC}^0}$, 23

$\mathbf{AC}^0$ reduction, 5, 18, 60, 68

aggregate function, 31

AID, 6, 39

alternating sets, 100

ALV$'$, 6, 73

approximate $\ln(x)$, 115

BASIC, **B1** – **B12**, 19

B12$'$, **B12**$''$, 22

Barrington's Theorem, 7, 78, 81

Bertrand's Postulate, 14, 112, 113, 126

bit graph, 17

bit recursion, **BIT-REC**, 54
boolean sentence value problem, 6, 78
bounded formula, 16
bounded number recursion, 60, 61
  3-BNR, 4-BNR, 65
  5-BNR, 72
bounded quantifier, 16
bounded recursion on notation, 73
Bounded Reverse Mathematics, 9, 131

Chebyshev's Theorem, 10, 13
classes, 16
comprehension, 19, 20
comprehension rule, 5
concatenation recursion on notation, 73
connectivity, 48, 49
conservative extension, 21
curve, 95
Cut, 29, 54

Definability Theorem, 28, 30
  for $\mathbf{VTC}^0$, 37
definable function, 21
  from $\mathcal{L}$, 18

extension
  conservative, 21
  universal, 22

$F^\star$,$f^\star$, 31
$F_{\varphi,t}$,$f_{\varphi,t}$, 23
$f_{\mathbf{SE}}$, 22
Fval, 40, 72
factoring, 17
Fanin2, 58
FC, 17, 28
Finite Model Theory, 6
formula, 16
Formula Value Problem, 39
function algebra, 7, 60
function class, 17

Grzegorczyk's class $\mathcal{E}^2$, 113

Hall's Theorem, 131
heap, 39
Horn formula, 6

$\mathbf{I\Delta}_0$, 112
$\mathbf{I\Delta}_0 + \mathit{counting}$, 113
$\mathbf{I}\mathcal{E}^2$, 113
induction, 19
  string induction, 54
integer division, 12, 131
isomorphism, 71

Jordan Curve Theorem, 10, 94

Krom formula, 6

L, 7, 16, 49, 60, 61

relativized, 85
$\mathcal{L}_A^2$, 15
layered circuit, 58
length function, 2, 15
$\log(x)$, 21
log time hierarchy, **LTH**, 17
logspace, **L**, 16

MCV,*Mcv*, 53
MFV, 40, 78, 83
minimization, 20
mod_m, 38
monotone circuit, 53
layered, 57
monotone formula (see also *MFV*), 39
multiple comprehension, 20, 21
multiplication function, 2

NC hierarchy, 16
$\mathbf{NC}^1$, 60
$\mathbf{NC}^k$, 57
relativized, 85
NL, 16, 48
relativized, 85
nondeterministic logspace, 16
number induction, 19
number minimization, 20
number quantifier, 16
number recursion, 60
number summation, 62
number term, 15, 16
number variable, 15
NUMONES, 26, 41
numones, 5, 26
$numones^\star$, 32, 37
in $\mathbf{VNC}^1$, 41

P, 16, 53
p-bounded function, 17
pairing, 20
Parikh's Theorem, 19
parity, 38, 67, 96
permutation, 63
Pigeonhole Principle, 114
polynomial time hierarchy, **PH**, 1
polynomial-bounded number recursion, 61
polynomially bounded function, 17
polynomially bounded theory, 19
polytime, 16
predecessor function, *pd*, 22
predicate calculus, **PK**, 1
Prime Number Theorem, 113
propositional proof system, 1
PV, 6

QALV, 6, 10, 73

quantifier, 16

recursion, 60

reduction, $\mathbf{AC}^0$, 18, 60, 68

reduction, $\mathbf{L}(\alpha)$, 89

relativized classes, 85

relativized theory, 7

Representation Theorem, $\mathbf{\Sigma}_0^B$, 17

Reverse Mathematics, 10

Row, 22

RSUV isomorphism, 71

$\mathbf{S}_2^i$, 1

SE, 19, 22

  $f_{\mathbf{SE}}$, 22

sequence of numbers, *seq*, 22

set variable, 15

sharply bounded minimization, 76

SinglePath, 49

Skolem function, 22

solvability, 65

st-connectivity, 11, 94, 110

string comprehension, 60, 67

string induction, 54

string quantifier, 16

string term, 15, 16

string variable, 15

summation, 62

$\mathbf{TAC}^k$, 7

$\mathbf{TC}^0$, 2, 16, 26, 60, 62

  $\mathbf{FTC}^0$, 26

  $\mathcal{L}_{\mathbf{FTC}^0}$, 27

  relativized, 85

term, 15, 16

$\mathbf{TNC}^k$, 7

tree recursion, 40

Trim, 75

$\mathbf{TV}^0$, 6, 54

two-sorted, 2

two-sorted class, 16

two-sorted logic, 15

uniformity, 16

universal conservative extension, 4, 22

$\mathbf{V}^0$, 4, 11, 19, 94

  seq, 22

  Row, 22

  $\overline{\mathbf{V}}^0$, 4, 22, 23

  and $\mathbf{I\Delta}_0$, 20

  finitely axiomatizable, 19

$\mathbf{V}^0(m)$, 38

$\mathbf{V}^0(2)$, 11

$\mathbf{VAC}^k$, 7, 57

VACC, 38

VALV, 71–73

variable, 15

VC, 4, 25, 28

$\overline{\mathbf{VC}}$, 4, 22, 29

application, 28

$\mathbf{V}^1$-**HORN**, 6

$\mathbf{V}^1$-**KROM**, 6

VL, 49

$\mathbf{VNC}^k$, 7, 57

$\mathbf{VNC}^1$, 6, 10, 39

$\subseteq$ **VL**, 51

RSUV with **QALV**, 71

VNL, 48

VP, 6, 53

$= \mathbf{TV}^0$, 54

$\mathbf{VTC}^0$, 5, 10, 13, 25, 72, 112

$\subseteq \mathbf{VNC}^1$, 41

$\overline{\mathbf{VTC}}^0$, 25, 27

Definability Theorem, 26, 37

word problem, S_5, 78