CHAPTER II

# First-Order Proof Theory of Arithmetic

Samuel R. Buss
*Departments of Mathematics and Computer Science, University of California, San Diego, La Jolla, CA 92093-0112, USA*

*Contents*



HANDBOOK OF PROOF THEORY
Edited by S. R. Buss

This chapter discusses the proof-theoretic foundations of the first-order theory of the non-negative integers. This first-order theory of numbers, also called 'first-order arithmetic', consists of the first-order sentences which are true about the integers. The study of first-order arithmetic is important for several reasons. Firstly, in the study of the foundation of mathematics, arithmetic and set theory are two of the most important first-order theories; indeed, the usual foundational development of mathematical structures begins with the integers as fundamental and from these constructs mathematical constructions such as the rationals and the reals. Secondly, the proof theory for arithmetic is highly developed and serves as a basis for proof-theoretic investigations of many stronger theories. Thirdly, there are intimate connections between subtheories of arithmetic and computational complexity; these connections go back to Gödel's discovery that the numeralwise representable functions of arithmetic theories are exactly the recursive functions and are recently of great interest because some weak theories of arithmetic have very close connection of feasible computational classes.

Because of Gödel's second incompleteness theorem that the theory of numbers is not recursive, there is no good proof theory for the complete theory of numbers; therefore, proof-theorists consider axiomatizable subtheories (called fragments) of first-order arithmetic. These fragments range in strength from the very weak theories $R$ and $Q$ up to the very strong theory of Peano arithmetic ($PA$).

The outline of this chapter is as follows. Firstly, we shall introduce the most important fragments of arithmetic and discuss their relative strengths and the bootstrapping process. Secondly, we give an overview of the incompleteness theorems. Thirdly, section 3 discusses the topics of what functions are provably total in various fragments of arithmetic and of the relative strengths of different fragments of arithmetic. Finally, we conclude with a proof of a theorem of J. Paris and A. Wilkie which improves Gödel's incompleteness theorem by showing that $I\Delta_0 + \exp$ cannot prove the consistency of $Q$. The main prerequisite for reading this chapter is knowledge of the sequent calculus and cut-elimination, as contained in Chapter I of this volume. The proof theory of arithmetic is a major subfield of logic and this chapter necessarily omits many important and central topics in the proof theory of arithmetic; the most notable omission is theories stronger than Peano arithmetic. Our emphasis has instead been on weak fragments of arithmetic and on finitary proof theory, especially on applications of the cut-elimination theorem. The articles of Fairtlough-Wainer, Pohlers, Troelstra and Avigad-Feferman in this volume also discuss the proof theory of arithmetic.

There are a number of book length treatments of the proof theory and model theory of arithmetic. Takeuti [1987], Girard [1987] and Schütte [1977] discuss the classical proof theory of arithmetic, Buss [1986] discusses the proof of the bounded arithmetic, and Kaye [1991] and Hájek and Pudlák [1993] treat the model theory of arithmetic. The last reference gives an in-depth and modern treatment both of classical fragments of Peano arithmetic and of bounded arithmetic.

## 1. Fragments of arithmetic

This section introduces the most commonly used axiomatizations for fragments of arithmetic. These axiomatizations are organized into the categories of 'strong fragments', 'weak fragments' and 'very weak fragments'. The line between strong and weak fragments is somewhat arbitrarily drawn between those theories which can prove the arithmetized version of the cut-elimination theorem and those which cannot; in practice, this is equivalent to whether the theory can prove that the superexponential function $i \mapsto 2^1_i$ is total. The very weak theories are theories which do not admit any induction axioms.

**Non-logical symbols for arithmetic.** We will be working exclusively with first-order theories of arithmetic: these have all the usual first-order symbols, including propositional connectives and quantifiers and the equality symbol $(=)$. In addition, they have *non-logical* symbols specific to arithmetic. These will always include the constant symbol $0$, the unary successor function $S$, the binary functions symbols $+$ and $\cdot$ for addition and multiplication, and the binary predicate symbol $\leq$ for 'less than or equal to'.[1] Very often, terms are abbreviated by omitting parentheses around the arguments of the successor function, and we write $St$ instead of $S(t)$. In addition, for $n \geq 0$ an integer, we write $S^n t$ to denote the term with $n$ applications of $S$ to $t$.

For weak theories of arithmetic, especially for bounded arithmetic, it is common to include further non-logical symbols. These include a unary function $\lfloor \frac{1}{2}x \rfloor$ for division by two, a unary function $|x|$ which is defined by

$$|n| \ = \ \lceil \log_2(n+1) \rceil,$$

and Nelson's binary function # (pronounced 'smash') which we define by

$$m\#n \ = \ 2^{|m|\cdot|n|}.$$

It is easy to check that $|n|$ is equal to the number of bits in the binary representation of $n$.

An alternative to the # function is the unary function $\omega_1$, which is defined by $\omega_1(n) = n^{\lfloor \log_2 n \rfloor}$ and has growth rate similar to #. The importance of the use of the $\omega_1$ function and the # function lies mainly in their growth rate. In this regard, they are essentially equivalent since $\omega_1(n) \approx n\#n$ and $m\#n = O(\omega_1(\max\{m,n\}))$. Both of these functions are generalizable to faster growing functions by defining $\omega_n(x) = x^{\omega_{n-1}(\lfloor \log_2 x \rfloor)}$ and $x\#_{n+1}y = 2^{|x|\#_n|y|}$ where $\#_2$ is #. It is easy to check that the growth rates of $\omega_n$ and $\#_{n+1}$ are equivalent in the sense that any term involving one of the function symbols can be bounded by a term involving the other function symbol.

[1]Many authors use $<$ instead of $\leq$; however, we prefer the use of $\leq$ since this sometimes makes axioms and theorems more elegant to state.

For strong theories of arithmetic, it is sometimes convenient to enlarge the set of non-logical symbols to include function symbols for all primitive recursive functions. The usual way to do this is to inductively define the primitive recursive functions as the smallest class of functions which contains the constant function 0 and the successor function $S$, is closed under a general form of composition, and is closed under primitive recursion. The closure under primitive recursion means that if $g$ and $h$ are primitive recursive functions of arities $n$ and $n+2$, then the $(n+1)$-ary function $f$ defined by

$$\begin{aligned} f(\vec{x},0) &= g(\vec{x}) \\ f(\vec{x},m+1) &= h(\vec{x},m,f(\vec{x},m)) \end{aligned}$$

is also primitive recursive. These equations are called the *defining equations* of $f$.

A *bounded* quantifier is of the form $(\forall x \le t)(\cdots)$ or $(\exists x \le t)(\cdots)$ where $t$ is a term not involving $x$. These may be used as abbreviations for $(\forall x)(x \le t \supset \cdots)$ and $(\exists x)(x \le t \wedge \cdots)$, respectively; or, alternatively, the syntax of first-order logic may be expanded to incorporate bounded quantifiers directly. In the latter case, the sequent calculus is enlarged with additional inference rules, shown in section 1.4. A usual quantifier is called an *unbounded* quantifier; when $|x|$ is in the language, a bounded quantifier of the form $(Qx \le |t|)$ is called a *sharply bounded* quantifier.

A theory is said to be *bounded* if it is axiomatizable with a set of bounded formulas. Since free variables in axioms are implicitly universally quantified, this is equivalent to being axiomatized with a set of $\Pi_1$-sentences (which are defined in section 1.2.1).

### 1.1. Very weak fragments of arithmetic

The most commonly used induction-free fragment of arithmetic is Robinson's theory $Q$, introduced by Tarski, Mostowski and Robinson [1953]. The theory $Q$ has non-logical symbols $0$, $S$, $+$ and $\cdot$ and is axiomatized by the following six axioms:

$$\begin{aligned} &(\forall x)(\neg Sx \neq 0) \\ &(\forall x)(\forall y)(Sx = Sy \supset x = y) \\ &(\forall x)(x \neq 0 \supset (\exists y)(Sy = x)) \\ &(\forall x)(x + 0 = x) \\ &(\forall x)(\forall y)(x + Sy = S(x+y)) \\ &(\forall x)(x \cdot 0 = 0) \\ &(\forall x)(x \cdot Sy = x \cdot y + x) \end{aligned}$$

Unlike most of the theories of arithmetic we consider, the language of $Q$ does not contain the inequality symbol; however, we can conservatively extend $Q$ to include $\le$ by giving it the defining axiom:

$$x \le y \;\leftrightarrow\; (\exists z)(x + z = y).$$

This conservative extension of $Q$ is denoted $Q_{\leq}$.

A yet weaker theory is the theory $R$, also introduced by Tarski, Mostowski and Robinson [1953]. This has the same language as $Q$ and is axiomatized by the following infinite set of axioms, where we let $s \leq t$ abbreviate $(\exists z)(s + z = t)$.

$$\begin{array}{ll}
S^m 0 \neq S^n 0 & \text{for all } 0 \leq m < n, \\
S^m 0 + S^n 0 = S^{m+n} 0 & \text{for all } m, n \geq 0, \\
S^m 0 \cdot S^n 0 = S^{m \cdot n} 0 & \text{for all } m, n \geq 0, \\
(\forall x)(x \leq S^m 0 \vee S^m 0 \leq x) & \text{for all } m \geq 0, \text{ and} \\
\multicolumn{2}{l}{(\forall x)(x \leq S^m 0 \leftrightarrow x = 0 \vee x = S0 \vee x = S^2 0 \vee \cdots \vee x = S^m 0)} \\
 & \text{for all } m \geq 0.
\end{array}$$

We leave it to the reader to prove that $Q \vDash R$.

### 1.2. Strong fragments of arithmetic

This section presents the definitions and the basic capabilities of some strong fragments of arithmetic. These fragments are defined by using induction axioms, minimization axioms or collection axioms; these axioms do not always apply to all first-order formulas, but rather apply to formulas that satisfy certain restrictions on quantifier alternation. For this purpose, we make the following definitions:

**1.2.1. Definition.** A formula is called a *bounded formula* if it contains only bounded quantifiers. The set of bounded formulas is denoted $\Delta_0$. For $n \geq 0$, the classes $\Sigma_n$ and $\Pi_n$ of first-order formulas are inductively defined by:

(1) $\Sigma_0 = \Pi_0 = \Delta_0$,

(2) $\Sigma_{n+1}$ is the set of formulas of the form $(\exists \vec{x})A$ where $A \in \Pi_n$ and $\vec{x}$ is a possibly empty vector of variables.

(3) $\Pi_{n+1}$ is the set of formulas of the form $(\exists \vec{x})A$ where $A \in \Sigma_n$ and $\vec{x}$ is a possibly empty vector of variables.

These classes $\Sigma_n$, $\Pi_n$ form the *arithmetic hierarchy.*

**1.2.2. Definition.** The *induction axioms* are specified as an axiom scheme; that is, if $\Phi$ is a set of formulas then the $\Phi$-IND axiom are the formulas

$$A(0) \wedge (\forall x)(A(x) \supset A(Sx)) \supset (\forall x)A(x),$$

for all formulas $A \in \Phi$. Note that $A(x)$ is permitted to have other free variables in addition to $x$. Similarly, the *least number principle* axioms or *minimization* axioms for $\phi$ are denoted $\Phi$-MIN and consist of all formulas

$$(\exists x)A(x) \supset (\exists x)(A(x) \wedge \neg(\exists y)(y < x \wedge A(y))),$$

for all $A \in \Phi$. Likewise, the *collection* or *replacement* axioms for $\Phi$ are denoted $\Phi$-REPL and consist of the formulas

$$(\forall x \le t)(\exists y)A(x,y) \supset (\exists z)(\forall x \le t)(\exists y \le z)A(x,y),$$

for all $A \in \Phi$.

**1.2.3. Definition.** The above axioms form the basis for a hierarchy of strong fragments of arithmetic over the language containing the non-logical symbols 0, $S$, $+$, $\cdot$ and $\le$. The theory $I\Sigma_n$ is defined to be the theory axiomatized by the eight axioms of $Q_\le$ plus the $\Sigma_n$-IND axioms. Of particular importance is the special case of the theory $I\Delta_0$ which defined as $Q_\le$ plus the $\Delta_0$-IND axioms. The theory $L\Sigma_n$ is defined to be the theory $I\Delta_0$ plus the $\Sigma_n$-MIN axioms. Similarly, $B\Sigma_n$ is the theory consisting of $I\Delta_0$ plus the $\Sigma_n$-REPL axioms. Other theories, especially $I\Pi_n$, $L\Pi_n$ and $B\Pi_n$, can be defined similarly.

The theory of *Peano arithmetic*, *PA*, is defined to be the theory $Q$ plus induction for all first-order formulas. The figure below shows that *PA* also admits the minimization and replacement axioms for all formulas.

**1.2.4.** The figure below shows the containments between the various strong fragments of arithmetic, where $S \Rightarrow T$ indicates that the theory $S$ logically implies the theory $T$. The two arrows $I\Sigma_{n+1} \Rightarrow B\Sigma_{n+1}$ and $B\Sigma_{n+1} \Rightarrow I\Sigma_n$ do not reverse, i.e., the containments are proper. These facts are due to Parsons [1970] and Paris and Kirby [1978]. (The figure is taken from the latter reference.)

$$\begin{array}{lllll}
I\Sigma_{n+1} & & & & \\
\Downarrow & & & & \\
B\Sigma_{n+1} & \Longleftrightarrow B\Pi_n & & & \\
\Downarrow & & & & \\
I\Sigma_n & \Longleftrightarrow I\Pi_n & \Longleftrightarrow L\Sigma_n & \Longleftrightarrow L\Pi_n &
\end{array}$$

Most of these containments are proved in section 1.2.9. The fact that $B\Sigma_{n+1}$ is a subtheory of $I\Sigma_{n+1}$ is proved as Theorem 1.2.9 below. The fact that it is a *proper* subtheory of $I\Sigma_{n+1}$ is proved as Theorem 3.4.2.

**1.2.5. $\Sigma_n^+$ and $\Pi_n^+$ formulas.** Some authors use a different definition of the arithmetic hierarchy than definition 1.2.1. These alternative classes, which we denote $\Sigma_n^+$ and $\Pi_n^+$, are inductively defined by

(1) $\Sigma_0^+ = \Pi_0^+ = \Delta_0$,

(2) $\Sigma_{n+1}^+$ is the set of formulas obtained by prepending an arbitrary block of existential quantifiers and bounded universal quantifiers to $\Pi_n^+$-formulas.

(3) $\Pi_{n+1}^+$ is the set of formulas obtained by prepending an arbitrary block of universal quantifiers and bounded existential quantifiers to $\Sigma_n^+$-formulas.

Thus $\Sigma_n^+$ and $\Pi_n^+$ are defined analogously to $\Sigma_n$ and $\Pi_n$, except arbitrary bounded quantifiers may be inserted without adding to the quantifier complexity.

It is straightforward to prove that $\Sigma_n$-REPL proves that every $\Sigma_n^+$-formula is equivalent to a $\Sigma_n$-formula, using induction on the number of unbounded quantifiers which are in the scope of a bounded quantifier, with a sub-induction on the number of bounded quantifiers which have the outermost unbounded quantifier in their scope. Therefore, $I\Sigma_n^+$ is equivalent to $I\Sigma_n$ and $B\Sigma_n^+$ is equivalent to $B\Sigma_n$.

### 1.2.6. Bootstrapping $I\Delta_0$, Phase 1

The axioms of $Q$ are very simplistic and, by themselves, do not imply many elementary facts about addition and multiplication, such as commutativity and associativity. When combined with induction axioms, however, the axioms of $Q$ imply many basic facts about the integers. The process of establishing such basic facts as commutativity and associativity of addition and multiplication, the transitivity of $\leq$, the totality of subtraction, etc. is called *bootstrapping*, named after the expression "to lift oneself by one's bootstraps". That is to say, in order to use the full power of a set of axioms, it is necessary to do some relatively tedious work establishing that the axioms of $Q$ are sufficient as a base theory.

This section will give a sketch of the bootstrapping process for $I\Delta_0$; to keep things brief, only an outline will be given, with most of the proofs left to the reader. Because $I\Delta_0$ is a subtheory of all the strong fragments defined above, this bootstrapping applies equally well to all of them.

To begin the bootstrapping process, show that the following formulas are $I\Delta_0$ provable.

(a) *Addition is commutative:* $(\forall x,y)(x+y=y+x)$. In order to prove this, first prove the formulas (a.1) $(\forall x)(0+x=x)$ and (a.2) $(\forall x,y)(Sx+y=S(x+y))$. In order to prove (a.1), use induction on the formula $0+x=x$, and to prove (a.2), use induction on $Sx+y=S(x+y)$ with respect to the variable $y$. Note that the variable $x$ is being used as a parameter in the latter induction. From these two, one can use induction on $x+y=y+x$ and prove the commutativity of addition.

(b) *Addition is associative:* $(\forall x,y,z)((x+y)+z=x+(y+z))$. Use induction on $(x+y)+z=x+(y+z)$.

(c) *Multiplication is commutative:* $(\forall x,y)(x\cdot y=y\cdot x)$. Analogously to (a), first prove (c.1) $0\cdot x=0$ and (c.2) $(Sx)\cdot y=x\cdot y+y$ by induction with respect to $x$ and $y$, respectively.

(d) *Distributive law:* $(\forall x,y,z)((x+y)\cdot z=x\cdot z+y\cdot z)$. Use induction.

(e) *Multiplication is associative:* $(\forall x,y,z)((x\cdot y)\cdot z=x\cdot(y\cdot z))$. Use induction.

(f) *Cancellation laws for addition:* $(\forall x,y,z)(x+z=y+z\leftrightarrow x=y)$ and $(\forall x,y,z)(x+z\leq y+z\leftrightarrow x\leq y)$. Use induction w.r.t. $z$ for the forward implications.

(g) *Discreteness of $\leq$:* $(\forall x,y)(x\leq Sy\supset x\leq y\lor x=Sy)$. This can be proved from $Q$ without any induction: if $x\leq Sy$, then $x+z=Sy$ for some $z$; either $z=0$ so $x=Sy$, or there is a $u$ such that $u=Sz$ and so $x+Su=Sy$, in which case $x+u=y$ and thus $x\leq y$.

(h) *Transitivity of $\leq$:* $(\forall x, y, z)(x \leq y \land y \leq z \supset x \leq y)$. Follows from $Q$ and the associativity of addition.

(i) *Anti-idempotency laws:* $(\forall x, y)(x + y = 0 \supset x = 0 \land y = 0)$ and $(\forall x, y)(x \cdot y = 0 \supset x = 0 \lor y = 0)$. These both follow from $Q$ without any induction. Use the fact that if $y \neq 0$, then $y = Sz$ for some $z$.

(j) *Reflexivity, trichotomy and antisymmetry of the $\leq$ ordering:* $(\forall x)(x \leq x)$, $(\forall x, y)(x \leq y \lor y \leq x)$ and $(\forall x, y)(x \leq y \land y \leq x \supset x = y)$. To prove trichotomy, use induction on $y$; the argument splits into two cases, depending on whether $x \leq y$ or $y + Sz = x$ for some $z$. To prove antisymmetry, reason as follows: if $x + u = y$ and $y + v = x$, then $x + u + v = x$, so by the cancellation law for addition, $u + v = 0$ and by anti-idempotency $u = v = 0$ and thus $x = y$.

(k) *Cancellation laws for multiplication:* $(\forall x, y, z)(z \neq 0 \land x \cdot z = y \cdot z \supset x = y)$ and $(\forall x, y, z)(z \neq 0 \land x \cdot z \leq y \cdot z \supset x \leq y)$. These can be proved using (j) to have $x = y$ or $x + Sv = y$ or $y + Sv = x$ for some $v$. Then use the distributive law, the anti-idempotency of multiplication, and the cancellation laws for addition.

(l) *Strict inequality:* $s < t$ is an abbreviation for $S(s) \leq t$. Thus, we can use bounded existential quantifiers $(\forall x < t)(\cdots)$ to mean $(\forall x \leq t)(x < t \supset \cdots)$, and similarly for bounded universal quantifiers.

**Theorem.** *$I\Delta \vdash \Delta_0$-MIN.*

**Proof.** The minimization axiom for $A(x)$ is easily seen to be equivalent to complete induction on $\neg A(x)$, namely to

$$(\forall y)(((\forall z < y)\neg A(z)) \supset \neg A(y)) \supset (\forall x)\neg A(x).$$

This is equivalent to induction on the bounded formula $(\forall y \leq x)(\neg A(y))$, and therefore is provable in $I\Delta_0$.

### 1.2.7. Extending the language of arithmetic.

We now introduce two useful means of conservatively extending the language of arithmetic with definitions of new predicate symbols and function symbols. It will be of particular importance that we can use the new function and predicate symbols in induction formulas.

**Definition.** A predicate symbol $R(\vec{x})$ is $\Delta_0$*-defined* if it has a defining axiom

$$R(\vec{x}) \leftrightarrow \phi(\vec{x})$$

with $\phi$ a $\Delta_0$-formula with all free variables as indicated.

The predicate $R$ is $\Delta_1$*-defined by a theory $T$* if there are $\Sigma_1$ formulas $\phi(\vec{x})$ and $\psi(\vec{x})$ such that $R$ has the defining axiom above and $T \vdash (\forall \vec{x})(\phi \leftrightarrow \neg\psi)$.

**Definition.** Let $T$ be a theory of arithmetic. A function symbol $f(\vec{x})$ is $\Sigma_1$-*defined by* $T$ if it has a defining axiom

$$y = f(\vec{x}) \quad \leftrightarrow \quad \phi(\vec{x}, y),$$

where $\phi$ is a $\Sigma_1$ formula with all free variables indicated, such that $T$ proves $(\forall \vec{x})(\exists! y)\phi(\vec{x}, y)$.[2]

The $\Sigma_1$-definable functions of a theory are sometimes referred to as the *provably recursive* or *provably total* functions of the theory. To see that this a reasonable definition for "provably recursive", let $M$ be a Turing machine which computes a function $y = M(x)$. Also choose some scheme for encoding computations of $M$ and let $T_M(x, w, y)$ be the predicate expressing "$w$ encodes a computation of $M$ on input $x$ which outputs $y$." From the bootstrapping below, it can be seen that the predicate $T_M$ can be a bounded formula. Therefore, the function computed by $M$ can be $\Sigma_1$-defined by the (true) formula $(\forall x)(\exists! y)(\exists w)(T_M(x, w, y))$. Conversely, for any true sentence $(\forall \vec{x})(\exists! y)\phi(\vec{x}, y)$ with $\phi$ a $\Sigma_1$-formula, the function mapping $\vec{x}$ to $y$ can be computed by a Turing machine that, given input values for $\vec{x}$, searches for a value for $y$ and for values of the existential quantified variables in $\phi$ which witness the truth of $(\exists y)\phi(\vec{x}, y)$.

In the case of $\Sigma_1$-definability of functions in $I\Delta_0$, it is possible to give a stronger equivalent condition; this is based on the following theorem of Parikh [1971]:

**1.2.7.1. Parikh's Theorem.** *Let $A(\vec{x}, y)$ be a bounded formula and $T$ be a bounded theory. Suppose $T \vdash (\forall \vec{x})(\exists y)A(\vec{x}, y)$. Then there is a term $t$ such that $T$ also proves $(\forall \vec{x})(\exists y \le t)A(\vec{x}, y)$.*

The above theorem is stated with $y$ a single variable, but it also holds for a vector of existentially quantified variables. A proof-theoretic proof of a generalization of this theorem is sketched in section 1.4.3 below.

It is easily seen that $I\Delta_0$ is a bounded theory, since the defining axiom for $\le$ may be replaced by the ($I\Delta_0$-provable) formula

$$x \le y \;\leftrightarrow\; (\exists z \le y)(x + z = y),$$

and the induction axioms may be replaced by the equivalent axioms

$$(\forall z)(A(0) \land (\forall x \le z)(A(x) \supset A(Sx)) \supset A(z)).$$

Thus applying Parikh's theorem gives the following theorem. Its proof is straightforward and we leave it to the reader.

[2]The notation $(\exists! y)$ means "there exists a unique $y$ such that $\cdots$". This is not part of the syntax of first-order logic; but is rather an abbreviation for a more complicated first-order formula.

**1.2.7.2. Theorem.** *A function symbol $f(\vec{x})$ is* $\Sigma_1$-defined by $I\Delta_0$ *if and only if it has a defining axiom*

$$y = f(\vec{x}) \quad \leftrightarrow \quad \phi(\vec{x}, y),$$

*and it has a* bounding term $t(\vec{x})$ *such that $\phi$ is a $\Delta_0$ formula with all free variables indicated and $I\Delta_0$ proves $(\forall\vec{x})(\exists! y \leq t)\phi(\vec{x}, y)$.*[3]

*A predicate symbol $R$ is $\Delta_1$-defined by $I\Delta_0$ if and only if it is $\Delta_0$-defined by $I\Delta_0$ (and furthermore, $I\Delta_0$ can prove the equivalence of the two definitions).*

The next theorem states the crucial fact about $\Sigma_1$-definable functions that they may be used freely without increasing the quantifier complexity of formulas, even when reexpressed in the original language of arithmetic.

**1.2.7.3. Theorem.**
(Gaifman and Dimitracopoulos [1982,Prop 2.3], see also Buss [1986,Thm 2.2].)

*(a) Let $T \supseteq B\Sigma_1$ be a theory of arithmetic. Let $T$ be extended to a theory $T^+$ in an enlarged language $L^+$ by adding $\Delta_1$-defined predicate symbols, $\Sigma_1$-defined function symbols and their defining equations. Then $T^+$ is conservative over $T$. Also, if $i \geq 1$ and if $A$ is a $\Sigma_i$- (respectively, $\Pi_i$-) formula in the enlarged language $L^+$, then there is a formula $A^-$ in the language of $T$ such that $A$ is also in $\Sigma_i$ (respectively, $\Pi_i$) and such that*

$$T^+ \vdash (A \leftrightarrow A^-).$$

*(b) The same results holds for $T \supseteq Q$ a bounded theory in the language $0, S, +, \cdot$ and $i = 0$, i.e., for the class $\Delta_0$.*

**Proof.** We first sketch the proof for part (b). The proof shows that the new symbols can be eliminated from a formula $A$ by induction on the number of occurrences of the new $\Delta_0$-defined predicate symbols and $\Sigma_1$-defined function symbols in $A$. Firstly, any atomic formula involving a $\Delta_0$-defined $R$ may just be replaced by the defining equation for $R$. Secondly, eliminate $\Sigma_1$-defined function symbols from terms in quantifier bounds, by replacing each bounded quantifier $(\forall x \leq t)(\cdots)$ by $(\forall x \leq t^*)(x \leq t \supset \cdots)$ where $t^*$ is obtained from $t$ by replacing every new $\Sigma_1$-defined function symbol with its bounding term; and by similarly replacing bounds on existential bounded quantifiers. Thirdly, repeatedly replace any atomic formula $P(f(\vec{s}))$ where $s$ does not involve any new function symbols by either of the formulas

$$(\exists z \leq t(\vec{s}))(A_f(\vec{s}, z) \wedge P(z))$$

or

$$(\forall z \leq t(\vec{s}))(A_f(\vec{s}, z) \supset P(z)),$$

where $A_f$ is the $\Delta_0$-formula which $\Sigma_1$-defined $f$, and $t$ is the bounding term of $f$.

It is easy to see that each step removes new function and predicate symbols from $A$ and preserves equivalence to $A$ and this proves (b).

---

[3]The notation $(\exists! y \leq t)(\cdots)$ means "there is a unique $y$ such that $\cdots$, and this $y$ is $\leq t$".

The proof of part (a) is similar, but needs a little modification. Most notably, there is no bounding term $t$, so the two formulas which can replace $P(f(\vec{s}))$ use an unbounded quantification of $z$ and are thus in $\Sigma_1$ and $\Pi_1$, respectively. Since $A_f$ is a $\Sigma_i$-formula, it is necessary to pick the correct one of the two formulas for replacing $P(f(\vec{s}))$: since the first is $\Sigma_1$ and the second is $\Pi_1$ there is always an appropriate choice that does not increase the alternation of unbounded quantifiers. Also, in order to remove new function symbols from the terms in bounded quantifiers, it is necessary to use the $\Sigma_1$-replacement axioms. We leave the details to the reader. □

As an immediate corollary to the previous theorem, we get the following important bootstrapping fact.

**Corollary.** *Let $T$ be $I\Delta_0$, $I\Sigma_n$ or $B\Sigma_n$ for $n \geq 1$. Then in the conservative extension of $T$ with $\Sigma_1$-defined function symbols and $\Delta_1$-defined function symbols, the new function and predicate symbols may be used freely in induction, minimization and replacement axioms.*

### 1.2.8. Bootstrapping $I\Delta_0$, Phase 2

To begin the second phase of the bootstrapping for $I\Delta_0$, several elementary functions and relations are shown to be $\Sigma_1$- and $\Delta_0$-definable in $I\Delta_0$.

(a) *Restricted subtraction.* The function $x \dot{-} y$ which equals $\max\{0, x-y\}$ can be $\Sigma_1$-defined by $I\Delta_0$ by the formula

$$M(x,y,z) \;\Leftrightarrow (y+z=x) \vee (x \leq y \wedge z=0).$$

The existence of $z$ follows immediately from the trichotomy of $\leq$; thus $I\Delta_0$ can prove $(\forall x,y)(\exists z \leq x)M(x,y,z)$. Furthermore, $I\Delta_0$ can prove the uniqueness of $z$ satisfying $M(x,y,z)$ using the cancellation law for addition. This then is a $\Sigma_1$-definition of the restricted subtraction function.

(b) *Predecessor.* The predecessor function is easily $\Sigma_1$-defined by $y = x \dot{-} 1$.

(c) *Division.* The division function $(x,y) \mapsto \lfloor x/y \rfloor$ can be $\Sigma_1$-defined by $\Delta_0$ using the formula

$$M(x,y,z) \;\Leftrightarrow\; (y \cdot z \leq x \wedge x < y(z+1)) \vee (y=0 \wedge z=0).$$

Note that in order to make the division function total, we have arbitrarily defined $x/0$ to equal $0$. The existence of $z$ is proved using induction on the formula $(\exists z \leq x)M(x,y,z)$. The uniqueness of $z$ is provable as follows, arguing inside $I\Delta_0$: suppose $M(x,y,z)$ and $M(x,y,z')$, w.l.o.g. $z \leq z'$; thus, using restricted subtraction and the distributive law, $(z' \dot{-} z)(y+1) < y+1$; and from this, $z'=z$ follows easily.

Two particularly useful special cases of division are when the divisor is two or four, i.e., $\lfloor \frac{1}{2}x \rfloor$ and $\lfloor \frac{1}{4}x \rfloor$.

(d) *Remainder.* The remainder function is $\Sigma_1$-definable in $I\Delta_0$ since $x \bmod y = x \dot{-} y \cdot \lfloor x/y \rfloor$. The *divides* relation, $x|y$, is defined by $y \bmod x = 0$.

(e) *Square root.* The square root function $x \mapsto \lfloor \sqrt{x} \rfloor$ is $\Sigma_1$-definable $I\Delta_0$ with the formula

$$M(x,y) \;\Leftrightarrow\; y \cdot y \le x \wedge x < (y+1)(y+1).$$

(f) *Primes.* The set of primes is $\Delta_0$-definable by the formula

$$(\forall y \le x)(y|x \supset y = x \vee y = 1) \wedge 1 < x.$$

$I\Delta_0$ can prove many useful facts about primes and remainders. In particular, it proves that if $x$ is prime and $x|ab$ then $x|a$ or $x|b$. The sequence coding tools developed below will enable $I\Delta_0$ to prove the unique factorization theorem; however, more bootstrapping is needed before we can even express the unique factorization theorem in first-order logic.

(g) *Prime powers.* The predicate "$x$ is prime and $y$ is a power of $x$" is $\Delta_0$-definable by

$$x \text{ is prime and } (\forall z \le y)(1 < z \wedge z|y \supset x|z).$$

$I\Delta_0$ can prove simple properties about prime powers, such as the fact that if $y$ is a power of the prime $x$ then $x \cdot y$ is the least power of $x$ greater than $y$. This fact can be proved by using $\Delta_0$-minimization with respect to $y$.

The bootstrapping is not yet sufficiently developed for us to give $\Delta_0$ definitions of powers of composite numbers; however, we shall next define powers of prime powers.

(h) *Powers of two, of four, and of prime powers.* We already have shown how to define powers of the prime two. For powers of fours, we can give two equivalent definitions:

$$y \text{ is a power of four} \;\Leftrightarrow\; y \text{ is a power of two and } y \bmod 3 = 1,$$

and

$$y \text{ is a power of four} \;\Leftrightarrow\; y \text{ is a power of two and } y = (\lfloor \sqrt{y} \rfloor)^2.$$

The equivalence of these definitions can be proved using $\Delta_0$-induction with respect to $y$. $I\Delta_0$ can also prove that when $y < y'$ are a powers of four, then $y|y'$ and that when $y$ is a power of four, then $4y$ is the least power of four greater then $y$.

More generally, the predicate "$x$ is a prime power and $y$ is a power of $x$" can be $\Delta_0$-defined by

$$(\exists p \le x)(p \text{ is prime, } x \text{ and } y \text{ are powers of } p, \text{ and } y \bmod (x-1) = 1).$$

(i) The *LenBit* function is defined so that $LenBit(2^i, x)$ is equal to the $i$-th bit in the binary expansion of $x$, where the least significant bit is by convention the zeroth bit. This is $\Sigma_1$-definable by $I\Delta_0$ since $LenBit(y,x) = \lfloor x/y \rfloor \bmod 2$. Although it is defined for all values $y$, we shall use $LenBit(y,x)$ only when $y$ is a power of two.

The next theorem states that $I\Delta_0$ can prove that the binary representation of a number uniquely determines the number. This theorem also introduces a new notation; namely, we will write quantifiers of the form $(\forall 2^i)$ and $(\exists 2^i)$ to mean, "for all powers of two" and "there exists a power of two." It is important to note that although we use this notation for quantifying over powers of two, we have not yet shown how to $\Delta_0$-define $i$ in terms of $2^i$.

**Theorem.** *$I\Delta_0$ proves* $(\forall x)(\forall y < x)(\exists 2^i \leq x)(LenBit(2^i, x) > LenBit(2^i, y))$.

To prove the theorem, $I\Delta_0$ uses strong induction with respect to $x$ and argues that if $2^i$ is the greatest power of two less than $x$, then $LenBit(2^i, x)$ equals one, and then, when $LenBit(2^i, y)$ also equals one, applies the induction hypothesis to $x \dot{-} 2^i$ and $y \dot{-} 2^i$. □

(j) We next show how to $\Delta_0$-define the relation $x = 2^y$ as a predicate of $x$ and $y$. As a preliminary step, we consider numbers of the form

$$m_p = \sum_{i=0}^{p} 2^{2^p}$$

for $p \geq 0$ and show that these numbers are $\Delta_0$-definable. In fact, the set $\{m_p\}_p$ is definable by the formula

$$\begin{aligned} &LenBit(1, x) = 0 \wedge LenBit(2, x) = 1 \wedge \\ &\quad \wedge (\forall 2^i \leq x)(2 < 2^i \supset \\ &\qquad [LenBit(2^i, x) = 1 \leftrightarrow (2^i \text{ is a power of } 4 \wedge LenBit(\left\lfloor \sqrt{2^i} \right\rfloor, x) = 1)]) \end{aligned}$$

As an immediate corollary we get a $\Delta_0$-formula defining the the numbers of the form $x = 2^{2^p}$; namely, they are the powers of two for which $LenBit(x, m_p) = 1$ holds for some $m_p < 2x$.

Now the general idea of defining $2^y$ is to express $y$ in binary notation as $y = 2^{p_1} + 2^{p_2} + \cdots + 2^{p_k}$ with distinct values $p_j$, and thus define $x = \prod_{j=1}^{k} 2^{2^{p_j}}$. To carry this out, we define an extraction function $Ext(u, x)$ which will be applied when $u$ is of a number of the form $2^{2^p}$. Formally we define

$$Ext(u, x) = \lfloor x/u \rfloor \bmod u.$$

Note that when $u = 2^{2^p}$, then $Ext(u, x)$ returns the number with binary expansion equal to the $2^p$-th bit through the $(2^{p+1} - 1)$-th bit of $x$. We will think of $x$ coding the sequence of numbers $Ext(2^{2^p}, x)$ for $p = 0, 1, 2, \ldots$. We also define $Ext'(u, x)$ as $Ext(u^2, x)$; this is of course the number which succeeds $Ext(u, x)$ in the sequence of numbers coded by $x$.

We are now ready to $\Delta_0$-define $x = 2^i$. We define it with a formula $\phi(x, i)$ which states there are numbers $a, b, c, d \leq x^2$ such that the following hold:

(1) $a$ is of the form $m_p$ and $a > x$.
(2) $Ext(2, b) = 1$, $Ext(2, c) = 0$ and $Ext(2, d) = 1$.
(3) For all $u$ of the form $2^{2^p}$ such that $a > u^2$, $Ext'(u, b) = 2 \cdot Ext(u, b)$,
(4) For all $u$ of the form $2^{2^p}$ such that $a > u^2$, either
  (a) $Ext'(u, c) = Ext(u, c)$ and $Ext'(u, d) = Ext(u, d)$, or
  (b) $Ext'(u, c) = Ext(u, c) + Ext(u, b)$ and $Ext'(u, d) = Ext(u, d) \cdot Ext(u, a)$.
(5) There is a $u < a$ of the form $2^{2^p}$ such that $Ext(u, c) = i$ and $Ext(u, d) = x$.

Obviously this is a $\Delta_0$-formula; we leave it to the reader the nontrivial task of checking that $I\Delta_0$ can prove simple facts about this definition of $2^i$, including (1) the fact that if $\phi(x,i)$ and $\phi(x,j)$ both hold, then $i=j$, (2) the fact that $2^i\cdot 2^j = 2^{i+j}$, (3) the fact that if $x$ is a power of two, then $x=2^i$ for some $i<x$, and (4) that if $\phi(x,i)$ and $\phi(y,i)$ then $x=y$.

(k) *Length function.* The length function is $|x| = \lceil \log_2(x+1)\rceil$ and can be $\Delta_0$-defined in $I\Delta_0$ as the value $i$ such that $y=2^i$ is the least power of two greater than $x$. Note that $|0|=0$ and for $x>0$, $|x|$ is the number of bits in the binary representation of $x$.

The reader should check that $I\Delta_0$ can prove elementary facts about the $|x|$ function, including that $|x|\le x$ and that $36<x \supset |x|^2<x$.

(l) The $Bit(i,x)$ function is definable as $LenBit(2^i,x)$. This is equivalent to a $\Delta_0$-definition, since when $2^i>x$, $Bit(i,x)=0$. $Bit(i,x)$ is the $i$-th bit of the binary representation of $x$; by convention, the lowest order bit is bit number $0$.

(m) *Sequence coding.* Sequences will be coded in the base 4 representation used by Buss [1986]; many prior works have used similar encodings. A number $x$ is viewed as a bit string in which pairs of bits code one of the three symbols comma, "0" or "1". The $i$-th symbol from the end is coded by the two bits $Bit(2i+1,x)$ and $Bit(2i,x)$. This is best illustrated by an example: consider the sequence $\langle 3,0,4\rangle$. Firstly, a comma is prepended to the the sequence and the entries are written in base two, preserving the commas, as the string: ",11,,100"; leading zeros are optionally dropped in this process. Secondly, each symbol in the string is replaced by a two bit encoding by replacing each "1" with "11", each "0" with "10", and each comma with "01". This yields "0111110101111010" in our example. Thirdly, the result is interpreted as a binary representation of a number; in our example it is the integer 32122. This then is a Gödel number of the sequence $\langle 3,0,4\rangle$.

This scheme for encoding sequences has the advantage of being relatively efficient from an information theoretic point of view and of making it easy to manipulate sequences. It does have the minor drawbacks that not every number is the Gödel number of a sequence and that Gödel numbers of sequences are non-unique since it is allowable that elements of the sequence be coded with excess leading zeros.

Towards arithmetizing Gödel numbers, we define predicates $Comma(i,x)$ and $Digit(i,x)$ as

$$\begin{aligned} Comma(i,x) &\Leftrightarrow Bit(2i+1,x)=0 \land Bit(2i,x)=1 \\ Digit(i,x) &= 2\cdot(1 \dot{-} Bit(2i+1,x)) + Bit(2i,x). \end{aligned}$$

Note $Digit$ equals zero or one for encodings of "0" and "1" and equals 2 or 3 otherwise.

It is now fairly simple to recognize and extract values from a sequence's Gödel number. We define $IsEntry(i,j,x)$ as

$$(i=0 \lor Comma(i\dot{-}1,x)) \land Comma(j,x) \land (\forall k<j)(k\ge i \supset Digit(j,x)\le 1)$$

which states that the $i$-th through $(j-1)$-st symbols coded by $x$ code an entry in the sequence. And we define $\mathit{Entry}(i,j,x) = y$ by

$$|y| \le j \dot{-} i \wedge (\forall k < j \dot{-} i)(\mathit{Bit}(k,y) = \mathit{Digit}(i+k,x)).$$

When $\mathit{IsEntry}(i,j,x)$ is true, then $\mathit{Entry}(i,j,x)$ equals the value of that entry in the sequence coded by $x$. Checking that *Entry* is $\Delta_0$-definable by $I\Delta_0$ is left to the reader; note that the quantifier $(\forall k < j \dot{-} i)$ may be replaced by a sharply bounded quantifier since, w.l.o.g., $j \le |x|$.

(n) *Length-bounded counting* and *Numones*. Although we have defined *Entry* already, we are not quite done with arithmetizing sequence coding; in particular, we would like to define the Gödel beta function, $\beta(i,x)$, which equals the $i$-th entry of the sequence coded by $x$. One way to do this would be by encoding a sequence of numbers $\langle a_n, a_{n-1}, \ldots, a_1\rangle$ as the sequence $\langle b_n, \ldots, b_1\rangle$ where $b_i = \langle i, a_i\rangle$. The drawback of this approach is that when the values $a_i$ are small, the length of the Gödel number encoding the sequence $\langle \vec{b}\rangle$ is longer than the length of the Gödel number encoding the sequence $\langle \vec{a}\rangle$; in fact, it is longer by a logarithmic factor and thus the function $\langle \vec{a}\rangle \mapsto \langle \vec{b}\rangle$ cannot be $\Delta_0$-defined by $I\Delta_0$ by virtue of the function's superlinear growth rate.

Upon reflection, one sees that the basic difficulty in defining the $\beta$ function is the difficulty of counting the number of commas encoded in a Gödel number of a sequence. This basically the same as the problem as counting of ones in the binary representation of a number $x$. Supposing $x$ has binary representation $(x_n x_{n-1} \cdots x_1 x_0)_2$, we would like to be able to let $a_0 = x_0$ and $a_i = a_{i-1} + x_i$ and then let $b_i = \langle i, a_i\rangle$ and finally let $y$ be the Gödel number of the sequence $\langle b_n, \ldots, b_1\rangle$. Now, $I\Delta_0$ can prove that, if $y$ exists, it is unique, and from $y$ the number of 1's in the binary representation of $x$ is easily computed. The catch is that, as above, $\langle \vec{b}\rangle$ will in general not be bounded by a term involving $x$ since its length is not necessarily $O(|x|)$. However, the length of the Gödel number of $\langle \vec{b}\rangle$ is $O(|x|^2)$ so this this method does work when $x$ is small; in particular, it works if $x = |y|$ for some $y$. Thus, $I\Delta_0$ can $\Sigma_1$-define the function *LenNumones* defined so that

$$\mathit{LenNumones}(y) \;=\; \text{the number of 1's in the binary representation of } |y|.$$

To define a *Numones* function that works for all numbers, we use a trick that allows more efficient encoding of successive numbers. The basic idea is that a sequence $a_1, a_2, a_3, \ldots, a_k$ of numbers can be encoded with fewer bits if, when writing the number $a_{i+1}$, one only writes the bits of $a_{i+1}$ which are different from the corresponding bits in $a_i$. This works particularly well when we have $a_i \le a_{i+1} \le a_i + 1$ for all $i$; in this case we formally define the succinct encoding as follows: for $i > 0$, define $i^*$ to be the greatest power of 2 which divides $i$; and define $0^*$ to equal $0$. Now define $a'_0 = a_0$ and define $a'_i$ to be $a^*_i$ if $a_i \ne a_{i-1}$ or to be 0 otherwise. (Example: if $a = 24$, then $a' = 8$.) Then the sequence $\vec{a}$ can be more succinctly described by $\vec{a'}$.

It is now important to see that $I\Delta_0$ can extract the sequence $\langle \vec{a}\rangle$ from the sequence $\langle \vec{a'}\rangle$, at least in a certain limited sense. In particular, we have that if $x = \langle a'_k, \ldots, a'_1, a'_0\rangle$ and if $\mathit{IsEntry}(i,j,x)$ and if this entry is the entry for $a'_\ell$, then the value $a_\ell$ can be $\Sigma_1$-defined in terms of $i, j, x$. To describe this $\Sigma_1$-definition, note that the $k$-th bit of the binary representation of $a_\ell$ is computed by finding the maximum values $i_0 < j_0 \le i$ such that $IsEntry(i_0, j_0, x)$ and such that $|Entry(i_0, j_0, x)| > k$ and letting the $k$-th bit of $a_\ell$ equal the $k$-th bit of $Entry(i_0, j_0, x)$.

The whole point of using $\langle \vec{a'} \rangle$ is to give a sufficiently succinct encoding of $\langle \vec{a} \rangle$. Of course, the fact that the encoding is sufficiently succinct also needs to be provable in $I\Delta_0$. It is easily checked that the Gödel number of the sequence $\langle 0^*, 1^*, \ldots, x^* \rangle$ uses exactly $6x - 2Numones(x) + 2$ many bits; this is proved by first showing that there are $2x - Numones(x)$ bits in the numbers in the sequence, i.e., $\sum_{i=0}^{x} |i^*| = 2x - Numones(x)$, and second noting that there are $x+1$ commas, and noting that each bit and comma is encoded by two bits in the Gödel number. Furthermore, when $x = |y|$ for some $y$, $I\Delta_0$ can prove this fact, using *LenNumones* in place of *Numones*.

We are now able to $\Sigma_1$-define the function $Numones(x)$ equal to the number of 1's in the binary representation of $x$. This is done by defining the sequence

$$u = \langle \langle k^*, a'_k \rangle, \langle (k-1)^*, a'_{k-1} \rangle, \ldots, \langle 0, a'_0 \rangle \rangle$$

such that $k = |x|$, $a_0 = 0$, and each $a_{i+1}$ is equal to $a_i + Bit(i, x)$. By the considerations in the previous paragraph, $I\Delta_0$ can prove that this sequence is bounded by a term involving only $x$; also, $I\Delta_0$ can compute the values of $0, \ldots, k$ from $0^*, \ldots, k^*$ and therefore can compute the values of $a_i$ as a function of $i$ and $u$.

(o) *Sequence coding.* Once we have the *Numones* function, it is an easy matter to define the Gödel $\beta$ function by counting commas. The $\beta$ function is defined so that $\beta(m, x) = a_m$ provided $x$ is the Gödel number of a sequence $\langle a_1, \ldots, a_k \rangle$ with $m \le k$. It is also useful to define the length function $Len(x)$ which equals $k$ when $x$ is as above. These are defined easily in terms of the *Numones* function: the value $\beta(m, x)$ equals $Entry(i, j, x)$ where there are $m - 1$ commas encoded in $x$ to the left of bit $i$; and $Len(x)$ equals the number commas coded by $x$.

Once sequence encoding has been achieved, the rest of the bootstrapping process is fairly straightforward. The next stage in bootstrapping is to arithmetize metamathematics, and this is postponed until section 2 below. Stronger theories, such as $I\Sigma_1$, can define all primitive recursive functions: this is discussed in section 1.2.10.

### 1.2.9. Relationships amongst the axioms of *PA*

We are now ready to sketch the proofs of the relationships between the various fragments of Peano arithmetic pictured in paragraph 1.2.4 above.

**Theorem.** *Let* $n \ge 0$.
(a) $B\Pi_n \vDash B\Sigma_{n+1}$.
(b) $I\Sigma_{n+1} \vDash B\Sigma_{n+1}$.
(c) *If* $A(x, \vec{w}) \in \Sigma_n$ *and* $t$ *is a term, then* $B\Sigma_n$ *can prove that* $(\forall x \le t)A(x, \vec{w})$ *is equivalent to a* $\Sigma_n$*-formula.*

**Proof.** To prove (a), suppose $A(x,y)$ is a formula in $\Sigma_{n+1}$. We want to show that

$$(\forall x \le u)(\exists y)A(x,y) \supset (\exists v)(\forall x \le u)(\exists y \le v)A(x,y).$$

is a consequence of $B\Pi_n$. This is proved by the following trick: take all leading existential quantifiers $(\exists \vec{z})$ from the beginning of $A$ and replace these quantifiers and the existential quantifier $(\exists y)$ by a single existential quantifier $(\exists w)$ which is intended to range over Gödel numbers of sequences coding values for all of the variables $y$ and $\vec{z}$, say by letting $\beta(1,w) = y$ and letting $\beta(i+1,w)$ be a value for the $i$-th variable in $\vec{z}$. Since $y = \beta(1,w) < w$, it follows that the collection axiom for this new formula implies the collection axiom for $A$.

Part (c) is proved by induction on $n$. Note that (c) is obvious when $n = 0$. For $n > 0$, (c) is proved by noting that, by using a sequence to code multiple values, we may assume without loss of generality that there is only one (unbounded) existential quantifier at the front of $A$, so $A$ is $(\exists y)B$ with $B \in \Pi_{n-1}$. Then $(\forall x \le t)A$ is equivalent to $(\exists u)(\forall x \le t)(\exists y \le u)B$; and finally by using the induction hypothesis that (c) holds for $n-1$, we have that $(\exists y \le u)B$ is equivalent to a $\Pi_{n-1}$-formula. From this $(\exists u)(\forall x \le t)A$ is equivalent to a $\Sigma_n$-formula.

We prove (b) by induction on $n$. Suppose $A(x,y)$ is a $\Sigma_{n+1}$-formula, possibly containing other free variables. We need to show that $I\Sigma_{n+1}$ proves the formula displayed above, and by part (a) we may assume that $A$ is a $\Pi_n$-formula. We argue informally inside $I\Sigma_{n+1}$, assuming that $(\forall x \le u)(\exists y)A(x,y)$ holds. Let $\phi(a)$ be the formula

$$(\exists v)(\forall x \le a)(\exists y \le v)A(x,y).$$

It follows from our assumption that $\phi(0)$ and that $\phi(a) \supset \phi(a+1)$ for all $a < u$. The induction hypothesis that $I\Sigma_n \vDash B\Sigma_n$ together with part (c) implies that the formula $(\forall x \le u)(\exists y \le v)A$ is equivalent to a $\Pi_{n-1}$-formula and thus $\phi$ is equivalent to a $\Sigma_n$-formula. Therefore, by induction on $\phi$, $\phi(u)$ holds; this is what we needed to show. $\square$

With the aid of the above theorem, the other relationships between fragments of Peano arithmetic are relatively easy to prove. To prove that $I\Sigma_n$ implies $\Pi_n$-IND, let $A(x)$ be a $\Pi_n$ formula and argue informally inside $I\Sigma_n$ assuming $A(0)$ and $(\forall x)(A(x) \supset A(x+1))$. Letting $a$ be arbitrary, and letting $B(x)$ be the formula $\neg A(a \dot{-} x)$, one has $\neg B(a)$ and $B(x) \supset B(x+1)$. Thus, by induction, $\neg B(0)$, and this is equivalent to $A(a)$. Since $a$ was arbitrary, $(\forall x)A(x)$ follows. A similar argument shows that $I\Pi_n$ implies $\Sigma_n$-IND.

To show that the $\Sigma_n$-MIN axioms are consequences of $I\Sigma_n$, note that by the argument given at the end of section 1.2.6 above, the minimization axiom for $A(x)$ follows from induction on the formula $(\forall x \le y)\neg A(x)$ with respect to the variable $y$. If $A \in \Sigma_n$, then from part (c) of the above theorem, the formula $(\forall x \le y)(\neg A)$ is equivalent to a $\Pi_n$-formula, so the minimization axiom for $A$ is a consequence of $I\Pi_n = I\Sigma_n$.

It is easy to derive the induction axiom for $A$ from the minimization axiom for $A$, so $L\Sigma_n = L\Pi_n = I\Sigma_n = I\Pi_n$.

Finally, the theorem of Clote [1985] that the strong $\Sigma_n$-replacement axioms are consequences of $I\Sigma_n$ can be proved as follows. Assume $n \geq 1$ and $A \in \Sigma_n$ and consider the strong replacement axiom

$$(\exists w)(\forall x \leq a)[(\exists y)A(x,y) \leftrightarrow A(x, \beta(x+1, w))].$$

for $A$. (Note $A$ may have free variables other than $x, y$.) Let $Num_A(u)$ be a $\Sigma_n$-formula which expresses the property that there exists a $w$ for which $A(x, \beta(x+1, w))$ holds for at least $u$ many values of $x \leq a$. Clearly $Num_A(0)$ holds and $Num_A(a+2)$ fails. So, by $\Sigma_n$-maximization (which follows easily from $\Sigma_n$-minimization), there is a maximum value $u_0 \leq a+1$ for which $Num_A(u_0)$ holds. A value $w$ that works for this $u_0$ satisfies the strong replacement axioms for $A$.

### 1.2.10. Definable functions of $I\Sigma_n$.

When bootstrapping theories stronger than $I\Delta_0$, such as $I\Sigma_n$ for $n > 0$, the main theorem of section 1.2.7 still applies, and $\Delta_1$-definable predicates and $\Sigma_1$-definable functions may be introduced into the language of arithmetic and used freely in induction axioms, without increasing the strength of the theory. Of particular importance is the fact that the primitive recursive functions can be $\Sigma_1$-defined in (any theory containing) $I\Sigma_1$.

**Definition.** The *primitive recursive* functions are functions on $\mathbb{N}$ and are inductively defined as follows:

(1) The constant function with value 0 is primitive recursive. We can view this a nullary function.

(2) The unary successor function $S(x) = x+1$ is primitive recursive.

(3) For each $1 \leq k \leq n$, then $n$-ary projection function $\pi^n_k(x_1, \ldots, x_n) = x_k$ is primitive recursive.

(4) If $g$ is an $n$-ary primitive recursive function and $h_1, \ldots, h_n$ are $m$-ary primitive recursive functions, then the $m$-ary function $f$ defined by $f(\vec{x}) = g(h_1(\vec{x}), \ldots h_n(\vec{x}))$ is primitive recursive.

(5) If $n \geq 1$ and $g$ is an $(n-1)$-ary primitive recursive function and $h$ is an $(n+1)$-ary primitive recursive function, then the $n$-ary function $f$ defined by:

$$\begin{aligned} f(0, \vec{x}) &= g(\vec{x}) \\ f(m+1, \vec{x}) &= h(m, f(m, \vec{x}), \vec{x}) \end{aligned}$$

is primitive recursive.

The only use of the projection functions is as a technical tool to allow generalized substitutions with case (4) above.

A predicate is *primitive recursive* if its characteristic function is primitive recursive.

**Theorem.** *$I\Sigma_1$ can $\Sigma_1$-define the primitive recursive functions.*

The converse to this theorem holds as well; namely, $I\Sigma_1$ can $\Sigma_1$-define exactly the primitive recursive functions. This converse is proved later as Theorem 3.1.1.

**Proof.** It is obvious that the base functions, zero, successor and projection, are $\Sigma_1$-definable in $I\Sigma_1$. It is easy to check that set of functions $\Sigma_1$-definable by $I\Sigma_1$ is closed under composition. Finally suppose that $g$ and $h$ are $I\Sigma_1$-definable in $I\Sigma_1$. Then, the function $f$ defined from $g$ and $h$ by primitive recursion can be $\Sigma_1$-defined with the following formula expressing $f(m, \vec{x}) = y$:

$$(\exists w)[Len(w) = m + 1 \wedge y = \beta(m+1, w) \wedge \beta(1, w) = g(\vec{x}) \wedge \\ (\forall i < m)(\beta(i+2, w) = h(i, \beta(i+1, w), \vec{x}))].$$

This formula expresses the condition that there is a sequence, coded by $w$, containing all the values $f(0, x), \ldots, f(m, \vec{x})$, such that each value in the sequence is correctly computed from the preceding value and such that the final value is $y$. The theorem of section 1.2.7 shows that the above formula defining $f$ is (equivalent to) a $\Sigma_1$-formula. We leave it to the reader to check that $I\Sigma_1$ can prove the requisite existence and uniqueness conditions for this definition of $f$. □

As an easy consequence, we have

**Corollary.** *Every primitive recursive predicate is $\Delta_1$-definable by $I\Sigma_1$.*

In the theories $I\Sigma_n$ with $n > 1$, even more functions are $\Sigma_1$-definable. A characterization of the functions $\Sigma_1$-definable in $I\Sigma_n$ is given in Chapter III of this volume. Other proof-theoretic characterizations of these functions can be found in Takeuti [1987], Buss [1994] and in references cited therein.

### 1.3. Fragments of bounded arithmetic

A subtheory of Peano arithmetic is called a bounded theory of arithmetic, or a theory of *bounded arithmetic*, if it is axiomatized by $\Pi_1$-formulas. The potential strength of such theories depends partly on the growth rates of the function symbols in the language, and usually bounded arithmetic theories have only functions of subexponential growth rate, including addition, multiplication and possibly polynomial growth rate functions such as $\omega_1$ or $\#$. These theories are typically weaker than the strong theories considered in section 1.2, but stronger than the theories $Q$ and $R$ discussed in section 1.1.

There are two principal approaches to bounded arithmetic. The original approach involved theories such as $I\Delta_0$ and $I\Delta_0 + \Omega_1$; more recently, bounded theories such as $S_2^i$ and $T_2^i$ have been extensively studied. One of the main motivations for studying bounded arithmetics is their close connection to low-level computational complexity, especially regarding questions relating expressibility and provability in bounded arithmetics to questions about the linear time hierarchy and the polynomial time hierarchy.

### 1.3.1. $I\Delta_0$ and $\Omega_n$

We have already defined $I\Delta_0$ and described its bootstrapping process in fairly complete detail in section 1.2. One corollary of the bootstrapping process is that the graph of exponentiation is $\Delta_0$-definable in $I\Delta_0$; that is to say, there is a bounded formula $exp(x,y,z)$ which expresses the condition $x^y = z$ and such that $I\Delta_0$ can prove facts like $exp(x,0,1)$, $exp(x,1,x)$,

$$exp(x,y,z) \wedge exp(x,y',z') \Rightarrow exp(x,y+y',z\cdot z')$$

and that for any $x$ and $y$, there is at most one $z$ such that $exp(x,y,z)$. The underlying idea of the $\Delta_0$-definition of $exp(x,y,z)$ is to define the sequence $\langle x^{\lfloor y/2^i \rfloor}\rangle_i$ where $i$ ranges from $|y|$ down to $0$; however, we leave it to the reader to supply the details behind this $\Delta_0$-definition. The fact that exponentiation is $\Delta_0$-definable is essentially due to Bennett [1962] and was first (and independently) proved in the setting of $I\Delta_0$ by Gaifman and Dimitracopoulos [1982].

Once the graph of exponentiation has been shown to be $\Delta_0$-definable, one can formulate the axioms $\Omega_k$. Firstly, when working in bounded arithmetic, we define $\log x$ to equal the greatest $y$ such that $2^y \le x$. Then the function $\omega_1(x,y)$ is defined to equal $x^{\log y}$. Since $|\omega_1(x,y)| = \Theta(|x|\cdot|y|)$, it is evident $\omega_1(x,y)$ cannot be bounded by a polynomial of $x$ and $y$. Therefore, by Parikh's Theorem 1.2.7.1, the function $\omega_1$ is not $\Sigma_1$-definable in $I\Delta_0$. As we shall see later, it is often very desirable to have the $\omega_1$ function be total; therefore it is common to extend $I\Delta_0$ to a stronger theory containing the axiom

$$\Omega_1: \qquad (\forall x)(\forall y)(\exists z)(z = \omega_1(x,y)).$$

This stronger theory is called $I\Delta_0 + \Omega_1$.

The function $\omega_1$ has what is called *polynomial growth rate*, i.e., for any term $t(\vec{a})$ constructed with the functions $S$, $+$, $\cdot$ and $\omega_1$ there is a polynomial $p_t$ such that for all $\vec{a}$, $|t(\vec{a})| \le p_t(|a_1|,\ldots,|a_n|)$. There is also a hierarchy of functions $\omega_n$, $n \ge 1$, which have subexponential growth rates, defined by $\omega_{n+1}(x,y) = 2^{\omega_n(\log x, \log y)}$. The axioms $\Omega_n$ are $\Pi_2$-axioms which say that the function $\omega_n$ is total. By using Parikh's Theorem 1.2.7.1, it is immediate that $I\Delta_0 + \Omega_n \nvdash \Omega_{n+1}$.

Although the $\omega_n$ functions, for $n \ge 2$, are superpolynomial, they are much more similar in nature to polynomial growth rate functions than to exponential growth rate functions. Using a technique due to Solovay [1976], it can be shown that, for each $n$, $I\Delta_0$ can define an inductive cut on which the $\omega_n$ function is provably total; for an explanation of this construction, see Pudlák [1983], Nelson [1986] or Chapter VIII of this volume. However, Paris and Dimitracopoulos [1982] showed that it is not possible to define an inductive cut on which the exponentiation function is provably total. For this reason, we view the $\omega_n$ functions as being more akin to feasible polynomial growth rate functions than to the infeasible exponential function (see Nelson [1986] for a strong expression of this viewpoint).

### 1.3.2. $\Delta_0$-formulas and the linear-time hierarchy

There is a very close connection between $\Delta_0$-expressibility and computational complexity. Recall that the linear time hierarchy consists of those predicates which can be recognized by some Turing machine which runs in linear time and which makes a bounded number of alternations between existential and universal states. Lipton [1978,sect. 4], building on work of Smullyan, Bennett and Wrathall, proved that the $\Delta_0$ definable predicates on $\mathbb{N}$ are precisely the subsets which are in the linear time hierarchy.

The original motivation for the definition of the theory $I\Delta_0$ by Parikh [1971] was to give a proof theory that would be appropriate to linear bounded automata, i.e., to predicates computable by linear space bounded Turing machines. It is still an open problem whether the linear time hierarchy equals linear space; although it is commonly conjectured that they are not equal. It is known that the linear time hierarchy contains log space, and also contains the predicates which can be computed by a Turing machine which simultaneously polynomial time and $n^{1-\epsilon}$ space for a constant $\epsilon > 0$ (see Bennett [1962] and Nepomnjaščii [1970]).

### 1.3.3. The theories $S_2^i$ and $T_2^i$ of bounded arithmetic

The second approach to theories of bounded arithmetic is due to Buss [1986] and gives a (conjectured) hierarchy of fragments of $I\Delta_0 + \Omega_1$, which are very closely related to the computational complexity classes of the polynomial time hierarchy. These fragments, $S_2^i$ and $T_2^i$ and others, use the language $0$, $S$, $+$, $\cdot$, $\#$, $|x|$, $\lfloor \frac{1}{2}x \rfloor$, and $\leq$; where the $\#$ function (pronounced 'smash') is defined so that $x\#y = 2^{|x|\cdot|y|}$. The $\#$ function was first introduced by Nelson [1986], and it is evident that the $\#$ function has essentially the same growth rate as the $\omega_1$-function.

The second difference between the $S_2^i$ and the $T_2^i$ theories and the $I\Delta_0 + \Omega_1$ approach is that the former theories are based on restricting the power of induction; firstly by further restricting the formulas for which induction holds, and secondly by using (apparently) weaker forms of induction. It is for this reason that the functions $|x|$ and $\lfloor \frac{1}{2}x \rfloor$ are included in the non-logical language, since they are needed to elegantly state the axioms of the theories $S_2^i$ and $T_2^i$.

Before defining the theories $S_2^i$ and $T_2^i$, we define the classes $\Sigma_i^b$ and $\Pi_i^b$ of formulas, which are defined by counting alternations of bounded quantifiers, ignoring sharply bounded quantifiers. (Bounded and sharply bounded quantifiers are defined in section 1 above.)

**Definition.** The set $\Delta_0^b = \Sigma_0^b = \Pi_0^b$ is equal to the set of formulas in which all quantifiers are sharply bounded. For $i \geq 1$, the sets $\Sigma_i^b$ and $\Pi_i^b$ are inductively defined by the following conditions:

(a) If $A$ and $B$ are $\Sigma_i^b$-formulas, then so are $A \vee B$ and $A \wedge B$. If $A$ is a $\Pi_i^b$ formula and $B$ is a $\Sigma_i^b$-formula, then $A \supset B$ and $\neg A$ are $\Sigma_i^b$-formulas.

(b) If $A$ is a $\Pi_{i-1}^b$-formula, then $A$ is a $\Sigma_i^b$-formula.

(c) If $A$ is a $\Sigma_i^b$-formula and $t$ is a term, then $(\forall x \le |t|)A$ is a $\Sigma_i^b$-formula.

(d) If $A$ is a $\Sigma_i^b$-formula and $t$ is a term, then $(\exists x \le t)A$ is a $\Sigma_i^b$-formula. Note this quantifier may be sharply bounded.

The four inductive conditions defining $\Pi_i^b$ are dual to (a)-(d) with the roles of existential and universal quantifiers and the roles of $\Pi_i^b$ and $\Sigma_i^b$ reversed.

This is a good place to justify the presence, in bounded arithmetic, of the $\#$ function or the equivalently growing $\omega_1$. There are essentially three reasons why it is natural to include $\#$ or $\omega_1$. Firstly, it gives a natural bound to the Gödel number of a formula $A(t)$ in terms of the Gödel numbers of $A$ and $t$; namely, the number of symbols in $A(t)$ is bounded by the product of the numbers of symbols in $A$ and in $t$. This allows a smooth arithmetization of metamathematics. Secondly, it arises naturally from consideration of bounded versus sharply bounded quantifiers, since it has exactly the growth rate necessary to make the following *quantifier exchange property* hold:

$$\begin{aligned}&(\forall x \le |a|)(\exists y \le b)A(x,y)\\ &\qquad \leftrightarrow \quad (\exists w \le SqBd(b,a))(\forall x \le |a|)(A(x,\beta(x+1,y)) \wedge \beta(x+1,y) \le b)\end{aligned}$$

where $SqBd$ is a term involving $\#$. In fact, the size of $w$ can be bounded in terms of $a$ and $b$, by noting that $w$ must encode $|a|+1$ many numbers of at most $|b|$ bits each; therefore, $w \le 2^{c\cdot|a|\cdot|b|}$ for some constant $c$, and $SqBd$ can easily be expressed using $\#$. The quantifier exchange property allows sharply bounded quantifiers to be pushed inside non-sharply bounded quantifiers (at least when the $\beta$ function is available). Thirdly, the use of $\#$ function means that any term $t(x)$ can be bounded by $2^{|x|^c}$ for some constant $c$, and conversely, any $2^{|x|^c}$ can be bounded by a term $t(x)$ in the language of bounded arithmetic. In other words, the terms define functions of *polynomial growth rate*. This leads to the principal importance of the classes $\Sigma_i^b$ and $\Pi_i^b$ of formulas, which is that they express precisely the corresponding classes of the polynomial time hierarchy. This fact is discussed in more depth in section 1.3.6 below, but in brief, a set of natural numbers is definable by a $\Sigma_i^b$-formula (respectively, a $\Pi_i^b$-formula) if and only if the set is recognizable by a predicate in the class $\Sigma_i^p$ (respectively, $\Pi_i^p$) from the polynomial time hierarchy. This is essentially due to Wrathall [1976] and Stockmeyer [1976] and was first proved in this exact form by Kent and Hodgson [1982]. Thus we have that $NP$, the set of nondeterministic polynomial time predicates, consists of precisely the predicates expressible by $\Sigma_1^b$-formulas, etc.

**1.3.3.1.** The theory $T_2^i$ will be defined by restricting induction to $\Sigma_i^b$-formulas, where by induction we mean the usual 'IND' flavor of induction. For $S_2^i$, we need some additional varieties of induction:

**Definition.** Let $\Phi$ be a set of formulas. The $\Phi$-PIND axioms are the formulas

$$A(0) \land (\forall x)(A(\lfloor \tfrac{1}{2}x \rfloor) \supset A(x)) \supset (\forall x)A(x)$$

for all formulas $A \in \Phi$. As usual, $A$ may have other free variables in addition to $x$ that serve as parameters. The length-induction $\Phi$-LIND axioms are the formulas

$$A(0) \land (\forall x)(A(x) \supset A(Sx)) \supset (\forall x)A(|x|)$$

for all $A \in \Phi$. The length-minimization axioms, $\Phi$-LMIN, are the formulas

$$(\exists x)A(x) \supset (\exists x)(A(x) \land (\forall y)(|y| < |x| \supset \neg A(y)))$$

for all $A \in \Phi$.

In addition to induction and minimization axioms, there are replacement axioms that will be defined below after the Gödel $\beta$ function has been introduced. All of these axiom schemes are used in conjunction with a set of purely universal axioms called the BASIC axioms. The set of BASIC axioms consists of:

| | |
|---|---|
| $a \le b \supset a \le Sb$ | $\|a\| = \|b\| \supset a\#c = b\#c$ |
| $a \ne Sa$ | $\|a\| = \|b\| + \|c\| \supset a\#d = (b\#d) \cdot (c\#d)$ |
| $0 \le a$ | $a \le a + b$ |
| $a \le b \land a \ne b \leftrightarrow Sa \le b$ | $a \le b \land a \ne b \supset$ |
| $a \ne 0 \supset 2 \cdot a \ne 0$ | $\quad S(2 \cdot a) \le 2 \cdot b \land S(2 \cdot a) \ne 2 \cdot b$ |
| $a \le b \lor b \le a$ | $a + b = b + a$ |
| $a \le b \land b \le a \supset a = b$ | $a + 0 = a$ |
| $a \le b \land b \le c \supset a \le c$ | $a + Sb = S(a + b)$ |
| $\|0\| = 0$ | $(a + b) + c = a + (b + c)$ |
| $\|S0\| = S0$ | $a + b \le a + c \leftrightarrow b \le c$ |
| $a \ne 0 \supset \|2 \cdot a\| = S(\|a\|) \land \|S(2 \cdot a)\| = S(\|a\|)$ | $a \cdot 0 = 0$ |
| $a \le b \supset \|a\| \le \|b\|$ | $a \cdot (Sb) = (a \cdot b) + a$ |
| $\|a\#b\| = S(\|a\| \cdot \|b\|)$ | $a \cdot b = b \cdot a$ |
| $0\#a = S0$ | $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$ |
| $a \ne 0 \supset 1\#(2 \cdot a) = 2 \cdot (1\#a)$ | $S0 \le a \supset (a \cdot b \le a \cdot c \leftrightarrow b \le c)$ |
| $\quad \land 1\#(S(2 \cdot a)) = 2 \cdot (1\#a)$ | $a \ne 0 \supset \|a\| = S(\|\lfloor \tfrac{1}{2}a \rfloor\|)$ |
| $a\#b = b\#a$ | $a = \lfloor \tfrac{1}{2}b \rfloor \leftrightarrow 2 \cdot a = b \lor S(2 \cdot a) = b$ |

These BASIC axioms serve the same role for $S_2^i$ and $T_2^i$ that the axioms of $Q$ served for the fragments $I\Sigma_n$ of Peano arithmetic. There is a certain amount of flexibility in the choice of BASIC axioms; essentially any finite set of purely universal axioms which both are sufficiently strong to carry out the bootstrapping of $S_2^1$ and are contained in the theory $S_2^1$ would serve as well for the BASIC axioms.[4]

[4]We have given the BASIC axioms as defined by Buss [1986]. This choice is not entirely optimal, since, for instance, the second axiom $a \le S(a)$, follows from the first, fourth and sixth axioms. An alternative, and weaker, set of BASIC axioms are given by Cook and Urquhart [1993]; see Buss [1992] for a discussion of their BASIC axioms. Buss and Ignjatović [1995] propose that $|a| \le a$ should be added to the BASIC axioms.

**Definition.** Let $i \geq 0$. $S_2^i$ is the theory axiomatized by the BASIC axioms plus $\Sigma_i^b$-PIND. $T_2^i$ is the theory axiomatized by BASIC plus $\Sigma_i^b$-IND. The theories $S_2^{(-1)}$ and $T_2^{(-1)}$ are equal to just BASIC.

$S_2$ is $\cup_{i \geq 0} S_2^i$ and $T_2$ is $\cup_{i \geq 0} T_2^i$. Section 1.3.5 shows that $S_2$ and $T_2$ are the same theory.

**1.3.3.2. Bootstrapping and $\Sigma_1^b$-definable functions.** The bootstrapping of $S_2^1$ and $T_2^1$ is analogous to the bootstrapping of $I\Delta_0$ as described in sections 1.2.6-1.2.8 above. There is now the additional difficulty that the induction axioms are more severely restricted; but on the other hand, the language of $S_2^i$ and $T_2^i$ is richer since it contains the function symbol $|x|$ and its BASIC axioms and this makes the definition of the graph of $y = 2^x$ essentially trivial, and thereby helps with defining Gödel numbering of sequences. The most outstanding difference between the bootstrapping of $S_2^1$ and $T_2^1$ and the above bootstrapping of $I\Delta_0$ is that quantifiers are more carefully counted; namely, whereas $I\Delta_0$ could use $\Delta_0$-defined predicates and $\Sigma_1$-defined functions, the theories $S_2^1$ and $T_2^1$ can introduce $\Delta_1^b$-defined predicates and $\Sigma_1^b$-defined functions. Accordingly, we make the following important definitions:

**Definition.** A predicate symbol $R(\vec{x})$ is *$\Delta_i^b$-defined by a theory $T$* if there is a $\Sigma_i^b$-formula $\phi(\vec{x})$ and a $\Pi_i^b$-formula $\psi(\vec{x})$ such that $R$ has *defining axiom*

$$R(\vec{x}) \leftrightarrow \phi(\vec{x})$$

and such that $T \vdash (\forall \vec{x})(\phi \leftrightarrow \psi)$.

**Definition.** Let $T$ be a theory of arithmetic. A function symbol $f(\vec{x})$ is *$\Sigma_i^b$-defined by $T$* if it has a defining axiom

$$y = f(\vec{x}) \quad \leftrightarrow \quad \phi(\vec{x}, y),$$

where $\phi$ is a $\Sigma_i^b$ formula with all free variables indicated such that $T$ proves $(\forall \vec{x})(\exists! y)\phi(\vec{x}, y)$.

By Parikh's theorem 1.2.7.1, when $f$ is $\Sigma_i^b$-defined then $T \vdash (\forall \vec{x})(\exists y \leq t(\vec{x}))\phi(\vec{x}, y)$ for some term $t$.

The analogue of Theorem 1.2.7.3 for fragments of bounded arithmetic is the following theorem.

**1.3.3.3. Theorem.** (Buss [1986,Thm 2.2]) *Let $T \supseteq BASIC$ be a theory of arithmetic. Let $T$ be extended to a theory $T^+$ in an enlarged language $L^+$ by adding $\Delta_1^b$-defined predicate symbols, $\Sigma_1^b$-defined function symbols and their defining equations. Then $T^+$ is conservative over $T$. Also, if $A$ is a $\Sigma_i^b$ (respectively, a $\Pi_i^b$) formula in the enlarged language $L^+$, then there is a formula $A^-$ in the language of $T$ such that $A^-$ is also in $\Sigma_i^b$ (respectively, $\Pi_i^b$) and such that*

$$T^+ \vdash (A \leftrightarrow A^-).$$

An immediate corollary to this theorem is that, for $i \geq 1$, theories such as $S_2^i$ and $T_2^i$ can introduce $\Sigma_1^b$-defined function symbols and $\Delta_1^b$-predicate symbols and use them freely in induction axioms.

With the aid of Theorem 1.3.3.3, the bootstrapping for $S_2^1$ and $T_2^1$ is analogous to the bootstrapping for $I\Delta_0$ in section 1.2.8; indeed, every single function and predicate symbol which was claimed to be $\Sigma_1$-definable or $\Delta_0$-definable (respectively) in $I\Delta_0$ in section 1.2.8 is likewise $\Sigma_1^b$-definable or $\Delta_1^b$-definable in each of the six theories $S_2^1$, $T_2^1$, $BASIC + \Pi_1^b\text{-}PIND$, $BASIC + \Sigma_1^b\text{-}LIND$, $BASIC + \Pi_1^b\text{-}LIND$ and $BASIC + \Pi_1^b\text{-}IND$. We shall omit the details of this bootstrapping here; they can be found in Buss [1986,1992] and Buss and Ignjatović [1995].

One consequence of the bootstrapping process is that some of the other forms of induction follow from $\Sigma$-*PIND* and $\Pi$-*IND*:

**1.3.3.4. Theorem.** (Buss [1986]) *Let $i \geq 1$.*

(1) *$T_2^i$ proves $\Pi_i^b$-IND and $T_2^i \vDash S_2^i$.*

(2) *$S_2^i$ proves $\Sigma_i^b$-LIND, $\Pi_i^b$-PIND and $\Pi_i^b$-LIND.*

### 1.3.4. Polynomial time computable functions in $S_2^1$

The last section discussed the fact that $\Sigma_1^b$-definable functions and $\Delta_1^b$-defined predicates can be introduced into theories of bounded arithmetic and used freely in induction axioms. Of particular importance is the fact that these include all polynomial time computable functions and predicates.

A function or predicate is said to be *polynomial time* computable provided there exists a Turing machine $M$ and a polynomial $p(n)$, such that $M$ computes the function or recognizes the predicate, and such that $M$ runs in time $\leq p(n)$ for all inputs of length $n$. The inputs and outputs for $M$ are integers coded in binary notation, thus the length of an input is proportional to the total length of its binary representation.

For our purposes, it is convenient to use an alternative definition of the polynomial time computable functions; the equivalence of this definition is due to Cobham [1965].

**Definition.** The *polynomial time* functions on $\mathbb{N}$ are inductively defined by

(1) The following functions are polynomial time:
   - The nullary constant function $0$.
   - The successor function $x \mapsto S(x)$.
   - The doubling function $x \mapsto 2x$.
   - The conditional function $Cond(x, y, z) = \begin{cases} y & \text{if } x = 0 \\ z & \text{otherwise.} \end{cases}$

(2) The projection functions are polynomial time functions and the composition of polynomial time functions is a polynomial time function.

(3) If $g$ is a $(n-1)$-ary polynomial time function and $h$ is a $(n+1)$-ary polynomial time function and $p$ is a polynomial, then the following function $f$, defined by *limited iteration on notation from $g$ and $h$*, is also polynomial time:

$$\begin{aligned} f(0,\vec{x}) &= g(\vec{x}) \\ f(z,\vec{x}) &= h(z,\vec{x},f(\lfloor \tfrac{1}{2}z \rfloor,\vec{x})) \quad \text{for } z \neq 0 \end{aligned}$$

provided $|f(z,\vec{x})| \leq p(|z|,|\vec{x}|)$ for all $z,\vec{x}$.

A predicate is *polynomial time* computable provided its characteristic function is polynomial time. The class of polynomial time functions is denoted $\square_1^p$, and the class of polynomial time predicates is denoted $\Delta_1^p$.

**1.3.4.1. Theorem.** (Buss [1986])
(a) *Every polynomial time function is $\Sigma_1^b$-definable in $S_2^1$.*
(b) *Every polynomial time predicate is $\Delta_1^b$-definable in $S_2^1$.*

Once one has bootstrapped $S_2^1$ sufficiently to intensionally introduce sequence coding functions, it is fairly straightforward to prove this theorem using Cobhams's inductive definition of polynomial time computability. The main case in the proof by induction is the case where $f$ is defined from $g$ and $h$ by limited iteration on notation: in this case the predicate $f(z,\vec{x}) = y$ is defined similarly to the way $f(m,\vec{x}) = y$ was defined in the proof of Theorem 1.2.10; the main difference now is that $\Sigma_1^b$-PIND is used to prove $w$ exists, and for this it is necessary to bound $w$ with a term. Fortunately, the bounding condition $|f(z,\vec{x})| \leq p(|z|,|\vec{x}|)$ makes it possible to bound the elements of $w$, and hence $w$, with a term. We leave the details to the reader. □

A second way to approach defining the polynomial time function in $S_2^1$ is to directly formalize polynomial time computability using Turing machine computations, instead of using Cobham's definition. This can also be formalized in $S_2^1$; furthermore $S_2^1$ can prove the equivalence of the two approaches. See Buss [1986] for more details.

For $i \geq 1$, $S_2^i \supseteq S_2^1$ and also, by Theorem 1.3.5 below, $T_2^i \supseteq S_2^1$. Therefore, the above theorem, combined with Theorem 1.3.3.3 gives:

**1.3.4.2. Theorem.** (Buss [1986]) *Let $i \geq 1$. The theories $S_2^i$ and $T_2^i$ can introduce symbols for polynomial time computable functions and predicates and use them freely in induction axioms.*

We shall show later (Theorem 3.2) that the converse to Theorem 1.3.4.1 also holds and that $S_2^1$ can $\Sigma_1^b$-define and $\Delta_1^b$-define precisely the polynomial time computable functions and predicates, respectively.

### 1.3.5. Relating $S_2^i$ and $T_2^i$

It is clear that $S_2^i \supseteq S_2^1$ and $T_2^i \supset T_2^1$, for $i \geq 1$. In addition we have the following relationships among these theories:

**Theorem.** (Buss [1986]) *Let* $i \geq 1$.
(1) $T_2^i \supseteq S_2^i$.
(2) $S_2^i \supseteq T_2^{i-1}$.

It is however open whether the theories

$$S_2^1 \subseteq T_2^1 \subseteq S_2^2 \subseteq T_2^2 \subseteq S_2^3 \subseteq \cdots$$

are distinct.

**Proof.** A proof of (1) can be found in Buss [1986,sect 2.6]: this proof mostly involves bootstrapping of $T_2^i$, and we shall not present it here.

The proof of (2) uses a divide-and-conquer method. Fix $i \geq 1$ and fix a $\Sigma_{i-1}^b$-formula $A(x)$; we must prove that $S_2^i$ proves the IND axiom for $A$. We argue informally inside $S_2^i$, assuming $(\forall x)(A(x) \supset A(x+1))$. Let $B(x,z)$ be the formula

$$(\forall w \leq x)(\forall y \leq z+1)(A(w \dot{-} y) \supset A(w)).$$

Clearly $B$ is equivalent to a $\Pi_i^b$-formula. By the definition of $B$, it follows that

$$(\forall x)(\forall z)(B(x, \lfloor \tfrac{1}{2}z \rfloor) \supset B(x,z)),$$

and hence by $\Pi_i^b$-PIND on $B(x,z)$ with respect to $z$,

$$(\forall x)(B(x,0) \supset B(x,x)).$$

Now, $(\forall x)B(x,0)$ holds as it is equivalent to the assumption $(\forall x)(A(x) \supset A(x+1))$, and therefore $(\forall x)B(x,x)$ holds. Finally, $(\forall x)B(x,x)$ immediately implies $(\forall x)(A(0) \supset A(x))$: this completes the proof of the IND axiom for $A$.

The theorem immediately implies the following corollary:

**Corollary.** (Buss [1986]) $S_2 = T_2$.

In the proof of the above theorem, it would suffice for $A(x)$ to be $\Delta_i^b$ with respect to $S_2^i$. Therefore, $S_2^i$ proves $\Delta_i^b$-IND.

### 1.3.6. Polynomial hierarchy functions in bounded arithmetic

The polynomial time hierarchy is a hierarchy of bounded alternation polynomial time computability; the base classes are the class $P = \Delta_1^p$ of polynomial time recognizable predicates, the class $FP = \square_1^p$ of polynomial time computable functions, the class $NP = \Sigma_1^p$ of predicates computable in nondeterministic polynomial time, the class $coNP = \Pi_1^p$ of complements of $NP$ predicates, etc. More generally, $\Delta_i^p$, $\square_i^p$, $\Sigma_i^p$ and $\Pi_i^p$ are defined as follows:

**Definition.** The classes $\Delta^p_1$ and $\square^p_1$ have already been defined. Further define, by induction on $i$,

(1) $\Sigma^p_i$ is the class of predicates $R(\vec{x})$ definable by

$$R(\vec{x}) \Leftrightarrow (\exists y \leq s(\vec{x}))(Q(\vec{x}, y))$$

for some term $s$ in the language of bounded arithmetic, and some $\Delta^p_i$ predicate $Q$.

(2) $\Pi^p_i$ is the class of complements of predicates in $\Sigma^p_i$.

(3) $\square^p_{i+1}$ is class of predicates computable on a Turing in polynomial time using an oracle from $\Sigma^p_i$.[5]

(4) $\Delta^p_{i+1}$ is the class of predicates which have characteristic function in $\square^p_{i+1}$.

The connection between the syntactically defined classes of formulas $\Sigma^b_i$ defined by counting alternations of quantifiers and the computationally defined classes $\Sigma^p_i$ is given by the next theorem.

**Theorem.** (Wrathall [1976], Stockmeyer [1976], Kent and Hodgson [1982])
*A predicate is $\Sigma^p_i$ if and only if there is a $\Sigma^b_i$-formula which defines it.*

**Proof.** The easier part of the proof is that every $\Sigma^b_i$-formula defines a $\Sigma^p_i$-predicate. For this, start by noting that a sharply bounded formula defines a polynomial time predicate, even when the $\beta$ function and pairing functions are present. Then, given a $\Sigma^b_i$-formula, one can use the quantifier exchange property to push sharply bounded quantifiers inward and can use pairing functions to combine adjacent like quantifiers; this transforms the formula into an equivalent formula which explicitly defines a $\Sigma^p_i$ property according to the above definition.

For the reverse inclusion, use induction on $i$. To start the induction, note that Theorem 1.3.4.1 already implies that every $\Delta^p_1$ predicate is defined by both a $\Sigma^b_1$- and a $\Pi^b_1$-formula. For the first part of the induction step, assume that every $\Delta^p_i$ predicate is definable by both a $\Sigma^b_1$ and a $\Pi^b_i$-formula. Then it is immediate that every $\Sigma^p_i$ predicate is definable by a $\Sigma^b_i$-formula. For the second part of the induction step, we must prove that every $\Delta^p_{i+1}$-predicate is definable by both a $\Sigma^p_{i+1}$- and a $\Pi^p_{i+1}$-formula. For this, note that it suffices to prove that every $\square^p_{i+1}$-function has its graph defined by a $\Sigma^b_{i+1}$-formula. To prove this last fact, use induction on the definition of the functions in $\square^p_{i+1}$: it is necessary to show that this condition is preserved by definition using composition as well as by definition by limited iteration on notation. The proofs of these facts are fairly straightforward and can even be formalized by $S^i_2$, which gives the following theorem:

[5] An oracle from $\Sigma^p_i$ is just a predicate from $\Sigma^p_i$. For our purposes, the most convenient way to define the class of functions *polynomial time relative to an oracle* $R$ is as the smallest class of functions containing all polynomial time functions and the characteristic function of $R$ and closed under composition and limited iteration on notation.

**Theorem.** (Buss [1986])) *Let $i \geq 1$.*
(a) *Every $\square_i^p$ function is $\Sigma_i^b$-definable in $S_2^i$.*
(b) *Every $\Delta_i^p$ predicate is $\Delta_i^b$-definable in $S_2^i$.*

**Proof.** The proof proceeds by induction on $i$. The base case has already been done as Theorem 1.3.4.1. Part (b) is implied by (a), so it suffices to prove (a). To prove the inductive step, we must show the following three things (and show they are provable in $S_2^i$):
(1) If $f(\vec{x}, y)$ is a $\square_{i-1}^p$-function, then the characteristic function $\chi(\vec{x})$ of $(\exists y \leq t(\vec{x}))(f(\vec{x}, y) = 0)$ is $\Sigma_i^b$-definable. To prove this, we have by the induction hypothesis that $f(\vec{x}, y) = z$ is equivalent to a $\Sigma_{i-1}^b$ formula $A(\vec{x}, y, z)$. The $\Sigma_i^b$-definition of $\chi(\vec{x})$ is thus[6]

$$\chi(\vec{x}) = z \;\Leftrightarrow\; (z = 0 \wedge (\exists y \leq t)A(\vec{x}, y, 0)) \vee (z = 1 \wedge \neg(\exists y \leq t)A(\vec{x}, y, 0))$$

which is clearly equivalent to a $\Sigma_i^b$-formula by prenex operations.
(2) If functions $g$ and $\vec{h}$ have graph definable by $\Sigma_i^b$-formulas, then so does their composition. As an example of how to prove this, suppose $f(\vec{x}) = g(\vec{x}, h(\vec{x}))$; then the graph of $f$ can be defined by

$$f(\vec{x}) = y \;\Leftrightarrow\; (\exists z \leq t_h(\vec{x}))(h(\vec{x}) = z \wedge g(\vec{x}, z) = y),$$

where $t_h$ is a term bounding the function $h$.
(3) If $f$ is defined by limited iteration from $g$ and $h$ with bounding polynomial $p$, and $g$ and $h$ have $\Sigma_i^b$-definable graphs, then so does $f$. To prove this, show that $f(z, \vec{x}) = y$ is expressed by the formula

$$\begin{aligned}(\exists w \leq SqBd(2^{p(|z|,|\vec{x}|)}, z))[\beta(|z|+1, \vec{x}) = y \wedge \beta(1, w) = g(\vec{x}) \wedge \\ \wedge (\forall i < |z|)(\beta(i+2, w) = \min\{h(\lfloor \tfrac{z}{2^{|z| \dot{-} i \dot{-} 1}} \rfloor, \vec{x}, \beta(i+1, w)), 2^{p(|i+1|,|\vec{x}|)}\})].\end{aligned}$$

Here the term $SqBd(\cdots)$ has been chosen sufficiently large to bound the size of the sequence $w$ encoding the steps in the computation of $f(z, \vec{x})$. The formula is clearly in $\Sigma_i^b$, and the theory $S_2^i$ can prove the existence and uniqueness of $w$ by PIND induction up to $z$. $\Box$

A more complicated proof can establish the stronger result that $T_2^{i-1}$ can also $\Sigma_i^b$-define the $\square_i^p$-functions:

**Theorem.** (Buss [1990]) *Let $i > 1$.*
(a) *Every $\square_i^p$ function is $\Sigma_i^b$-definable in $T_2^{i-1}$.*
(b) *Every $\Delta_i^p$ predicate is $\Delta_i^b$-definable in $T_2^{i-1}$.*

[6] We use the convention that a characteristic function of a predicate equals zero when the predicate is true.

It is a very interesting question whether the possible collapse of the polynomial-time hierarchy is related to the possible collapse of the hierarchy of bounded arithmetic theories. So far what is known is that if $S_2$ is finitely axiomatized (more precisely, if $T_2^i = S_2^{i+1}$ for some $i \geq 1$), then the polynomial time hierarchy collapses provably in $T_2$ (see Krajíček, Pudlák and Takeuti [1991], Buss [1995], Zambella [1996], and section 3.3.2). This means that the hierarchy of theories of bounded arithmetic collapses if and only if the polynomial time hierarchy collapses $S_2$-provably.

### 1.3.7. The theories $PV_i$

Since $T_2^{i-1}$ and $S_2^i$ can $\Sigma_i^b$-define the $\square_i^p$ functions, it is often convenient to conservatively extend the language of bounded arithmetic with symbols for these functions. Accordingly, we define $T_2^{i-1}(\square_i^p)$ and $S_2^i(\square_i^p)$ to be the (conservative) extensions of $T_2^{i-1}$ and $S_2^i$ to the language containing symbols for the $\square_i^p$-functions with their $\Sigma_i^b$-defining equations as new axioms. For $i = 1$, the theory $T_2^0(\square_1^p)$ has to be defined slightly differently, since $T_2^0$ does not have sufficient bootstrapping power to $\Sigma_1^b$-define the polynomial time functions. Instead, $T_2^0(\square_1^p)$ is defined to have first-order language consisting of symbols for all polynomial time functions and predicates, and to have as axioms (1) the $BASIC$ axioms, (2) axioms that define the non-logical symbols in the spirit of Cobham's definition of the polynomial time and (3) IND for all sharply bounded (equivalently, all atomic) formulas.[7]

One must be careful when working with $T_2^{i-1}(\square_i^p)$ and $S_2^i(\square_i^p)$ since, for $i > 1$, the functions symbols for $\square_i^p$ cannot be used freely in induction axioms (modulo some open questions).

Since the notation $T_2^{i-1}(\square_i^p)$ is so atrocious, it is sometimes denoted $PV_i$ instead. Krajíček, Pudlák and Takeuti [1991] prove that $PV_i$ can be axiomatized by purely universal axioms: to see the main idea of the universal axiomatization, note that if $A$ is $\Delta_i^b$, then $PV_i$ proves $A$ is equivalent to a quantifier-free formula via Skolemization and thus induction on $A(x, \vec{c})$, can be obtained from the universal formula

$$(\forall \vec{c})(\forall t)[A(0, \vec{c}) \land \neg A(t, \vec{c})) \supset A(f_A(t, \vec{c}) \dot{-} 1, \vec{c}) \land \neg A(f_A(t, \vec{c}), \vec{c})]$$

where $f_A$ is computed by a binary search procedure which asks $\Delta_i^p$ queries to find a value $b$ for which $A(b-1, \vec{c})$ is true and $A(b, \vec{c})$ is false. Of course, this $f$ is a $\square_i^p$-function and therefore is a symbol in the language of $PV_i$.

### 1.3.8. More axiomatizations of bounded arithmetic

For any theory $T$ in which the Gödel $\beta$ function is present or is $\Sigma_1^b$-definable, in particular, for any theory $T \supseteq S_2^1$, there are two further possible axiomatizations that are useful for bounded arithmetic:

---

[7] The original definition of a theory of this type was the definition of equational theory $PV$ of polynomial time functions by Cook [1975]. $T_2^0(\square_1^p)$ can also be defined as the conservative extension of $PV$ to first-order logic.

**Definition.** Let $\Phi$ be a set of formulas. The $\Phi$*-replacement* axioms are the formulas

$$(\forall x \le |s|)(\exists y \le t)A(x,y) \supset (\exists w)(\forall x \le |s|)(A(x,\beta(x+1,w)) \wedge \beta(x+1,w) \le t)$$

for all formulas $A \in \Phi$ and all appropriate (semi)terms $s$ and $t$. As usual $A$ may have other free variables in addition to $x$ that serve as parameters.

The *strong* $\Phi$*-replacement* axioms are similarly defined to be the formulas

$$(\exists w)(\forall x \le |s|)[(\exists y \le t)A(x,y) \leftrightarrow A(x,\beta(x+1,w)) \wedge \beta(x+1,w) \le t].$$

The replacement and strong replacement axioms contain an apparently unbounded quantifier $(\exists w)$; however, $S^1_2$ can always bound $w$ by a term $SqBd(t,s)$ which is large enough to bound a sequence of $|s|+1$ values $\le t$. For example, setting $SqBd(t,s)$ equal to $(2t+1)\#(2(2s+1)^2)$ will work for the sequence encoding given in section 1.2.8.

It is known that the $\Sigma^b_i$-replacement axioms are consequences of the $\Sigma^b_i$-PIND axioms, and that the strong $\Sigma^b_i$-replacement axioms are equivalent to the $\Sigma^b_i$-PIND axioms (for $i \ge 1$, and over the base theory $S^1_2$). Figure 1 shows these and other relationships among the axiomatizations of bounded arithmetic.

### 1.4. Sequent calculus formulations of arithmetic

This section discusses the proof theory of theories of arithmetic in the setting of the sequent calculus: this will be an essential tool for our analysis of the proof-theoretic strengths of fragments of arithmetic and of their interrelationships. The sequent calculus used for arithmetic is based on the system $LK_e$ described in Chapter I of this volume; $LK_e$ will be enlarged with additional rules of inference for induction, minimization, etc., and for theories of bounded arithmetic, $LK_e$ is enlarged to include inference rules for bounded quantifiers.

**1.4.1. Definition.** $LKB$ (or $LKB_e$) is the sequent calculus $LK$ (respectively, $LK_e$) extended as follows: First, the language of first-order arithmetic is expanded to allow bounded quantifiers as a basic part of the syntax. Second, the following new rules of inference are allowed:

**Bounded quantifier rules**

$$\forall\le\!\textit{:left}\;\frac{A(t),\Gamma\longrightarrow\Delta}{t\le s,(\forall x\le s)A(x),\Gamma\longrightarrow\Delta} \qquad \forall\le\!\textit{:right}\;\frac{b\le s,\Gamma\longrightarrow\Delta,A(b)}{\Gamma\longrightarrow\Delta,(\forall x\le s)A(x)}$$

$$\exists\le\!\textit{:left}\;\frac{b\le s,A(b),\Gamma\longrightarrow\Delta}{(\exists x\le s)A(x),\Gamma\longrightarrow\Delta} \qquad \exists\le\!\textit{:right}\;\frac{\Gamma\longrightarrow\Delta,A(t)}{t\le s,\Gamma\longrightarrow\Delta,(\exists x\le s)A(x)}$$

where the variable $b$ is an eigenvariable and may not occur in in $s$ or in $\Gamma,\Delta$.

The Cut Elimination and Free-cut Elimination Theorems still hold for $LKB$ and $LKB_e$, in the exact same form as they were proved to hold for $LK$ and $LK_e$

$$
\begin{array}{l}
\Sigma_i^b\text{-IND} \iff \Pi_i^b\text{-IND} \iff \Sigma_i^b\text{-MIN} \iff \Delta_{i+1}^b\text{-IND} \\
\quad\Big\Downarrow \\
\Sigma_i^b\text{-PIND} \iff \Pi_i^b\text{-PIND} \iff \Sigma_i^b\text{-LIND} \iff \Pi_i^b\text{-LIND} \\
\quad\Big\Updownarrow \\
\Sigma_i^b\text{-LMIN} \iff (\Sigma_{i+1}^b \cap \Pi_{i+1}^b)\text{-PIND} \\
\quad\Big\Downarrow \\
\Sigma_{i-1}^b\text{-IND}
\end{array}
$$

$$\Sigma_{i+1}^b\text{-MIN} \iff \Pi_i^b\text{-MIN}$$

$$S_2^i \underset{\Sigma_i^b}{\succ} T_2^{i-1}$$

$$S_2^i \underset{\mathcal{B}(\Sigma_i^b)}{\succ} T_2^{i-1} + \Sigma_i^b\text{-replacement}$$

$$\Sigma_1^b\text{-}\mathit{PIND} + \Sigma_{i+1}^b\text{-replacement} \implies \Sigma_i^b\text{-PIND} \implies \Sigma_i^b\text{-replacement}$$

$$\Sigma_i^b\text{-PIND} \iff \Sigma_1^b\text{-}\mathit{PIND} + \text{strong } \Sigma_i^b\text{-replacement}$$

Figure 1

Relationships among axiomatizations for Bounded Arithmetic relative to the base theory $BASIC$ with $i \geq 1$; $T_2^0$ should be interpreted as $PV_1$. See Buss [1986,1990], Buss and Ignjatović [1995] for proofs.

in Chapter I. The principal formulas of the bounded quantifier inferences are the formulas $t \leq s$ and $(Qx \leq s)A$ introduced in the lower sequent; as usual, a cut on a direct descendent of a principal formula is anchored.

**1.4.2. Rule forms of induction.** We next introduce inference rules which are equivalent to induction axioms; the reason for using rules of inference for induction in place of induction axioms is that the use of free-cut free proofs provides a powerful proof-theoretic tool for the analysis of fragments of arithmetic.

**Definition.** Let $\Psi$ be a class of formulas. Then $\Psi$-IND induction rules are the inferences of the form

$$\frac{A(b), \Gamma \longrightarrow \Delta, A(b+1)}{A(0), \Gamma \longrightarrow \Delta, A(t)}$$

where $A \in \Psi$ and where the *eigenvariable* $b$ does not occur except as indicated.

The $\Psi$-PIND induction rules are the inferences of the form

$$\frac{A(\lfloor \frac{1}{2}b \rfloor), \Gamma \longrightarrow \Delta, A(b)}{A(0), \Gamma \longrightarrow \Delta, A(t)}$$

where again $A \in \Psi$ and $b$ occurs only as indicated.

We leave it to the reader to check that the induction rules $\Psi$-IND and $\Psi$-PIND are equivalent to the induction axioms $\Psi$-IND and $\Psi$-PIND (respectively); in fact, this is true for *any* class $\Psi$ of formulas. The fact that the induction rules are equivalent to the induction axioms depends crucially on the presence of the the side formulas $\Gamma$ and $\Delta$ in the inference; when side formulas are not allowed, the inference rules are often slightly weaker than the induction axioms; see, e.g., Parsons [1972] and Sieg [1985]. It follows that theories such as $I\Delta_0$, $I\Sigma_n$, $I\Pi_n$, $S_2^i$ and $T_2^i$ can be equivalently formulated using induction rules instead of induction axioms. For the rest of this chapter, we will presume that these theories are formulated with the induction rules.

As was discussed in Chapter I, the free-cut elimination theorem holds for theories such as $I\Sigma_n$, $I\Pi_n$, $S_2^i$ and $T_2^i$. In particular, we have the following corollary to the free-cut elimination theorem, which generalizes the subformula property to fragments of arithmetic. For this theorem, $\Psi$ must be a class of formulas which is closed under the operations of taking subformulas and freely substituting terms for variables. Strictly speaking, classes such as $\Sigma_i$ are not closed under subformulas, since a $\Sigma_i$-formula may contain a (negated) $\Pi_i$-subformula; however, one may instead use the class of $\Sigma_i$-formulas in which all negation signs are in front of atomic subformulas. This can be done without loss of generality and then this class of formulas is closed both under subformulas and under term substitution.

**Theorem.** *Let $\Psi$ be a class of formulas closed under subformulas and under term substitution and containing the atomic formulas. Let $R$ be a fragment of arithmetic axiomatized by $\Psi$-IND (or by $\Psi$-PIND) plus initial sequents containing only formulas from $\Psi$. Also suppose that the sequent $\Gamma \longrightarrow \Delta$ contains only formulas from $\Psi$ and that $R \vdash \Gamma \longrightarrow \Delta$. Then there is an $R$-proof of $\Gamma \longrightarrow \Delta$ such that every formula appearing in the proof is a $\Psi$-formula.*

The proof of this theorem is of course based on the fact that every formula appearing in an $R$-proof either is a direct descendent of a formula in an initial sequent or is an ancestor (and hence subformula in the wide sense) of either a cut formula or a formula in the endsequent. Furthermore, in a free-cut free $R$-proof, all cut-formulas are in $\Psi$ and by free-cut elimination, $\Gamma \longrightarrow \Delta$ has a free-cut free proof.

The above theorem turns out be an extremely powerful tool for the proof-theoretic analysis of fragments of arithmetic.

**1.4.3.** We now state and prove a generalization of Theorem 1.2.7.1 which applies to very general bounded theories $R$, possibly including induction inferences for bounded formulas. Assume that $R$ contains $\leq$ in its language and that $R$ proves that $\leq$ is reflexive and transitive. Also suppose that for all terms $r$ and $s$, there is a term $t$ so that $R$ proves $r \leq t$ and $s \leq t$. Further suppose that for all terms $t(\vec{a}, b)$ and $r(\vec{a})$, there is a term $s$ so that $R$ proves that $b \leq r(\vec{a}) \supset t(\vec{a}, b) \leq s(\vec{a})$.

**Parikh's Theorem.** *Let $R$ be a bounded theory satisfying the above conditions and $A(\vec{x}, y)$ a bounded formula. Suppose $R \vdash (\forall \vec{x})(\exists y)A(\vec{x}, y)$. Then there is a term $t$ such that $R$ also proves $(\forall \vec{x})(\exists y \leq t)A(\vec{x}, y)$.*

**1.4.4. Proof.** (Sketch). By the free-cut elimination theorem, there is a free-cut free $R$-proof $P$ of $(\exists y)A(\vec{b}, y)$, where the $b$'s are new free variables. By the subformula property, every sequent $\Gamma \longrightarrow \Delta$ in the proof $P$ contains only bounded formulas in its antecedent $\Gamma$ and its antecedent $\Delta$ contains only bounded formulas plus possibly occurrences of the formula $(\exists y)A(\vec{b}, y)$. Given such a $\Delta$ and given a term $t$, let $\Delta^{\leq t}$ denote the result of removing all occurrences of $(\exists y)A(\vec{b}, y)$ from $\Delta$ and adding the formula $(\exists y \leq t)A(\vec{b}, y)$. It is straightforward to prove by induction on the number of inferences in $P$ that, for each sequent $\Gamma \longrightarrow \Delta$ in $P$, there is a term $t$ such that $R$ proves $\Gamma \longrightarrow \Delta^{\leq t}$. □

**1.4.5. Inference rules for collection.** Just as it possible to replace induction axioms with induction inferences, it is also possible to formulate the collection axioms of $B\Sigma_i$ as rules of inference. The $\Sigma_i$-collection inferences, $\Sigma_i$-REPL, are

$$\frac{\Gamma \longrightarrow \Delta, (\forall x \leq t)(\exists y)A(x, y)}{\Gamma \longrightarrow \Delta, (\exists z)(\forall x \leq t)(\exists y \leq z)A(x, y)}$$

It is not difficult to check that the inference rule for collection (replacement) is equivalent to the axiom form of collection. Furthermore, the free-cut elimination theorem holds as before; however, the notion of 'free-cut' is changed by also declaring every direct descendent of the principal formula of a collection inference to be anchored.

One easy consequence of free-cut elimination for collection inferences is that Parikh's Theorem 1.4.3 holds also for theories $R$ that contain $\Sigma_1$-REPL; compare this to Theorem 3.4.1 about the conservativity of $B\Sigma_{i+1}$ over $I\Sigma_i$.

## 2. Gödel incompleteness

Gödel's incompleteness theorems, on the impossibility of giving an adequate and complete axiomatization for mathematics, were of great philosophical and foundational importance to mathematics. They are arguably the most important results in mathematical logic since the development of first-order logic. Loosely speaking, the incompleteness theorems state that any sufficiently expressive, consistent theory with a decidable axiomatization is not complete; and furthermore, for any such theory, the second incompleteness theorem gives an explicit, non self-referential true statement which is not a consequence of the theory. More generally, the set of $\Pi_1$-sentences true about the integers[8] is not recursively enumerable, so there is no way to generalize or replace first-order logic with any other kind of formal system which both admits a decidable notion of 'provability' and is complete in the sense of 'proving' every true $\Pi_1$-sentence of the integers.

### 2.1. Arithmetization of metamathematics

The usual methods of proving Gödel's incompleteness theorems involve coding metamathematical concepts (i.e., coding the syntax of first-order logic) with integers and then using a self-referential or diagonal construction to obtain non-provable true statements;[9] this process of coding syntactic aspects of logic with integers is called 'arithmetization'. There are essentially two different approaches to the arithmetization of syntax. The first approach uses numeralwise representability as a means of representing computable functions: a numeralwise representation of a function gives a characterization of the function's values for particular choice of inputs to the function. To be precise, a formula $A(\vec{x}, y)$ *numeralwise represents* a function $f(\vec{x}) = y$ in a theory $T$ if and only if, for every particular integers $n_1, \ldots, n_k$ with $f(\vec{n}) = m$, the theory $T$ proves

$$(\forall y)(A(S^{n_1}0, \ldots, S^{n_k}, y) \leftrightarrow y = S^m 0).$$

It turns out that every recursive function is numeralwise representable even in very weak theories such as $R$ and $Q$; and conversely, only recursive functions are numeralwise representable in *any* axiomatizable theory, no matter how strong.

However, numeralwise representation of $f$ in the theory $T$ only implies that $T$ can 'represent' all particular, fixed values of $f$; this in no way implies that $T$ can prove general properties of the function $f$. This in is in contrast to the second approach to the arithmetization of syntax which involves giving intensional definitions of certain (but not all) recursive functions. In the intensional approach to arithmetization of metamathematics, one gives formulas which define concepts such as "formula", "term", "substitution", "proof", "theorem", etc; these definitions are said to be *intensional* provided the theory $T$ can prove simple properties of these concepts. For instance, one wants the theory $T$ to be able to define the notion of substituting a term into a formula and prove the result is a formula; similarly, $T$ should be able to prove that the set of theorems is closed under modus ponens; etc. (See Feferman [1960] for a comprehensive discussion of intensionality.)

It is significantly more work to carry out the details an intensional arithmetization of syntax than a numeralwise representation of recursive functions; indeed, an

[8] By Matijacevič's theorem, the same holds for the true, purely universal sentences (in the language of $PA$).

[9] There are approaches to the first incompleteness theorem that avoid this arithmetization of metamathematics; for instance, one can directly prove that the true $\Pi_1$ sentences of arithmetic do not form a recursively enumerable set, say by encoding Turing machine computations with integers. For somewhat different approaches based on Berry's paradox, see Chaitin [1974] and Boolos [1989].

intensional definition of a function is typically a numeralwise representation of the same function. Furthermore, the intensional representation requires the additional verification that the underlying theory can prove simple facts about the function. Nonetheless, the intensional definition has significant advantages, most notably, in allowing a smoother treatment of the Gödel incompleteness theorem, especially the second incompleteness theorem.

Since many textbooks discuss the first approach based on numeralwise representability and since we prefer the intensional approach, this article will deal only with the intensional approach. The reader who wants to see the numeralwise representability approach can consult Smorynski [1977] and any number of textbooks such as Mendelson [1987]. The intensional approach is due to Feferman [1960]. An effective unification of the two approaches can be given using the fact (independently due to Wilkie and to Nelson [1986]) that $I\Delta_0+\Omega_1$ and $S^1_2$ are interpretable in $Q$; since both $I\Delta_0+\Omega_1$ and $S^1_2$ admit a relatively straightforward intensional arithmetization of metamathematics (see Wilkie and Paris [1987] and Buss [1986]), this allows strong forms of incompleteness obtained via the intensional approach to apply also to the theory $Q$; paragraph 2.1.4 below sketches how the interpretation of $S^1_2$ in $Q$ can be used to give an intensional arithmetization in $Q$. The book of Smullyan [1992] gives a modern, in-depth treatment of Gödel's incompleteness theorems.

**2.1.1. Overview of an intensional arithmetization of metamathematics.** We now sketch some of the details of an arithmetization of metamathematics; this arithmetization can be carried out intensionally in $I\Delta_0 + \Omega_1$ and in $S^1_2$. Detailed explanations of similar arithmetizations in these theories can be found in Wilkie and Paris [1987] and in Buss [1986]. We shall always work in the (apparently) weaker theory $S^1_2$.

To arithmetize metamathematics, we need to assign Gödel numbers to syntactic objects such as 'terms', 'formulas', 'proofs', etc. Each such syntactic object is viewed as an expression consisting of string of symbols from a finite alphabet. This finite alphabet contains logical connective symbols "$\land$", "$\lor$", "$\neg$", "$\supset$", "$\exists$", "$\forall$", etc., and the comma symbol and parentheses; it also contains non-logical connective symbols for the function and relation symbols of arithmetic. The alphabet also needs symbols for variables: for this there is a variable symbol "$x$" (and possibly "$a$" for free variables) and there are symbols "0" and "1" used to write the values of subscripts of variables in binary notation. In addition, when first-order proofs are formalized in the sequent calculus, it will contain a symbol "$\longrightarrow$" for the sequent connective and the semicolon symbol for separating sequents. At times, it will be convenient to enlarge the finite alphabet with other symbols that can be used for describing the skeleton of a proof.

Since the alphabet is finite, we can identity the alphabet with some finite set $\{0,\ldots,s\}$ of integers, thereby giving each alphabet symbol $\sigma$ a *Gödel number* which is denoted $\ulcorner\sigma\urcorner$. Then, given an expression $\alpha$ which consists of the symbols $a_1a_2\cdots a_m$, let $n_1,\ldots,n_m$ be their Gödel numbers, then the least Gödel number of the sequence $\langle n_1,\ldots,n_m\rangle$ is, by definition, the *Gödel number* of the expression $\alpha$. The Gödel number of an expression $\alpha$ is denoted $\ulcorner\alpha\urcorner$. There is a subtle difference between the Godel number of symbol $\sigma$ and the Gödel number of the expression containing just $\sigma$; these are both denoted $\ulcorner\sigma\urcorner$ and it should always be clear from context which one is intended.[10]

Since we have already discussed that intensional definitions of Gödel numbers of sequences can be given; it is straightforward to further give intensional definitions of (Gödel numbers of) syntactic objects; in particular, $S^1_2$ can $\Delta^b_1$-define the following predicates:

$FreeVar(w)$ - $w$ codes a free variable

$BdVar(w)$ - $w$ codes a bound variable

$\mathit{Term}(w)$ - $w$ codes a term

$\mathit{Fmla}(w)$ - $w$ codes a formula

$\mathit{Sequent}(w)$ - $w$ codes a sequent

For example, the formula $FreeVar(w)$ asserts that either $w = \langle\ulcorner a\urcorner, \ulcorner 0\urcorner\rangle$ (which codes the free variable "$a_0$") or $w$ is of the form $\langle\ulcorner a\urcorner, \ulcorner 1\urcorner, w_1, \ldots, w_k\rangle$ with $k \geq 0$ and with each $w_i$ equal to $\ulcorner 0\urcorner$ or $\ulcorner 1\urcorner$ (this encodes "$a_i$" where $i > 0$ has binary representation $1w_1 \cdots w_k$). The $\Delta^b_1$ definitions of *Term*, *Fmla* and *Sequent* are somewhat more complicated: they depend crucially on the fact that length-bounded counting is $\Sigma^b_1$-definable and that therefore terms and formulas may be parsed by means of counting parentheses. Counting commas also allows notions such as the $i$-th formula of a sequent to be $\Sigma^b_1$-defined.

In keeping with the convention that all syntactic objects are coded by expressions, we let the Gödel number of a proof be defined to the Gödel number of an expression consisting of a sequence of sequents separated by semicolons. A proof is intended to be valid provided that each sequent in the proof can be inferred by a valid rule of inference from sequents appearing earlier in the proof. Of course the notion of *valid inference* depends on the formal proof system in which the proof is being carried out. Accordingly, for an appropriate fixed formal system $T$, we wish $S^1_2$ to be able to $\Delta^b_1$-define the predicate to define $\mathit{Proof}_T(w)$ which states that $w$ codes a valid $T$-proof. For this to be possible, there must be a polynomial time procedure which determines whether $w$ codes a valid $T$-proof; in general, this will be based on a polynomial time procedure which checks whether a given inference is valid for $T$.

The theories $T$ that we will consider, such as $S^i_2$, $T^i_2$, $I\Sigma_i$, $B\Sigma_i$, etc., will have only axioms and unary and binary inference rules, and these will be specified by a finite set of schemes. For such schematic theories, $S^1_2$ can $\Delta^b_1$-define the relation[11]

$\mathit{ValidInference}_T(u, v, w)$ - $w$ can be inferred with a single $T$-inference from zero, one or both of $u$ and $v$.

[10]We have defined $\ulcorner A\urcorner$ in a nonconventional manner: the usual definition is to let $\ulcorner A\urcorner$ represent a closed term whose value is equal to the Gödel number of $A$. We shall represent this alternative concept with the notation $\underline{\ulcorner A\urcorner}$. The definition for $\ulcorner A\urcorner$ that we are using is better for our *intensional* development.

[11]We discuss the situation for non-schematic theories below in section 2.1.3.

With this, it is easy for $S^1_2$ to $\Delta^b_1$-define $\mathit{Proof}_T(w)$. In addition, $S^1_2$ can $\Delta^b_1$-define the predicate $\mathit{Prf}_T(w,u)$ which states that $\mathit{Proof}_T(w)$ and $\mathit{Fmla}(u)$ and that $w$ is a proof of the sequent $\longrightarrow A$ where $u = \ulcorner A \urcorner$ (i.e., that $w$ is a $T$-proof of the formula $A$). Finally, the set of theorems of $T$ can be defined by

$$\mathit{Thm}_T(u) \;\Leftrightarrow\; (\exists w)\mathit{Prf}_T(w,u).$$

However, $\mathit{Thm}_T$ is not generally $\Delta^b_1$-definable, since it not generally even decidable (a consequence of Gödel's incompleteness theorems, see below).[12]

A particularly important syntactic operation is the substitution of a term into a formula. Let $A$ be a formula and let $t$ be a term; because of the sequent calculus' conventions on free and bound variables, one can always form the formula $A(t/a_0)$, which is $A$ with $t$ substituted in for the free variable $a_0$, merely by replacing each occurrence of $a_0$ as a subexpression of $A$ with the expression $t$. This is clearly $\Sigma^b_1$-definable in $S^1_2$ and $\mathit{Sub}(u,v)$ denotes the function such that $\mathit{Sub}(\ulcorner A \urcorner, \ulcorner t \urcorner) = \ulcorner A(t/a_0) \urcorner$ for all formulas $A$ and terms $t$.

One final simple, but important formalization, is the definition of closed *canonical terms* $\underline{n}$ which represent an integer $n$. The term $\underline{0}$ is just the constant symbol $0$. And inductively, the term $\underline{2m}$ is $(SS0)\cdot\underline{m}$ and the term $\underline{2m+1}$ is $\underline{2m}+S0$. Note that the number of symbols in $\underline{n}$ is $O(|n|)$; also, $S^1_2$ can $\Sigma^b_1$-define the map $n \mapsto \ulcorner \underline{n} \urcorner$.

**2.1.2. Intensionality of the arithmetization.** In order for the above-sketched arithmetization to be considered *intensional*, it is necessary that $S^1_2$ can prove basic facts about the arithmetization. To simplify notation, we shall use abbreviations such as $(\forall \ulcorner A \urcorner)(\cdots \ulcorner A \urcorner \cdots)$, which abbreviates the formula $(\forall u)(\mathit{Fmla}(u) \supset (\cdots u \cdots))$. Some examples of what $S^1_2$ can prove include:

1. $(\forall u,v,w)(Fmla(u) \land Term(v) \supset Fmla(Sub(u,v)))$.
2. $(\forall \ulcorner A \urcorner)(\forall \ulcorner B \urcorner)(\mathit{Thm}_T(\ulcorner A \urcorner) \land \mathit{Thm}_T(\ulcorner A \supset B \urcorner) \supset \mathit{Thm}_T(\ulcorner B \urcorner))$.
3. $(\forall u)(\mathit{Proof}_T(u) \supset \mathit{Thm}_{S^1_2}(\ulcorner \mathit{Proof}_T(\underline{u}) \urcorner))$.
4. $(\forall \ulcorner A \urcorner)(\forall u)(\mathit{Prf}_T(u, \ulcorner A \urcorner) \supset \mathit{Thm}_{S^1_2}(\ulcorner \mathit{Prf}(\underline{u}, \underline{\ulcorner A \urcorner}) \urcorner))$.
5. $(\forall \ulcorner A \urcorner)(\mathit{Thm}_T(\ulcorner A \urcorner) \supset \mathit{Thm}_{S^1_2}(\ulcorner \mathit{Thm}_T(\underline{\ulcorner A \urcorner}) \urcorner))$.

These five formulas require some explanation. The first just states that when a term is substituted into a formula, a formula is obtained. The second codes the fact that the consequences of the sequent calculus are closed under modus ponens. $S^1_2$ proves this by the simple argument that if there are sequent calculus proofs of $\longrightarrow A$ and $\longrightarrow A \supset B$, then these can be combined with the simple sequent calculus proof of $A, A \supset B \longrightarrow B$ using two cuts to obtain a sequent calculus proof of $B$. The third formula states that any $u$ which encodes a $T$-proof can in fact be proved to be a $T$-proof. The intuitive idea behind the fact that $S^1_2$ can prove this formula is

[12] Even for decidable theories $T$, if $T \nvDash (\forall x)(\forall y)(x = y)$, then the predicate $\mathit{Thm}_T(u)$ is PSPACE-hard.

that, given $u$ encoding a proof as a string of symbols, it is possible to construct a $S^1_2$-proof of the statement $\mathit{Proof}_T(\underline{u})$ that $u$ codes a $T$-proof. In fact, the $S^1_2$-proof of $\mathit{Proof}_T(\underline{u})$, proceeds by verifying that $u$, viewed as a string of symbols, satisfies all the properties of being a valid $T$-proof. The provability of the fourth and fifth formulas in $S^1_2$ is similar to the provability of the third.

The fact that $S^1_2$ can prove the third formula is a special case of a more general fact:

**Theorem.** (Buss [1986,Thm 7.4]) *Let $A(b)$ be a $\Sigma^b_1$-formula with only the variable $b$ free. Then, $S^1_2$ can prove*

$$(\forall u)(A(u) \supset \mathit{Thm}_{S^1_2}(\ulcorner A(\underline{u})\urcorner)).$$

**Hilbert-Bernays-Löb derivability conditions.** The following three derivability conditions, introduced by Hilbert and Bernays [1934-39] and Löb [1955], give sufficient conditions on an arithmetization for the second incompleteness theorem to hold for a theory $T$, with respect to a given formalization $\mathit{Thm}_T$ of provability:

HBL1: For all $A$ and $B$, $T \vdash \mathit{Thm}_T(\underline{\ulcorner A\urcorner}) \land \mathit{Thm}_T(\underline{\ulcorner A \supset B\urcorner}) \supset \mathit{Thm}_T(\underline{\ulcorner B\urcorner})$.
HBL2: For all $A$, if $T \vdash A$, then $T \vdash \mathit{Thm}_T(\underline{\ulcorner A\urcorner})$.
HBL3: For all $A$, $T \vdash \mathit{Thm}_T(\underline{\ulcorner A\urcorner}) \supset \mathit{Thm}_T(\,\underline{\mathit{Thm}_T(\underline{\ulcorner A\urcorner})}\,)$.

Assuming $T \supseteq S^1_2$, the first and third conditions follow from the fact that formulas 2 and 5 above are provable in $S^1_2$. The second condition is the fact that formula 5 is true; which of course is an immediate consequence of fact that formula 5 above is provable in $S^1_2$ and hence is true.

**2.1.3. Arithmetization of syntax for non-schematic theories.** So far, we have considered only schematically axiomatized theories. This is not unreasonable, since many of the theories we are interested in, such as Peano arithmetic are schematically axiomatized. Many other theories such as $S^i_2$, $I\Delta_0$, $I\Sigma_1$, etc. are not schematic but are at least nearly schematic in that they are axiomatized by a finite set of schemes with substitution restricted to certain formula classes. The metamathematics for these latter theories can be arithmetized with only a slight modification of the above methods.

A theory is said to be *axiomatizable* provided that it has a recursive (i.e., decidable) set of axioms. By a theorem of Craig's this is equivalent to having a recursively enumerable set of axioms. In general, there are many axiomatizable theories which are not schematic; nonetheless, the arithmetization of metamathematics can be modified to apply to any axiomatizable theory as follows.

Let $T$ be an axiomatizable theory. Since the predicate $\mathit{ValidInference}_T$ may no longer be polynomial-time, it may not be $\Delta^b_1$-definable in $S^1_2$. However, it is possible to express $\mathit{ValidInference}_T(u,v,w)$ in equivalent form as

$$(\exists a)\,\mathit{ValidInfEvidence}(a,u,v,w),$$

where *ValidInfEvidence* is $\Delta_1^b$ w.r.t. $S_2^1$.[13] With this, a $T$-proof is then coded as a sequence containing the lines of the proof, plus any necessary evidence values, $a$, justifying the steps in the proof. In this way, the predicates $\mathit{Proof}_T(w)$ and $\mathit{Prf}_T(w,u)$ are $\Delta_1^b$-definable in $S_2^1$; likewise, $\mathit{Thm}_T$ is definable from $\mathit{Prf}_T$ as before.

It is easy to check that formulas 1-5 are still provable in $S_2^1$. Also, if $T \supseteq S_2^1$, the Hilbert derivability conditions hold as well.

**2.1.4. Arithmetization in theories which contain only $Q$.** All our proofs of incompleteness theorems will assume that the theory $T$ under consideration contains $S_2^1$. However, the results all hold as well for theories which only contain $Q$. The intensional arithmetization in $S_2^1$ can be extended to an intensional arithmetization in $Q$ based on the fact that $S_2^1$ is interpretable in $Q$. The interpretation of $S_2^1$ in $Q$ is a very special kind based on an inductive cut $J$; namely, there is a formula $J(a)$ such that $Q$ proves $J$ is closed downwards and is closed under $0$, $S$, $+$, $\cdot$ and $\#$. In addition, for $\phi$ any sentence, the sentence $\phi^J$, *$\phi$ relativized by $J$*, is obtained from $\phi$ by replacing every quantifier $(Qx)$ with $(Qx.J(x))$. Then, we have that $Q \vdash \phi^J$ for all theorems $\phi$ of $S_2^1$.

The first use of inductive cuts for interpretations was by Solovay. The fact that $I\Delta_0$ can be interpreted in $Q$ was first discovered by Wilkie; a local interpretation of $I\Delta_0$ in $Q$ was independently discovered by E. Nelson. For more details of inductive cuts and this interpretation of $S_2^1$ in $Q$ see Theorem 4.3.3 below, or Pudlák [1983], Nelson [1986] or Chapter VIII of this volume. Pudlák [1983] gives a very general form of the interpretation of $I\Delta_0$ in $Q$.

An intensional arithmetization of metamathematics can be given in $Q$ by replacing predicates such as $\mathit{Proof}(w)$ with their relativizations $J(w) \wedge (\mathit{Proof}_T(w))^J$. The reader can check that the proofs given in the next sections all still work with these relativized predicates.

## 2.2. The Gödel incompleteness theorems

In this section, we discuss the diagonal, or fixpoint, lemma, the first and second incompleteness theorems, and Löb's theorem.

**2.2.1. The Gödel diagonal lemma.** The Gödel diagonal, or fixpoint, lemma is a crucial ingredient in the proof of the incompleteness theorems. This lemma states that, for any first-order property $A$, there is a formula $B$ that states that the property $A$ holds of the Gödel number of $B$. Thus, since we know that provability is a first-order property, it will be possible to construct a formula which asserts "I am not provable".

[13] Our formulation works for any decidable set of axioms and rules of inference; we do require always that all the usual logical axioms and rules of inference are present. A similar construction will work for inference rules with any finite number of hypotheses.

**Gödel's Diagonal Lemma.** *Let $A(a_0)$ be a formula. Then there is a formula $B$ such that $S^1_2$ proves*

$$B \leftrightarrow A(\underline{\ulcorner B \urcorner}).$$

*Furthermore, if $A$ is a $\Sigma^b_i$, $\Pi^b_i$, $\Sigma_i$ or $\Pi_i$ formula (respectively), then so is $B$; and if $A$ involves free variables other than $a_0$, then so does $B$.*

**Proof.** This proof quite simple but rather tricky and difficult to conceptualize. We first define a diagonalization function $f$ which satisfies

$$f(\ulcorner C \urcorner) \;=\; \ulcorner C(\underline{\ulcorner C \urcorner}) \urcorner$$

for all formulas $C$, where $C(\underline{\ulcorner C \urcorner})$ means $C(\underline{\ulcorner C \urcorner}/a_0)$. To define $f$, recall that the function $n \mapsto Num(n) = \ulcorner \underline{n} \urcorner$ is $\Sigma^b_1$-definable in $S^1_2$. Then the function $f$ is $\Sigma^b_1$-definable by $S^1_2$ since

$$f(x) \;=\; Sub(x, Num(x)).$$

Next, we wish to let the formula $C(a_0)$ be $A(f(a_0))$; however, since $f$ is not a function symbol in the language of $S^1_2$, we must be more careful in defining $C$. Let $f(a) = b$ be $\Sigma^b_1$-defined with the formula $G_f(a,b)$ which defines the graph of $f$ and let $t_f$ be a term such that $S^1_2$ proves $f(a) \le t_f$. Now the formula $C$ can be taken to be either

$$(\exists x \le t_f)(G_f(a_0,x) \wedge A(x)) \qquad \text{or} \qquad (\forall x \le t_f)(G_f(a_0,x) \supset A(x)).$$

(With a little more care, we can choose $C$ to be in the same quantifier complexity class as $A$.) Finally, define $B$ to be the formula $C(\underline{\ulcorner C \urcorner})$.

We claim that $S^1_2$ proves $B \leftrightarrow A(\underline{\ulcorner B \urcorner})$. The proof of this claim is almost immediate. First, by the definitions of $f$ and $B$, we have $\ulcorner B \urcorner$ is equal to $f(\ulcorner C \urcorner)$; of course $S^1_2$ proves this fact. Second, by the definition of $C$, $B$ is $S^1_2$-provably equivalent to $A(f(\underline{\ulcorner C \urcorner}))$. Therefore, $B$ is $S^1_2$-provably equivalent to $A(\underline{\ulcorner B \urcorner})$.
Q.E.D.

**2.2.2. The first incompleteness theorem.** Gödel's first incompleteness theorem states that there is no complete, axiomatizable, consistent theory $T$ extending $Q$. We shall prove several variants of this in this section.

**Definition.** Let $T$ be an axiomatizable theory. $Con_T$ is the $\forall\Delta^b_1$-formula $\neg\, Thm_T(0 \ne 0)$ which expresses the condition "*$T$ is consistent.*"

$T$ is said to be $\omega$-consistent if there does not exist a formula $B(a)$ such that $T \vdash (\exists x)B(x)$ and such that $T \vdash \neg B(\underline{n})$ for all $n \ge 0$. $T \supseteq S^1_2$ is *weakly $\omega$-consistent* provided there is no such formula $B$ which is $\Delta^b_1$ w.r.t. $S^1_2$. Since every true $\Delta^b_1$-sentence is provable in $S^1_2$, $T$ is weakly $\omega$-consistent if and only if $T$ is consistent and proves only true $\exists\Delta^b_1$-sentences.

**Gödel's First Incompleteness Theorem.** *Let $T$ be an consistent, axiomatizable theory containing $Q$. Then there is a true sentence $\phi$ such that $T \nvdash \phi$. Further, if $T$ is weakly $\omega$-consistent, then $T \nvdash \neg\phi$.*

The formula $\phi$ is explicitly constructible from the axiomatization of $T$.

**Proof.** We'll prove this theorem under the assumption that $T \supseteq S^1_2$. Choose $\phi$ to a formula such that

$$S^1_2 \vdash \phi \leftrightarrow \neg Thm_T(\underline{\ulcorner\phi\urcorner}).$$

Intuitively, the formula $\phi$ is asserting "I am not provable in $T$"; the Diagonal Lemma 2.2.1 guarantees that $\phi$ exists.

First, let's show that $\phi$ is true. Suppose $\phi$ were false. Then, by the choice of $\phi$ and since $S^1_2$ is a true theory, $T \vdash \phi$. Therefore, $S^1_2 \vdash Thm_T(\underline{\ulcorner\phi\urcorner})$; and again, by the choice of $\phi$, $S^1_2 \vdash \neg\phi$. Since $T \supseteq S^1_2$, we also have $T \vdash \neg\phi$, which contradicts the consistency of $T$. So $\phi$ cannot be false.

Second, we show that $T \nvdash \phi$. Suppose, for sake of a contradiction, $T \vdash \phi$. Then $S^1_2 \vdash Thm_T(\underline{\ulcorner\phi\urcorner})$. By choice of $\phi$, $S^1_2 \vdash \neg\phi$. So also $T \vdash \neg\phi$, which again contradicts the consistency of $T$.

Third, we assume $T$ is weakly $\omega$-consistent and prove that $T \nvdash \neg\phi$. Suppose $T$ does prove $\neg\phi$. Then since $T \supseteq S^1_2$ and by choice of $\phi$, $T \vdash Thm_T(\underline{\ulcorner\phi\urcorner})$. But $Thm_T(\underline{\ulcorner\phi\urcorner})$ is a false $\exists\Delta^b_1$-sentence, which contradicts the weak $\omega$-consistency of $T$. Q.E.D.

The obvious question at this point is whether the hypothesis of weak $\omega$-consistency can be removed from the First Incompleteness Theorem; i.e., whether there is a consistent, axiomatizable, complete theory extending $Q$. It turns out that this hypothesis can be removed:

**Rosser's Theorem.** (Rosser [1936]) *There is no consistent, axiomatizable, complete theory $T \supseteq Q$.*

The proof of this theorem will give a constructive method of obtaining a formula $\phi$ from the axiomatization of a consistent theory $T$ such that $\phi$ is independent of $T$.

**Proof.** As before, we give the proof assuming $T \supseteq S^1_2$. We need to define a modified notion of provability called "Rosser provability". Let $Neg$ be the unary function, $\Sigma^b_1$-definable in $S^1_2$, such that $Neg(\ulcorner A\urcorner) = \ulcorner\neg A\urcorner$ for all formulas $A$. Then we define the predicate $R\text{-}Prf_T(w,a)$ as

$$R\text{-}Prf_T(w,a) \;\Leftrightarrow\; Prf_T(w,a) \wedge (\forall v \le w)(\neg Prf_T(v, Neg(a))).$$

Intuitively, $A$ has a Rosser proof if and only if $A$ has a (ordinary) proof such that its negation, $\neg A$, has no smaller proof. Note that, since $T$ is consistent, $R\text{-}Prf_T(n, \ulcorner A\urcorner)$

is true exactly when $Prf_T(n, \ulcorner A \urcorner)$ is; however, this fact is not provable in $S_2^1$.[14] The predicate $R\text{-}Thm_T(u)$ is defined to be the $\exists\Pi_1^b$-formula $(\exists w)(R\text{-}Prf_T(w, u))$.

Now use the Diagonal Lemma to choose a sentence $\phi$ so that

$$S_2^1 \ \vdash \ \phi \leftrightarrow \neg R\text{-}Thm(\underline{\ulcorner \phi \urcorner}).$$

As before, we have $\phi$ is true. In fact, the same proof works as in the proof of the First Incompleteness Theorem, since $Thm_T$ and $R\text{-}Thm_T$ are extensionally equivalent. Secondly, $T \nvdash \phi$ again by the same argument as in the previous theorem; in brief: if $T \vdash \phi$, then $\phi$ is false by choice of $\phi$, but we just claimed $\phi$ is true.

Thirdly, we want to show $T \nvdash \neg\phi$. Suppose $T$ does prove $\neg\phi$; let $n$ be a Gödel number of a $T$-proof of $\neg\phi$. Since $T$ is consistent, we have that there is no $T$-proof of $\phi$; thus $S_2^1$ proves the true $\Delta_0$-sentence $(\forall v \le n)\neg Prf_T(v, \underline{\ulcorner \phi \urcorner})$. And since $Prf_T(n, Neg(\phi))$ is true, $S_2^1$ also proves $(\forall v)(v > n \supset \neg R\text{-}Prf_T(v, \underline{\ulcorner \phi \urcorner}))$. Hence, $S_2^1$ proves $\neg R\text{-}Thm_T(\underline{\ulcorner \phi \urcorner})$. By the choice of $\phi$ and since $T \supseteq S_2^1$, this implies $T \vdash \phi$, which contradicts the consistency of $T$.
Q.E.D.

**2.2.3. The second incompleteness theorem.** Gödel's second incompleteness theorem improves on his first incompleteness theorem by giving an example of a true formula with an intuitive meaning which is not provable by a decidable, consistent theory $T$. This formula is the formula $Con_T$ which expresses the consistency of $T$. Note, however, that unlike the the formula $\phi$ in the first incompleteness theorem, $Con_T$ is not necessarily independent of $T$ since there are consistent theories that prove their own inconsistency. An example of such a theory is $T + \neg Con_T$.

**Gödel's Second Incompleteness Theorem.** *Let $T$ be a decidable, consistent theory and suppose $T \supseteq Q$. Then $T \nvdash Con_T$.*

**Proof.** As usual, we assume $T \supseteq S_2^1$. Let $\phi$ be the formula from the proof of Gödel's First Incompleteness Theorem which is $S_2^1$-provably equivalent to $\neg Thm_T(\underline{\ulcorner \phi \urcorner})$. Recall that we proved $T \nvdash \phi$. We shall prove that $S_2^1$ proves $Con_T \supset \phi$; which will suffice to show that $T \nvdash Con_T$, since $T \supseteq S_2^1$.

By choice of $\phi$, $S_2^1$ proves $\neg\phi \supset Thm_T(\underline{\ulcorner \phi \urcorner})$. Also, by the formula 5 in section 2.1.2, $S_2^1$ proves $Thm_T(\underline{\ulcorner \phi \urcorner}) \supset Thm_T(\underline{\ulcorner Thm_T(\underline{\ulcorner \phi \urcorner}) \urcorner})$. Also, by choice of $\phi$ and by formula 1 of section 2.1.2, $S_2^1$ proves $Thm_T(\underline{\ulcorner Thm_T(\underline{\ulcorner \phi \urcorner}) \urcorner}) \supset Thm_T(\underline{\ulcorner \neg\phi \urcorner})$. Putting these together shows that $S_2^1$ proves that

$$\neg\phi \supset [Thm_T(\underline{\ulcorner \phi \urcorner}) \wedge Thm_T(\underline{\ulcorner \neg\phi \urcorner})].$$

From whence $\neg\phi \supset \neg Con_T$ is easily proved. Therefore, $S_2^1$ proves $Con_T \supset \phi$. $\square$

[14]This is a good example (see Feferman [1960]) of an extensional definition which is not an intensional definition. For consistent theories $T$, $R\text{-}Prf_T$ and $R\text{-}Thm_T$ provide an extensionally correct definition of provability, since $R\text{-}Prf_T(n, \ulcorner A \urcorner)$ has the correct truth value for all particular $n$ and $\ulcorner A \urcorner$. However, they are not intensionally correct; since, in general, $T$ cannot prove that $R\text{-}Thm_T(\ulcorner A \urcorner)$ and $R\text{-}Thm_T(\ulcorner A \supset B \urcorner)$ implies $R\text{-}Thm_T(B)$.

The formula $\phi$ not only implies $Con_T$, but is actually $S^1_2$-equivalent to $Con_T$. For this, note that since $\phi$ implies $\neg Thm_T(\underline{\ulcorner\phi\urcorner})$, it can be proved in $S^1_2$ that $\phi$ implies $\neg Thm_T(\underline{\ulcorner 0=1\urcorner})$. (Since if a contradiction is provable, then every formula is provable.)

**2.2.4. Löb's theorem.** The self-referential formula constructed for the proof of the First and Second Incompleteness Theorems asserted "I am not provable". A related problem would be to consider formulas which assert "I *am* provable". As the next theorem shows, such formulas are necessarily provable. In fact, if a formula is implied by its provability, then the formula is already provable. This gives a strengthening of the Second Incompleteness Theorem, which implies that, in order to prove a formula $A$, one is not substantially helped by the assuming that $A$ is provable. More precisely, the assumption $Thm_T(\underline{\ulcorner A\urcorner})$ will not significantly aid a theory $T$ in proving $A$.

**Löb's Theorem.** *Let $T \supseteq Q$ be an axiomatizable theory and $A$ be any sentence. If $T$ proves $Thm_T(\underline{\ulcorner A\urcorner}) \supset A$, then $T$ proves $A$.*

**Proof.** As usual, we assume $T \supseteq S^1_2$. Let $T'$ be the axiomatizable theory $T \cup \{\neg A\}$. The proof of Löb's Theorem uses the fact that $T'$ is consistent if and only if $T \nvdash A$; and furthermore, that $S^1_2$ proves $Con(T')$ is equivalent to $\neg Thm_T(\underline{\ulcorner A\urcorner})$. From these considerations, the proof is almost immediate from the second incompleteness theorem. Namely, since $T$ proves $\neg A \supset \neg Thm_T(\underline{\ulcorner A\urcorner})$ by choice of $A$, $T$ also proves $\neg A \supset Con(T')$. Therefore, by the Deduction Theorem, $T' \vdash Con(T')$ so by Gödel's Second Incompleteness Theorem, $T'$ is inconsistent, i.e., $T \vdash A$.

**2.2.5. Further reading.** The above material gives only an introduction to the incompleteness theorems. Other significant aspects of incompleteness include: (1) the strength of reflection principles which state that the provability of a formula implies the truth of the formula, see, e.g., Smorynski [1977]; (2) provability and interpretability logics, for which see Boolos [1993], Lindström [1997], and Chapter VII of this handbook; and (3) concrete, combinatorial examples of independence statements, such as the Ramsey theorems shown by Paris and Harrington [1977] to be independent of Peano arithmetic.

## 3. On the strengths of fragments of arithmetic

### 3.1. Witnessing theorems

In section 1.2.10, it was shown that every primitive recursive function is $\Sigma_1$-definable by the theory $I\Sigma_1$. We shall next establish the converse which implies that the $\Sigma_1$-definable functions of $I\Sigma_1$ are precisely the primitive recursive functions. The principal method of proof is the 'witnessing theorem method': $I\Sigma_1$ provides the simplest and most natural application of the witnessing method.

**3.1.1. Theorem.** (Parsons [1970], Mints [1973] and Takeuti [1987]). *Every $\Sigma_1$-definable function of $I\Sigma_1$ is primitive recursive.*

Parsons' proof of this theorem was based on the Gödel Dialectica theorem and a similar proof is given by Avigad and Feferman in Chapter V in this volume. Takeuti's proof was based on a Gentzen-style assignment of ordinals to proofs. Mints's proof was essentially the same as the witness function proof presented next; except his proof was presented with a functional language.

**3.1.2. The Witness predicate for $\Sigma_1$-formulas.** For each $\Sigma_1$-formula $A(\vec{b})$, we define a $\Delta_0$-formula $\mathit{Witness}_A(w, \vec{b})$ which states that $w$ is a witness for the truth of $A$.

**Definition.** Let $A(\vec{b})$ be a formula of the form $(\exists x_1, \ldots, x_k)B(x_1, \ldots, x_k, \vec{b})$, where $B$ is a $\Delta_0$-formula. Then the formula $\mathit{Witness}_A(w, \vec{b})$ is is defined to be the formula

$$B(\beta(1, w), \ldots, \beta(k, w), \vec{b}).$$

If $\Delta = \Delta', A$ is a succedent, then $\mathit{Witness}_{\bigvee\Delta}(w, \vec{c})$ is defined to be

$$\mathit{Witness}_A(\beta(1, w), \vec{c}) \vee \mathit{Witness}_{\bigvee\Delta'}(\beta(2, w), \vec{c}).$$

Dually, if $\Gamma = A, \Gamma'$ is an antecedent, then $\mathit{Witness}_{\bigwedge\Gamma}$ is defined similarly as

$$\mathit{Witness}_A(\beta(1, w), \vec{c}) \wedge \mathit{Witness}_{\bigwedge\Gamma'}(\beta(2, w), \vec{c}).$$

Note the different conventions on ordering disjunctions and conjunctions; these are not intrinsically important, but merely reflect the conventions for the sequent calculus are that active formulas of strong inferences are at the beginning of an antecedent and at the end of a succedent.

It is, of course, obvious that $\mathit{Witness}_A$ is a $\Delta_0$-formula, and that $I\Delta_0$ can prove

$$A(b) \leftrightarrow (\exists w)\,\mathit{Witness}_A(w, \vec{b}).$$

**3.1.3. Proof.** (Sketch of the proof of Theorem 3.1.1.) Suppose $I\Sigma_1$ proves $(\forall x)(\exists y)A(x, y)$ where $A \in \Sigma_1$. Then there is a sequent calculus proof $P$ in the theory $I\Sigma_1$ of the sequent $(\exists y)A(c, y)$. We must prove that there is a primitive recursive function $f$ such that $A(n, f(n))$ is true, in the standard integers, for all $n \geq 0$. In fact, we shall prove more than this: we will prove that there is a primitive recursive function $f$, with a $\Sigma_1$-definition in $I\Sigma_1$, such that $I\Sigma_1$ proves $(\forall x)A(x, f(x))$. This will be a corollary to the next lemma.

**Witnessing Lemma for $I\Sigma_1$.** *Let $\Gamma$ and $\Delta$ be cedents of $\Sigma_1$-formulas and suppose $I\Sigma_1$ proves the sequent $\Gamma \longrightarrow \Delta$. Then there is a function $h$ such that the following hold:*

(1) *$h$ is $\Sigma_1$-defined by $I\Sigma_1$ and is primitive recursive, and*
(2) *$I\Sigma_1$ proves*

$$(\forall \vec{c})(\forall w)[\,Witness_{\bigwedge\Gamma}(w,\vec{c}) \supset Witness_{\bigvee\Delta}(h(w,\vec{c}),\vec{c})].$$

Note that Theorem 3.1.1 is an immediate corollary to the lemma, since we may take $\Gamma$ to be the empty sequent, $\Delta$ to be the sequent containing just $(\exists y)A(c,y)$, and let $f(x) = \beta(1,\beta(1,h(x)))$ where $h$ is the function guaranteed to exist by the lemma. This is because $h(x)$ will be a sequence of length one witnessing the cedent $(\exists y)A$, so its first and only element is a witness for the formula $(\exists y)A$, and the first element of that is a value for $y$ that makes $A$ true.

It remains to prove the Witnessing Lemma. For this, we know by the Cut Elimination Theorem 1.4.2, that there is a free-cut free proof $P$ of the sequent $\Gamma \longrightarrow \Delta$ in the theory $I\Sigma_1$; in this proof, every formula in every sequent can be assumed to be a $\Sigma_1$-formula. Therefore, we may prove the Witnessing Lemma by induction on the number of steps in the proof $P$.

The base case is where there are zero inferences in the proof $P$ and so $\Gamma \longrightarrow \Delta$ is an initial sequent. Since the initial sequents allowed in an $I\Sigma_1$ proof contain only atomic formulas, the Witnessing Lemma is trivial for this case.

For the induction step, the argument splits into cases, depending on the final inference of the proof. There are a large number of cases, one for each inference rule of the sequent calculus; for brevity, we present only three cases below and leave the rest for the reader.

For the first case, suppose the final inference of the proof $P$ is an $\exists$*:right* inference, namely,

$$\dfrac{\Gamma \longrightarrow \Delta, A(t)}{\Gamma \longrightarrow \Delta, (\exists x)A(x)}$$

Let $c$ be the free variables in the upper sequent. The induction hypothesis gives a $\Sigma_1$-defined, primitive recursive function $g(w,\vec{c})$ such that $I\Sigma_1$ proves

$$Witness_{\bigwedge\Gamma}(w,\vec{c}) \longrightarrow Witness_{\bigvee\{\Delta,A(t)\}}(g(w,\vec{c}),\vec{c}).$$

In order for $Witness_{\bigvee\{\Delta,A(t)\}}(g(w,\vec{c}),\vec{c})$ to hold, either $\beta(2,g(w,\vec{c}))$ witnesses $\bigvee\Delta$ or $\beta(1,g(w,\vec{c}))$ witnesses $A(t)$. So letting $h(w,\vec{c})$ be $\Sigma_1$-defined by

$$h(w,\vec{c}) \;=\; \langle\langle t(\vec{c})\rangle * \beta(1,g(w,\vec{c})), \beta(2,g(w,\vec{c}))\rangle,$$

where $*$ denotes sequence concatenation. It is immediate from the definition of *Witness* that

$$Witness_{\bigwedge\Gamma}(w,\vec{c}) \longrightarrow Witness_{\bigvee\{\Delta,(\exists x)A(x)\}}(h(w,\vec{c}),\vec{c}).$$

For the second case, suppose the final inference of the proof $P$ is an $\exists$*:left* inference, namely,

$$\dfrac{\begin{array}{c}\ddots\vdots\cdot^{\cdot^{\cdot}}\\ A(b),\Gamma\longrightarrow\Delta\end{array}}{(\exists x)A(x),\Gamma\longrightarrow\Delta}$$

where $b$ is an eigenvariable which occurs only as indicated. The induction hypothesis gives us a $\Sigma_1$-defined, primitive recursive function $g(w,\vec{c},b)$ such that $I\Sigma_1$ proves

$$\mathit{Witness}_{\bigwedge\{A(b),\Gamma\}}(w,\vec{c})\longrightarrow \mathit{Witness}_{\bigvee\Delta}(g(w,\vec{c},b),\vec{c}).$$

Let $\mathit{tail}(w)$ be the $\Sigma_1$-defined function so that $\mathit{tail}(\langle w_0,w_1,\ldots,w_n\rangle)=\langle w_1,\ldots,w_n\rangle$. Letting $h(w,\vec{c})$ be the function $g(\langle \mathit{tail}(\beta(1,w)),\beta(2,w)\rangle,\vec{c},\beta(1,\beta(1,w)))$, it is easy to check that $h$ satisfies the desired conditions of the Witnessing Lemma.

For the third case, suppose the final inference of $P$ is a $\Sigma_1$-IND inference:

$$\dfrac{\begin{array}{c}\ddots\vdots\cdot^{\cdot^{\cdot}}\\ A(b),\Gamma\longrightarrow\Delta,A(Sb)\end{array}}{A(0),\Gamma\longrightarrow\Delta,A(t)}$$

where $b$ is the eigenvariable and does not occur in the lower sequent. The induction hypothesis gives a $\Sigma_1$-defined, primitive recursive function $g(w,\vec{c},b)$ such that $I\Sigma_1$ proves

$$\mathit{Witness}_{\bigwedge\{A(b),\Gamma\}}(w,\vec{c},b)\longrightarrow \mathit{Witness}_{\bigvee\{\Delta,A(Sb)\}}(g(w,\vec{c},b),\vec{c},b).$$

Let $k(\vec{c},v,w)$ be defined as

$$k(\vec{c},v,w) \;=\; \begin{cases} v & \text{if } \mathit{Witness}_{\bigvee\{\Delta\}}(v,\vec{c}) \\ w & \text{otherwise}\end{cases}$$

Since $\mathit{Witness}$ is a $\Delta_0$-predicate, $k$ is $\Sigma_1$-defined by $I\Sigma_1$. Now define the primitive recursive function $f(w,\vec{c},b)$ by

$$\begin{aligned} f(w,\vec{c},0) &= \langle\beta(1,w),0\rangle \\ f(w,\vec{c},b+1) &= \langle\beta(1,g(\langle\beta(1,f(w,\vec{c},b)),\beta(2,w)\rangle,\vec{c},b)), \\ &\qquad k(\vec{c},\beta(2,f(w,\vec{c},b)),\beta(2,g(\langle\beta(1,f(w,\vec{c},b)),\beta(2,w)\rangle,\vec{c},b)))\rangle \end{aligned}$$

By Theorem 1.2.10, $f$ is $\Sigma_1$ definable by $I\Sigma_1$, and since $f$ may be used in induction formulas, $\Sigma_1$ can prove

$$\mathit{Witness}_{\bigwedge\{A(0),\Gamma\}}(w,\vec{c})\longrightarrow \mathit{Witness}_{\bigvee\{\Delta,A(b)\}}(f(w,\vec{c},b),\vec{c},b).$$

using $\Sigma_1$-IND with respect to $b$. Setting $h(w,\vec{c})=f(w,\vec{c},t)$ establishes the desired conditions of the Witnessing Lemma.
Q.E.D. Witnessing Lemma and Theorem 3.1.1.

**3.1.4. Corollary.** *The $\Delta_1$-definable predicates of $I\Sigma_1$ are precisely the primitive recursive predicates.*

**Proof.** Corollary 1.2.10 already established that every primitive recursive predicate is $\Delta_1$-definable by $I\Sigma_1$. For the converse, suppose $A(c)$ and $B(c)$ are $\Sigma_1$-formulas such that $I\Sigma_1$ proves $(\forall x)(A(x) \leftrightarrow \neg B(x))$. Then the characteristic function of the predicate $A(c)$ is $\Sigma_1$-definable in $I\Sigma_1$ since $I\Sigma_1$ can prove

$$(\forall x)(\exists! y)[(A(x) \wedge y = 0) \vee (B(x) \wedge y = 1)].$$

By Theorem 3.1.1, this characteristic function is primitive recursive, hence so is the predicate $A(c)$.

**3.1.5. Total functions of $I\Sigma_n$.** Theorem 1.2.1 provided a characterization of the $\Sigma_1$-definable functions of $I\Sigma_1$ as being precisely the primitive recursive functions. It is also possible to characterize the $\Sigma_1$-definable functions of $I\Sigma_n$ for $n > 1$ in terms of computational complexity; however, the $n > 1$ situation is substantially more complicated. This problem of characterizing the provably total functions of fragments of Peano arithmetic is classically one of the central problems of proof theory; and a number of important and elegant methods are available to solve it. Space prohibits us from explaining these methods, so we instead mention only a few references.

The first method of analyzing the strength of fragments of Peano is based on Gentzen's assignment of ordinals to proofs; Gentzen [1936,1938] used Cantor normal form to represent ordinals less than $\epsilon_0$ and gave a constructive method of assigning ordinals to proofs in such a way that allowed cuts and inductions to be removed from $PA$-proofs of sentences. This can then be used to characterize the primitive recursive functions of fragments of Peano arithmetic in terms of recursion on ordinals less than $\epsilon_0$. The textbooks of Takeuti [1987] and Girard [1987] contain descriptions of this approach. A second version of this method is based on the infinitary proof systems of Tait: Chapter III of this volume describes this for Peano arithmetic, and Chapter IV describes extensions of this ordinal assignment method to much stronger second-order theories of arithmetic. The books of Schütte [1977] and Pohlers [1980] also describe ordinal assignments and infinitary proofs for strong theories of arithmetic. A further use of ordinal notations is to characterize natural theories of arithmetic in terms of transfinite induction.

A second approach to analyzing the computational strength of theories of arithmetic is based on model-theoretic constructions; see Paris and Harrington [1977], Ketonen and Solovay [1981], Sommer [1990], and Avigad and Sommer [1997].

A third method is based on the Dialectica interpretation of Gödel [1958] and on Howard's [1970] assignment of ordinals to terms that arise in the Dialectica interpretation. Chapter V of this volume discusses the Dialectica interpretation.

A fourth method, due to Ackermann [1941] uses an ordinal analysis of $\epsilon$-calculus proofs.

More recently, Buss [1994] has given a characterization of the provably total functions of the theories $I\Sigma_n$ based on an extension of the witness function method used above.

### 3.2. Witnessing theorem for $S^i_2$

Theorem 1.3.4.1 stated that every polynomial time function and every polynomial time predicate is $\Sigma^b_1$-definable or $\Delta^b_1$-definable (respectively) by $S^1_2$. More generally, Theorem 1.3.6 stated that every $\square^p_i$-function and every $\Delta^p_i$-predicate is $\Sigma^b_i$-definable or $\Delta^b_i$-definable by $S^i_2$. The next theorem states the converse; this gives a precise characterization of the $\Sigma^b_1$-definable functions of $S^1_2$ and of the $\Sigma^b_i$-definable functions of $S^i_2$ in terms of their complexity in the polynomial hierarchy. The most interesting case is probably the base case $i=1$, where $S^1_2$ is seen to have proof-theoretic strength that corresponds precisely to polynomial time.

**Theorem.** (Buss [1986])
(1) *Every $\Sigma^b_1$-definable function of $S^1_2$ is polynomial time computable.*
(2) *Let $i \geq 1$. Every $\Sigma^b_i$-definable function of $S^i_2$ is in the $i$-th level, $\square^p_i$, of the polynomial hierarchy.*

**Corollary.** (Buss [1986])
(1) *Every $\Delta^b_1$-definable predicate of $S^1_2$ is polynomial time.*
(2) *Let $i \geq 1$. Every $\Delta^b_i$-definable predicate of $S^i_2$ is in the $i$-th level, $\Delta^p_i$, of the polynomial hierarchy.*

The corollary follows from the theorem by exactly the same argument as was used to prove Corollary 3.1.4 from Theorem 3.1.1. To prove the theorem, we shall use a witnessing argument analogous to the one use for $I\Sigma_1$ above. First, we need a revised form of the *Witness* predicate; unlike the usual definition of the *Witness* predicate for bounded arithmetic formulas, we define the *Witness* predicate only for prenex formulas, since this provides some substantial simplification. This simplification is obtained without loss of generality since every $\Sigma^b_i$-formula is logically equivalent to a $\Sigma^b_i$-formula in prenex form.

**3.2.1. Definition.** Fix $i \geq 1$. Let $A(\vec{c})$ be a $\Sigma^b_i$-formula which is in prenex form. Then $Witness^i_A(w,\vec{c})$ is defined by induction on the complexity of $A$ as follows:
(1) If $A$ is a $\Pi^b_{i-1}$-formula, then $Witness^i_A(w,\vec{c})$ is just the formula $A(\vec{c})$,
(2) If $A(\vec{c})$ is not in $\Pi^b_{i-1}$ and is of the form $(\exists x \leq t)B(\vec{c},x)$, then $Witness^i_A(w,\vec{c})$ is the formula

$$Witness^i_{B(\vec{c},b)}(\beta(2,w),\vec{c},\beta(1,w)) \wedge \beta(1,w) \leq t.$$

Intuitively, a witness for $(\exists x \leq t)B(x)$ is a pair $w$, the first element of the pair giving a value for $x$ and the second element witnessing the truth of $B(x)$ for that value of $x$.
(3) If $A(\vec{c})$ is not in $\Pi^b_{i-1}$ and is of the form $(\forall x \leq |t|)B(\vec{c},x)$, then $Witness^i_A(w,\vec{c})$ is the formula

$$(\forall x \leq |t|)\, Witness^i_{B(\vec{c},d)}(\beta(x+1,w),\vec{c},x).$$

Intuitively, a witness for $(\forall x \leq |t|)B(x)$ is a sequence $w$ of length $|t|+1$, $w = \langle w_0, w_1, \ldots, w_{|t|}\rangle$ such that each $w_x$ witnesses the truth of $B(x)$, for $0 \leq x \leq |t|$.

**Lemma.** *Let $i \geq 1$ and $A \in \Sigma_i^b$. Then*
(a) *$Witness_A^i$ is $\Delta_i^b$ with respect to $S_2^1$. If $i > 1$, then $Witness_A^i$ is a $\Pi_{i-1}^b$-formula.*
(b) *$Witness_A^i$ defines a $\Delta_i^p$-predicate.*
(c) *$S_2^i$ proves*

$$A(\vec{c}) \leftrightarrow (\exists w)\, Witness_A^i(w, \vec{c}).$$

(d) *There is a term $t_A$ and a polynomial time, $\Sigma_1^b$-definable function $g_A$ such that $S_2^1$ proves*

$$Witness_A^i(w, \vec{c}) \supset g_A(w) < t_A(\vec{c}) \wedge Witness_A^i(g_A(w), \vec{c}).$$

The lemma is easily proved by induction on the complexity of $A$. For part (d), the function $g_A(w)$ merely computes a succinct Gödel number of $w$; this just involves removing unnecessary leading zeros and removing unnecessary elements from the witness.

We extend the witness predicate to cedents of prenex form formulas as follows. If $\Delta = \Delta', A$ is a succedent, then $Witness^i_{\bigvee\Delta}(w, \vec{c})$ is defined to be

$$Witness_A^i(\beta(1, w), \vec{c}) \vee Witness^i_{\bigvee\Delta'}(\beta(2, w), \vec{c}).$$

Dually, if $\Gamma = A, \Gamma'$ is an antecedent, then $Witness^i_{\bigwedge\Gamma}$ is defined similarly as

$$Witness_A^i(\beta(1, w), \vec{c}) \wedge Witness^i_{\bigwedge\Gamma'}(\beta(2, w), \vec{c}).$$

**3.2.2. Proof.** (Proof sketch for Theorem 3.2.) We shall prove Theorem 3.2 by proving a slightly more general witnessing lemma that applies to sequents of $\Sigma_i^b$-formulas. Although the lemma holds for sequents of general $\Sigma_i^b$-formulas, we state it only for formulas in prenex form, since this simplifies the *Witness* predicate and the proof.

**Witnessing Lemma for $S_2^i$.** *Let $i \geq 1$. Let $\Gamma \longrightarrow \Delta$ be a sequent of formulas in $\Sigma_i^b$ in prenex form, and suppose $S_2^i$ proves $\Gamma \longrightarrow \Delta$. Let $\vec{c}$ include all free variables in the sequent. Then there is a $\square_i^p$-function $h(w, \vec{c})$ which is $\Sigma_i^b$-defined in $S_2^i$ such that $S_2^i$ proves*

$$Witness^i_{\bigwedge\Gamma}(w, \vec{c}) \longrightarrow Witness^i_{\bigvee\Delta}(h(w, \vec{c}), \vec{c}).$$

The proof of the Witnessing Lemma is by induction on the number of sequents in a free-cut free proof $P$ of $\Gamma \longrightarrow \Delta$. Since every $\Sigma_i^b$-formula is equivalent to a $\Sigma_i^b$-formula in prenex form, we may assume w.l.o.g. that every induction formula in the free-cut free proof $P$ is a prenex form $\Sigma_i^b$-formula. Then, by the subformula property, every formula appearing anywhere in the proof is also a $\Sigma_i^b$-formula in prenex form. The base case of the induction proof is when $\Gamma \longrightarrow \Delta$ is an initial sequent; in this case, every formula in the sequent is atomic, so the Witnessing Lemma trivially holds. The induction step splits into cases depending on the final inference of the proof. The structural inferences and the propositional inferences are essentially trivial, the latter because of our assumption that all formulas are in prenex form. So it remains to consider the quantifier inferences and the induction inferences. The cases where the final inference of $P$ is an $\exists\le$*:right* inference or an $\exists\le$*:left* inference are similar to the $\exists$*:right* and $\exists$*:left* cases of the proof of the Witnessing Lemma 3.1.3 for $I\Sigma_i$, so we omit these cases too.

Now suppose the final inference of $P$ is a $\Sigma_i^b$-PIND inference:

$$\frac{\begin{array}{c}\ddots\vdots\cdot^{\cdot^{\cdot}}\\ A(\lfloor\frac{1}{2}b\rfloor),\Gamma\longrightarrow\Delta,A(b)\end{array}}{A(0),\Gamma\longrightarrow\Delta,A(t)}$$

where $A\in\Sigma_i^b\setminus\Pi_{i-1}^b$ and where $b$ is the eigenvariable and does not occur in the lower sequent. The induction hypothesis gives a $\Sigma_i^b$-defined, $\square_i^p$-function $g(w,\vec{c},b)$ such that $S_2^i$ proves

$$\mathit{Witness}^i_{\bigwedge\{A(\lfloor\frac{1}{2}b\rfloor),\Gamma\}}(w,\vec{c},b)\longrightarrow\mathit{Witness}^i_{\bigvee\{\Delta,A(b)\}}(g(w,\vec{c},b),\vec{c},b).$$

Let $k(\vec{c},v,w)$ be defined as

$$k(\vec{c},v,w)\;=\;\begin{cases}v & \text{if } \mathit{Witness}^i_{\bigvee\Delta}(v,\vec{c})\\ w & \text{otherwise}\end{cases}$$

Since $\mathit{Witness}^i$ is a $\Delta_i^b$-predicate, $k$ is $\Sigma_i^b$-defined by $S_2^i$; and since $\mathit{Witness}^i$ is in $\Delta_i^p$, $k$ is in $\square_i^p$. Define the $\square_i^p$-function $f(w,\vec{c},b)$ by

$$\begin{aligned}
f(w,\vec{c},0) &= \langle\beta(1,w),0\rangle\\
f(w,\vec{c},b) &= \big\langle\beta(1,g(\langle\beta(1,f(w,\vec{c},\lfloor\tfrac{1}{2}b\rfloor)),\beta(2,w)\rangle,\vec{c},b)),\\
&\qquad k(\vec{c},\beta(2,f(w,\vec{c},\lfloor\tfrac{1}{2}b\rfloor)),\beta(2,g(\langle\beta(1,f(w,\vec{c},\lfloor\tfrac{1}{2}b\rfloor)),\beta(2,w)\rangle,\vec{c},b)))\big\rangle\\
&\qquad \text{for } b>0.
\end{aligned}$$

Since $f$ is defined by limited recursion on notation from $g$, and since $g$ is $\Sigma_i^b$-defined by $S_2^i$, $f$ is also $\Sigma_i^b$-defined by $S_2^i$. Therefore, $f$ may be used in induction formulas and $S_2^i$ can prove

$$\mathit{Witness}^i_{\bigwedge\{A(0),\Gamma\}}(w,\vec{c})\longrightarrow\mathit{Witness}^i_{\bigvee\{\Delta,A(b)\}}(f(w,\vec{c},b),\vec{c},b).$$

using $\Sigma_i^b$-PIND with respect to $b$. Setting $h(w,\vec{c})=f(w,\vec{c},t)$ establishes the desired conditions of the Witnessing Lemma.

Finally, we consider the inferences involving bounded universal quantifiers. The cases where the principal formula of the inference is a $\Pi_{i-1}^b$-formula are essentially trivial, since such formulas do not require a witness value, i.e., they are their own witnesses. This includes any inference where the principal connective is a non-sharply bounded universal quantifier. A $\forall\le$*:left* where the principal formula is in $\Sigma_i^b$ must have a sharply bounded universal quantifier as its principal connective; this case of the Witnessing Lemma is fairly simple and we leave it to the reader. Finally, we consider the case where the last inference of $P$ is a $\forall\le$*:right* inference

$$\dfrac{\begin{matrix}\ddots\vdots\;\cdot^{\cdot^{\cdot}}\\ b\le|t|,\Gamma\longrightarrow\Delta,A(b)\end{matrix}}{\Gamma\longrightarrow\Delta,(\forall x\le|t|)A(t)}$$

where $A\in\Sigma_i^b\setminus\Pi_{i-1}^b$ and where the eigenvariable $b$ occurs only as indicated. The induction hypothesis gives a $\Sigma_i^b$-defined, $\square_i^p$-function $g$ such that $S_2^i$ proves

$$\begin{aligned}&b\le|t|\wedge \mathit{Witness}^i_{\bigwedge\Gamma}(\beta(2,w),\vec{c})\longrightarrow\\ &\qquad \mathit{Witness}^i_{A(b)}(\beta(1,g(w,\vec{c},b)),\vec{c},b)\vee \mathit{Witness}^i_{\bigvee\Delta}(\beta(2,g(w,\vec{c},b)),\vec{c}).\end{aligned}$$

Let $f_1(w,\vec{c})$ be defined to equal $\beta(2,g(w,\vec{c},b))$ for the least value $b\le|t|$ such this value witnesses $\bigvee\Delta$, or if there is no such value of $b\le t$, let $f_1(w,\vec{c})=0$. Since the predicate $\mathit{Witness}^i_{\bigvee\Delta}(\beta(2,g(w,\vec{c},b)),\vec{c})$ is $\Delta_i^p$ and is $\Delta_i^b$-defined by $S_2^i$, the function $f_1$ is in $\square_i^p$ and is $\Sigma_i^b$-defined by $S_2^i$. Also, let $f_2(w,\vec{c})$ equal

$$\langle\beta(1,g(w,\vec{c},0)),\beta(1,g(w,\vec{c},1)),\beta(1,g(w,\vec{c},2)),\ldots,\beta(1,g(w,\vec{c},|t|))\rangle\,.$$

It is easy to verify that $f_2$ also is in $\square_i^p$ and is $\Sigma_i^b$-definable by $S_2^i$. Now let $h(w,\vec{c})$ equal $\langle f_2(\langle 0,w\rangle,\vec{c}),f_1(\langle 0,w\rangle,\vec{c})\rangle$. It is easy to check that $h$ satisfies the desired conditions of the Witnessing Lemma.
Q.E.D. Witnessing Lemma and Theorem 3.2.

### 3.3. Witnessing theorems and conservation results for $T_2^i$

This section takes up the question of the definable functions of $T_2^i$. For these theories, there are three witnessing theorems, one for each of the $\Sigma_i^b$-definable, the $\Sigma_{i+1}^b$-definable and the $\Sigma_{i+2}^b$-definable functions. In addition, there is a close connection between $S_2^{i+1}$ and $T_2^i$; namely, the former theory is conservative over the latter. We'll present most of these results without proof, leaving the reader to look up the original sources.

The results stated in this section will apply to $T_2^i$ for $i\ge0$; however, for $i=0$, the bootstrapping process does not allow us to introduce many simple functions. Therefore, when $i=0$, instead of $T_2^0$, we must use the theory $PV_1=T_2^0(\square_1^p)$ as defined in section 1.3.7. To avoid continually treating $i=0$ as a special case, we let $T_2^0$ denote $PV_1$ for the rest of this section.

#### 3.3.1. The $\Sigma_{i+1}^b$-definable functions of $T_2^i$

**Theorem.** (Buss [1990]) *Let $i\ge0$.*
(1) *$T_2^i$ can $\Sigma_{i+1}^b$-define every $\square_{i+1}^p$-function.*

(2) *Conversely, every $\Sigma^b_{i+1}$-definable function of $T^i_2$ is a $\Box^p_{i+1}$-function.*
(3) *$S^{i+1}_2$ is $\Sigma^b_{i+1}$-conservative over $T^i_2$.*
(4) *$S^{i+1}_2$ is conservative over $T^i_2 + \Sigma^b_{i+1}$-replacement with respect to Boolean combinations of $\Sigma^b_{i+1}$ formulas.*

We'll just state some of the main ideas of the proof of this theorem, and let the reader refer to Buss [1990] or Krajíček [1995] for the details. Part (1) with $i = 0$ is trivial because of the temporary convention that $T^0_2$ denotes $PV_1$. To prove part (1) for $i > 0$, one shows that $T^i_2$ can "$Q_i$-define" every $\Box^p_{i+1}$ formula, where $Q_i$-definability is a strong form of $\Sigma^b_{i+1}$-definability. The general idea of a $Q_i$-definable function is that it is computed by a Turing machine with a $\Sigma^p_i$-oracle such that every "yes" answer of the oracle must be supported by a witness. In the correct computation, the sequence of "yes/no" answers is maximum in a lexicographical ordering, and thus $T^i_2$ can prove that the correct computation exists using $\Sigma^b_i$-MAX axioms (which can be derived similarly to minimization axioms). This proof of (1) is reminiscent of the theorem of Krentel [1988] that MINSAT is complete for $\Box^p_1$.

Part (2) of the theorem is immediate from Theorem 3.2 and the fact that $T^i_2 \subseteq S^{i+1}_2$. Part (3) is based on the following strengthening of the Witnessing Theorem 3.2.2 for $S^{i+1}_2$:

**Witnessing Lemma for $S^{i+1}_2$.** *Let $i \geq 1$. Let $\Gamma \longrightarrow \Delta$ be a sequent of formulas in $\Sigma^b_{i+1}$ in prenex form, and suppose $S^{i+1}_2$ proves $\Gamma \longrightarrow \Delta$; let $\vec{c}$ include all free variables in the sequent. Then there is a $\Box^p_{i+1}$-function $h(w, \vec{c})$ which is $Q_i$-defined in $T^i_2$ such that $T^i_2$ proves*

$$Witness^{i+1}_{\bigwedge\Gamma}(w, \vec{c}) \longrightarrow Witness^{i+1}_{\bigvee\Delta}(h(w, \vec{c}), \vec{c}).$$

The proof of this Witnessing Lemma is almost exactly the same as the proof of the Witnessing Lemma in section 3.2.2; the only difference is that the witnessing functions are now proved to be $Q_i$-definable in $T^i_2$. In fact, (1) implies the necessary functions are $Q_i$-defined by $T^i_2$ since we already know they are $\Sigma^b_{i+1}$-defined by $S^{i+1}_2$. So the main new aspect is showing that $T^i_2$ can prove that the witnessing functions work.

Part (3) of the theorem is an immediate consequence of the Witnessing Lemma. Part (4) can also be obtained from the Witnessing Lemma using the fact that $T^i_2 + \Sigma^b_{i+1}$-replacement can prove that $A(\vec{c})$ is equivalent to $(\exists w)\, Witness^{i+1}_A(w, \vec{c})$ for any $A \in \Sigma^b_{i+1}$.

### 3.3.2. The $\Sigma^b_{i+2}$-definable functions of $T^i_2$

The $\Sigma^b_{i+2}$-definable functions of $T^i_2$ can be characterized by the following theorem, due to Krajíček, Pudlák and Takeuti [1991].

**KPT Witnessing Theorem.** *Let $i \geq 0$. Suppose $T^i_2$ proves*

$$(\forall x)(\exists y)(\forall z \leq t(x))A(y, x, z)$$

*where $A \in \Pi_i^b$. Then there is a $k > 0$ and there are $\Sigma_{i+1}^b$-definable function symbols $f_1(x)$, $f_2(x, z_1), \ldots,$ $f_k(x, z_1, \ldots z_{k-1})$ such that $T_2^i$ proves*

$$\begin{aligned}(\forall x)(\forall z_1 \le t)[A(f_1(x), x, z_1) \vee (\forall z_2 \le t)[A(f_2(x, z_1), x, z_2)\\ \vee(\forall z_3 \le t)[A(f_3(x, z_1, z_2), x, z_3)\\ \vee \cdots \vee (\forall z_k \le t)[A(f_k(x, z_1, \ldots, z_{k-1}), x, z_k)] \cdots]]]\end{aligned}$$

*Conversely, whenever the above formula is provable, then $T_2^i$ can also prove $(\forall x)(\exists y)(\forall z \le t)A(y, x, z)$.*

The variables $x$, $y$ and $z$ could just as well have been vectors of variables, since the replacement axioms and sequence coding can be used to combine adjacent like quantifiers. Also, the first half of the theorem holds even if $t$ involves both $x$ and $y$. The proof of the KPT Witnessing Theorem is now quite simple: by the discussion in section 1.3.7, we can replace each $T_2^i$ by its conservative, universally axiomatized extension $PV_{i+1}$, and now the theorem is an immediate corollary of the corollary to the generalized Herbrand's theorem in section 2.5.3 of Chapter I.

**3.3.2.1. Applications to the polynomial hierarchy.** The above theorem has had a very important application in showing an equivalence between the collapse of the hierarchy of theories of bounded arithmetic and the (provable) collapse of the polynomial time hierarchy. This equivalence was first proved by Krajíček, Pudlák and Takeuti [1991]; we state two improvements to their results. (We continue the convention that $T_1^0$ denotes $PV_1$.)

**Theorem.** (Buss [1995], Zambella [1996]) *Let $i \ge 0$. If $T_2^i \models S_2^{i+1}$, then (1) $T_2^i = S_2$ and therefore $S_2$ is finitely axiomatized, and (2) $T_2^i$ proves the polynomial hierarchy collapses, and in fact, (2.a) $T_2^i$ proves that every $\Sigma_{i+3}^b$-formula is equivalent to a Boolean combination of $\Sigma_{i+2}^b$-formulas and (2.b) $T_2^i$ proves the polynomial time hierarchy collapses to $\Sigma_{i+1}^p/poly$.*

**Corollary.** *$S_2$ is finitely axiomatized if and only if $S_2$ proves the polynomial hierarchy collapses.*

Let $g(x)$ be a $\Sigma_1^b$-definable function of $T_2^i$ such that for each $n > 0$ there is an $m > 0$ so that $T_2^i \vdash (\forall x)(x > n \supset g(x) > m)$ (for example, $g(n) = |n|$ or $g(n) = ||n||$, etc.) Let $g\Sigma_i^b$-IND denote the axioms

$$A(0) \wedge (\forall x)(A(x) \supset A(x+1)) \supset (\forall z \le g(x))A(z).$$

Let $\forall\Pi_i^b(\mathbb{N})$ denote the set of all $\forall\Pi_i^b$ sentences (in the language of $S_2$) true about the standard integers.

**3.3.2.2. Theorem.** (essentially Krajíček, Pudlák and Takeuti [1991])
*If $T_2^i + \forall\Pi_i^b(\mathbb{N}) \vDash g\Sigma_{i+1}^b$-IND, then the polynomial time hierarchy collapses to $\Delta_{i+1}^p/poly$.*

Note that second theorem differs from the first in that there is no mention of the provability of the collapse of the polynomial hierarchy; on the other hand, the second theorem states a stronger collapse. Krajíček, Pudlák and Takeuti [1991] prove the second theorem with $g(n) = |n|$ and without the presence of $\forall\Pi_i^b(\mathbb{N})$: their proof gives the stronger form stated here with only minor modifications.

### 3.3.3. The $\Sigma_1^b$-definable functions of $T_2^1$

Buss and Krajíček [1994] characterize the $\Sigma_1^b$-definable functions of $T_2^1$ as being precisely the functions which are projections of PLS functions.

**Polynomial Local Search.** Johnson, Papadimitriou and Yannakakis [1988] defined a *Polynomial Local Search* problem (PLS-problem) $L$ to be a maximization problem satisfying the following conditions: (we have made some inessential simplifications to their definition)

(1) For every instance $x \in \{0,1\}^*$, there is a set $F_L(x)$ of *solutions,* an integer valued cost function $c_L(s,x)$ and a neighborhood function $N_L(s,x)$,

(2) The binary predicate $s \in F_L(x)$ and the functions $c_L(s,x)$ and $N_L(s,x)$ are polynomial time computable. There is a polynomial $p_L$ so that for all $s \in F_L(x)$, $|s| \le p_L(|x|)$. Also, $0 \in F_L(x)$.

(3) For all $s \in \{0,1\}^*$, $N_L(s,x) \in F_L(x)$.

(4) For all $s \in F_L(x)$, if $N_L(s,x) \ne s$ then $c_L(s,x) < c_L(N_L(s,x),x)$.

(5) The problem is solved by finding a locally optimal $s \in F_L(x)$, i.e., an $s$ such that $N_L(s,x) = s$.

It follows from these conditions that all $s \in F_L(x)$ are polynomial size.

A PLS-problem $L$ can be expressed as a $\Pi_1^b$-sentence saying that the conditions above hold; if these are provable in $T_2^1$ then we say $L$ is a *PLS-problem in $T_2^1$*. The formula $Opt_L(x,s)$ is the $\Delta_1^b$-formula $N_L(s,x) = s$. A multivalued function $g$ such that for all $x$, $N_L(g(x),x) = g(x)$, is called a *PLS function*; $g$ must be total, but may be multivalued, since there may exist more than one optimal cost solution. The next theorem states, loosely speaking, that the (multivalued) $\Sigma_1^b$-definable functions of $T_2^1$ are precisely the functions $f$ which can be expressed in the form $f = \pi \circ g$, where $g$ is a PLS function and where $\pi$ is a polynomial time function (in fact, $\pi(y) = \beta(1,y)$ can always be used).

**Theorem.** (Buss and Krajíček [1994])

(1) *For every PLS problem $L$, $T_2^1$ can prove $(\forall x)(\exists y)Opt_L(x,y)$.*

(2) *If* $A \in \Sigma_1^b$ *and if* $T_2^1$ *proves* $(\forall \vec{x})(\exists y)A(\vec{x}, y)$, *then there is a polynomial time (projection) function* $\pi(y)$ *and a PLS problem* $L$ *such that* $T_2^1$ *proves*

$$(\forall \vec{x})(\forall y)(Opt_L(\vec{x}, y) \supset A(\vec{x}, \pi(y)).$$

*In other words, if* $g$ *is a PLS function solving* $L$, *then* $A(\vec{x}, \pi \circ g(\vec{x}))$ *holds for all* $\vec{x}$ *and all values of* $g(\vec{x})$.

**Natural Proofs.** The above theorem characterizing the $\Sigma_1^b$ consequences of $T_2^1$ in terms of PLS functions was used in an important way to establish the independence of some computational complexity conjectures from $S_2^2(\alpha)$. Razborov and Rudich [1994] introduced a notion of "P-natural proofs" of $P \neq NP$; which intuitively are proofs which provide a polynomial time method of separating out truth tables of Boolean functions that do not have polynomial size circuits. They then showed that under a certain strong pseudo-random number generator conjecture (henceforth: the SPRNG conjecture) that there cannot be P-natural proofs of $P \neq NP$. Razborov [1995] then showed that $S_2^2(\alpha)$ cannot prove superpolynomial lower bounds on the size of circuits for predicates in the polynomial hierarchy unless there are P-natural proofs that $P \neq NP$. This latter condition of course implies the SPRNG conjecture is false; however, most researchers in cryptography apparently do believe the SPRNG conjecture. Thus commonly believed cryptographic conjectures imply that $S_2^2(\alpha)$ cannot prove superpolynomial lower bounds for NP predicates. A further observation of Widgerson is that $S_2^2$ cannot prove the SPRNG conjecture. Razborov's proof used the conservativity of $S_2^2$ over $T_2^1$, and the above characterization of the $\Sigma_1^b$-consequences of $T_2^1$; he then combined this with a communication complexity result (analogous to Craig interpolation) to extract a P-natural proof from the resulting PLS function.

Razborov [1994] has subsequently given a simpler proof of the above-discussed theorem which uses the translations from bounded arithmetic into propositional logic (see Chapter VIII of this volume) plus interpolation theorems for propositional logic. A complete account of this simpler proof can be found in our survey article, Buss [1997].

### 3.4. Relationships between $B\Sigma_n$ and $I\Sigma_n$

Recall from section 1.2.9, that $B\Sigma_{n+1} \vdash I\Sigma_n \vdash B\Sigma_n$. We show in the next paragraphs that these three theories are distinct and that $B\Sigma_{n+1}$ is conservative over $I\Sigma_n$.

**3.4.1. Conservation of $B\Sigma_{n+1}$ over $I\Sigma_n$.** In this section we outline a proof of the well-known theorem that the $B\Sigma_{n+1}$ is $\Pi_{n+2}$-conservative over $I\Sigma_n$; this was first proved by Parsons [1970]. A model-theoretic proof was later given by Paris and Kirby [1978], and we sketch below a proof-theoretic proof from Buss [1994].

**Theorem.** *$B\Sigma_{n+1}$ is $\Pi_{n+2}$-conservative over $I\Sigma_n$.*

Recall that $B\Sigma_{n+1}$ is equivalent to the theory $B\Pi_n$, which has $\Pi_n$-REPL axioms of the form

$$(\forall x \le t)(\exists y)A(x,y) \longrightarrow (\exists z)(\forall x \le t)(\exists y \le z)A(x,y)$$

where $A \in \Pi_n$. In the above sequent, there are unbounded quantifiers in the scope of bounded quantifiers, so the formula in the antecedent is a $\Sigma^+_{n+1}$-formula, not a $\Sigma_{n+1}$-formula.

**Definition.** Fix $n$ and suppose $A \in \Sigma^+_{n+1}$.
(1) If $A \in \Pi^+_n$, then $A^{\leqslant s}$ is defined to be $A$.
(2) If $A$ is $(\exists x)B$ and $A \notin \Pi^+_n$, then $A^{\leqslant s}$ is defined to be $(\exists x \le s)B$.
(3) If $A$ is $(Qx \le t)B$ then $A^{\leqslant s}$ is defined to be $(Qx \le t)(B^{\leqslant s})$.
Let $\Gamma \longrightarrow \Delta$ be a sequent $A_1, \ldots, A_k \longrightarrow B_1, \ldots, B_\ell$ of $\Sigma^+_{n+1}$-formulas. Then $\Gamma^{\leqslant s}$ is the formula $\bigwedge_{i=1}^k A_i^{\leqslant s}$ and $\Delta^{\leqslant s}$ is the formula $\bigvee_{j=1}^{\ell} B_j^{\leqslant s}$. This notation should cause no confusion since antecedents and succedents are always clearly distinguished.

If $\vec{c} = c_1, \ldots c_s$ is a vector of free variables, then $\vec{c} \le u$ abbreviates the formula $c_1 \le u \wedge \cdots \wedge c_s \le u$. $(\forall \vec{c} \le u)$ and $(\exists \vec{c} \le u)$ abbreviate the corresponding vectors of bounded quantifiers.

**Lemma.** *Let $n \ge 1$. Suppose $\Gamma \longrightarrow \Delta$ is a sequent of $\Sigma^+_{n+1}$-formulas that is provable in $B\Sigma_{n+1}$. Let $\vec{c}$ include all the free variables occurring in $\Gamma \longrightarrow \Delta$. Then*

$$I\Sigma_n \vdash (\forall u)(\exists v)(\forall \vec{c} \le u)\Big(\Gamma^{\leqslant u} \supset \Delta^{\leqslant v}\Big).$$

Intuitively, this theorem is saying that given a bound $u$ on the sizes of the free variables and on the sizes of the witnesses for the formulas in $\Gamma$, there is a bound $v$ for the values of a witness for a formula in $\Delta$. The conservation theorem above is an immediate corollary of the lemma, so it remains to prove the lemma.

**Proof.** We give only a short sketch of the proof of the lemma here; a more detailed version is given in Buss [1994] although the definitions are slightly different there.

Firstly, formulate $B\Pi_n$-proofs in the sequent calculus, using the inference rule form of the $\Pi_n$-REPL axioms described in section 1.4.5. Secondly, since $B\Pi_n = B\Sigma_{n+1}$, we may assume there is a $B\Pi_n$-proof $P$ of $\Gamma \longrightarrow \Delta$, and by the Free-cut Elimination Theorem, we may assume that every formula appearing in $P$ is a $\Sigma^+_{n+1}$-formula. Thirdly, we shall use induction on the number of sequents in $P$ to prove that the Lemma holds for every sequent in $P$. The induction step involves a number of cases; we shall do only the two cases where the final inference of $P$ is a replacement inference and where the final inference of $P$ is a $\forall$*:right* inference. The latter case is the hardest of all the cases; the rest of the cases are left to the reader.

Suppose the final inference of $P$ is a replacement inference:

$$\frac{\begin{array}{c}\ddots\vdots\cdot^{\cdot}\\ \Gamma \longrightarrow \Delta, (\forall x \le t)(\exists y)A(x,y)\end{array}}{\Gamma \longrightarrow \Delta, (\exists z)(\forall x \le t)(\exists y \le z)A(x,y)}$$

The induction hypothesis is that $I\Sigma_n$ proves

$$(\forall u)(\exists v)(\forall \vec{c} \le u)[\Gamma^{\le u} \supset \Delta^{\le v} \vee (\forall x \le t)(\exists y \le v)A(x,y)].$$

From this, the desired result that

$$(\forall u)(\exists v)(\forall \vec{c} \le u)[\Gamma^{\le u} \supset \Delta^{\le v} \vee (\exists z \le v)(\forall x \le t)(\exists y \le z)A(x,y)].$$

follows immediately.

Now suppose that $P$ ends with a $\forall$*:right* inference:

$$\frac{\begin{array}{c}\ddots\vdots\cdot^{\cdot}\\ \Gamma \longrightarrow \Delta, B(\vec{c},d)\end{array}}{\Gamma \longrightarrow \Delta, (\forall x)B(\vec{c},x)}$$

Note that $B \in \Pi_n^+$ since $(\forall x)B$ is a $\Sigma_{n+1}^+$-formula. We reason inside $I\Sigma_n$. Let $u$ be arbitrary. By strong $\Sigma_n$-replacement (see the end of section 1.2.9) there is a $u' \ge u$ such that

$$(\forall \vec{c} \le u)\Big((\forall x)B(\vec{c},x) \leftrightarrow (\forall x \le u')B(\vec{c},u')\Big).$$

Let $v \ge u'$ be given by the induction hypothesis so that

$$(\forall \vec{c}, d \le u')\Big(\Gamma^{\leqslant u'} \supset \Delta^{\leqslant v} \vee B(\vec{c},d)\Big).$$

Now let $\vec{c} \le u$ be arbitrary such that $\Gamma^{\leqslant u}$. We need to show $\Delta^{\leqslant v} \vee (\forall x)B(\vec{c},x)$. Suppose that $(\forall x)B(\vec{c},x)$ is false: then there is a $d \le u'$ such that $\neg B(\vec{c},d)$, and by the induction hypothesis, $\Delta^{\leqslant v}$ holds. Thus $\Delta^{\leqslant v} \vee (\forall x)B(\vec{c},x)$ holds.

### 3.4.2. $I\Sigma_{n+1}$ properly contains $B\Sigma_{n+1}$

**Theorem.** (Parsons [1970]) *Let $n \ge 1$. $I\Sigma_{n+1}$ is not equal to $B\Sigma_{n+1}$.*

**Proof.** We'll give only a quick sketch of a proof-theoretic proof based on Gödel's second incompleteness theorem; see Paris and Kirby [1978] for a model-theoretic proof. The two main steps in our proof are:

(1) $I\Sigma_1 \vdash \mathit{Con}(I\Sigma_n) \supset \mathit{Con}(B\Sigma_{n+1})$. This is proved by formalizing, in $I\Sigma_1$, the proof of Theorem 3.4.1 sketched above. That proof was quite constructive: any $B\Sigma_{n+1}$ proof of a $\Sigma_n$-formula can be transformed into a free-cut free proof by a primitive recursive process. Then the transformation of the free-cut free $B\Sigma_{n+1}$-proof into a $I\Sigma_n$-proof, as in the proof of Lemma 3.4.1, is primitive recursive (in fact it is polynomial time).

Therefore, $I\Sigma_1$ proves that if $B\Sigma_{n+1}$ proves a contradiction, $0 = 1$, then so does $I\Sigma_n$; i.e., $I\Sigma_1$ proves that if $B\Sigma_{n+1}$ is inconsistent, then so is $I\Sigma_n$.

(2) $I\Sigma_{n+1} \vdash Con(I\Sigma_n)$. To prove this, first note that, since $I\Sigma_1$ can prove the free-cut elimination theorem, it is sufficient to prove that $I\Sigma_{n+1}$ can prove that there is no free-cut free $I\Sigma_n$-proof of $0 = 1$; in particular, it suffices to show that $I\Sigma_{n+1}$ can prove that there is no $I\Sigma_n$ sequent calculus proof of $0 = 1$ in which every formula is a $\Sigma_n$-formula. Second, $I\Sigma_{n+1}$ has a truth definition for $\Sigma_n$-formulas; i.e., there is a formula $Tr(x, y)$ such that when $x$ is the Gödel number of a $\Sigma_n$-formula and $y$ is a sequence encoding values for the free variables of the formula, then $Tr(x, y)$ defines the truth of the formula for those values. In addition, $I\Sigma_{n+1}$ can prove that the truth definition satisfies the usual properties of truth, in that it obeys the meanings of the logical connectives. Chapter VIII discusses truth definitions in depth, and the reader should refer to that for more details. Third, using the truth definition for $\Sigma_n$-formulas, $I\Sigma_{n+1}$ can prove, by induction on the number of lines in the free-cut free $I\Sigma_n$-proof, that every sequent in the proof is valid. Therefore, it cannot be a proof of $0 = 1$, since that is not valid. So by this means, $I\Sigma_{n+1}$ proves the consistency of $I\Sigma_n$.

(1) and (2) immediately that $I\Sigma_{n+1}$ proves the consistency of $B\Sigma_{n+1}$; therefore, by Gödel's incompleteness theorem, $I\Sigma_{n+1}$ is not equal to $B\Sigma_{n+1}$.

**3.4.3.** $B\Sigma_{n+1}$ **properly contains** $I\Sigma_n$**.** The fact that $I\Sigma_n$ does not prove the $\Sigma_{n+1}$-replacement axioms was first established independently by Lessan [1978] and Paris and Kirby [1978]. Their proofs were model-theoretic; Kaye [1993] gave a proof-theoretic proof based on an argument analogous to the proof of Theorem 3.3.2.1 using a Herbrand-style nocounterexample interpretation.

## 4. Strong incompleteness theorems for $I\Delta_0 + \exp$

**4.1.** Gödel's Second Incompleteness Theorem states that a sufficiently strong, axiomatized, consistent theory $T$ cannot prove its own consistency. One way to strengthen this incompleteness theorem is by working with two theories, $S$ and $T$, such that $S$ is a subtheory of $T$: in some cases, one can establish that $T$ cannot prove the consistency of its subtheory $S$.

There are many cases in which this strengthening of the second incompleteness theorem can be achieved. One important situation is where $T$ is conservative over $S$; for example, $B\Sigma_{n+1}$ cannot prove the consistency of $I\Sigma_n$, since $B\Sigma_{n+1}$ is conservative over $I\Sigma_n$ and the latter theory cannot prove its own consistency. A second important example is where $S$ is interpretable in $T$ and thus $Con(S) \supset Con(T)$; for example, it is known that $S_2$ is interpretable in $Q$ (see Wilkie and Paris [1987] and Nelson [1986]) and therefore $S_2$ cannot prove $Con(Q)$.

A third example, and the one that is the main subject of this section, is the theorem of Wilkie and Paris [1987] that $I\Delta_0 + \exp$ cannot prove the consistency of $Q$. This example does not fall into either of the above examples, since $I\Delta_0 + \exp$ is not interpretable in $Q$.

**4.2.** Before beginning a discussion of the proof that $I\Delta_0 + \exp$ does not prove $Con(Q)$, we discuss a few other extensions of the second incompleteness theorem. The first extension is that the second incompleteness theorem applies also to restricted notions of provability, such as "bounded consistency" and "$\Sigma_k$-consistency".

**Definition.** Let $S$ be a theory, formalized in the sequent calculus. We say that $S$ is *bounded consistent* if there is no $S$-proof of the empty sequent $\longrightarrow$ in which only bounded formulas appear. For $k \geq 0$, $S$ is $\Sigma_k$-consistent provided there is no $S$-proof of the the empty sequent in which only $\Sigma_k$-formulas appear. $S$ is *free-cut free consistent* if there is no formula $A$ such that $S$ has free-cut free proofs of both $\longrightarrow A$ and $A \longrightarrow$.

The formulas $BdCon(S)$, $Con_{\Sigma_k}(S)$ and $FCFCon(S)$ are $\forall\Pi^b_1$-formulas which express the bounded consistency, the $\Sigma_k$-consistency and the free-cut free consistency of $S$, respectively.

Of course, the cut-elimination theorem implies that a bounded theory $S$ satisfies these three notions of consistency if and only if $S$ is consistent in the usual sense; however, since the cut-elimination theorem is not provable in weak theories where the superexponentiation function is not provably total, these three new notions of consistency will not generally be provably equivalent to each other or to $Con(S)$.

**Definition.** We say that a proof is *bounded* provided every formula in the proof is $\Delta_0$. Similarly a proof is $\Sigma_k$, if every formula in the proof is in $\Sigma_k$. We write $S \vdash_{\overline{\Delta_0}} A$ and $S \vdash_{\overline{\Sigma_k}} A$ to denote the condition that $A$ has a sequent calculus $S$-proof in which every formula is in $\Delta_0$ or $\Sigma_k$, respectively.

Buss [1986] proved that if $S$ is a bounded theory (such as $I\Delta_0$, $S^i_2$, $T^i_2$, $S_2$, etc) then $S$ cannot prove its own free-cut free consistency; i.e., $S$ cannot prove $FCFCon(S)$ and hence $S \nvdash BdCon(S)$ and $S \nvdash \Sigma_k\text{-}Con$ for all $k > 0$. This was later strengthened to show that $S_2$ does not prove $BdCon(BASIC)$, the bounded consistency of its induction-free base theory: this result first appeared in Takeuti [1990] and Buss and Ignjatović [1995] building on the earlier work of Pudlák [1990]. A related result, proved by Buss and Ignjatović [1995], is that the theory $PV$ (and hence $S^1_2$) does not prove the consistency of a finitely axiomatized, induction-free fragment $PV^-$ of $PV$.

Like the theorem of Wilkie and Paris [1987] that we discuss below, these independence results provide situations where a stronger theory cannot prove a consistency statement about a weaker theory. These results are interesting in their own right of course; but in addition, they are motivated by a yet-unfulfilled hope that independence results of these kinds could lead to a proof that $P \neq NP$. This wistful hope is based on the intuitive idea that $P \neq NP$ is analogous to a finitary incompleteness theorem.

**4.2.1.** More strengthenings of Gödel's second incompleteness theorem can be found in Chapters VII and VIII of this volume.

### 4.3. A theorem of Wilkie and Paris

This section outlines a proof of the theorem that $I\Delta_0 + \exp$ cannot prove the consistency of $I\Delta_0$. Earlier, we used the notation $exp(x, y, z)$ as a $\Delta_0$-predicate expressing the condition that $x^y = z$. We also define "exp" to be the sentence

$$(\forall x)(\forall y)(\exists z) exp(x, y, z)$$

stating that the exponentiation function is total.

**Theorem.** (Wilkie and Paris [1987]) $I\Delta_0 + \exp$ *cannot prove* $Con(I\Delta_0)$. *Therefore,* $I\Delta_0 + \exp$ *cannot prove* $Con(Q)$.

It is worth noting some theories that are sufficiently strong to prove the consistency of $I\Delta_0$. First, if one considers bounded consistency, we have that

$$I\Delta_0 + \exp \vdash BdCon(I\Delta).$$

To prove this fact, one shows that $I\Delta_0 + \exp$ can define a truth definition for bounded formulas which is sufficient to allow $I\Delta_0 + \exp$ to prove the validity of every sequent which has a bounded $\Delta_0$-proof.

Second, let $2^x_k$ be the superexponentiation function defined by $2^x_0 = x$ and $2^x_{i+1} = 2^{2^x_i}$. By the bootstrapping techniques used earlier, there is a $\Delta_0$-formula $superexp(i, x, z)$ which intensionally expresses $2^x_i = z$. Similarly, $superexp(i, x, z)$ is $\Delta^b_1$-definable with respect to $S^1_2$. We let "superexp" be the axiom stating

$$(\forall x)(\forall y)(\exists z) superexp(x, y, z).$$

Since $I\Delta_0 + \text{superexp}$ can prove the free-cut elimination theorem, it can also prove that $BdCon(I\Delta_0)$ implies $Con(I\Delta_0)$. Therefore, $I\Delta_0 + \text{superexp} \vdash Con(I\Delta_0)$.

**4.3.1.** We are now ready to outline the proof of Theorem 4.3. It is more convenient to work with $S_2$ instead of $I\Delta_0$ and so we shall prove that $S_2 + \exp \nvdash Con(S_2)$. Note that we still have $S_2 + \exp \vdash BdCon(S_2)$. Also, $S^1_2$ (and $I\Delta_0 + \Omega_1$) proves $Con(Q) \supset Con(S_2)$, so $Con(S_2)$ and $Con(I\Delta_0)$ are equivalent.[15] We shall prove Theorem 4.3, by proving a series of lemmas, theorems and corollaries, namely, Lemma 4.3.2 through Theorem 4.3.10. The proof is a modified version of the original proof of Wilkie and Paris [1987].

**4.3.2. Lemma.** *Let* $\phi(x)$ *be a* $\Sigma_1$*-formula, and suppose* $S_2 + \exp \vdash (\forall x)\phi(x)$. *Then there is a constant* $k > 0$ *such that*

$$S_2 \vdash (\forall x)(2^x_k \text{ exists} \supset \phi(x)).$$

To improve readability, we shall often write, as above, a shorthand notation such as "$2^x_k$ exists" as an abbreviation for $(\exists y) superexp(k, x, y)$.

[15]The proof below that $S_2 + \exp \nvdash Con(S_2)$ can be understood without knowing how to prove in $S^1_2$ that $I\Delta_0$ and $S_2$ are equiconsistent.

**Proof.** Without loss of generality, $\phi(x)$ is of the form $(\exists u)\phi_M(x,u)$ with $\phi_M \in \Delta_0$. The hypothesis implies that $S_2$ proves $(\forall y)(\exists z)(2^y = z) \supset (\forall x)\phi(x)$, which can be put in prenex form as

$$(\forall x)(\exists u)(\exists y)(\forall z)[2^y \neq z \vee \phi_M(x,u)].$$

Now, momentarily enlarge $S_2$ to have Skolem functions for all $\Delta_0$ formulas, thereby making $S_2$ axiomatized by purely universal formulas. The strong form of Herbrand's theorem (section 2.5.3 of Chapter I) implies that there is an $\ell > 0$ and there are terms $s_1(u)$, $t_1(x)$, $s_2(x,u_1,y_1)$, $t_2(x,u_1,y_1),\ldots,$ $s_\ell(x,u_1,\ldots,u_{\ell-1},y_1,\ldots,y_{\ell-1})$, $t_\ell(x,u_1,\ldots,u_{\ell-1},y_1,\ldots,y_{\ell-1})$ so that $S_2$ proves

$$(\forall x)[(\forall z_1)[2^{t_1(x)} \neq z_1 \vee \phi_M(x,s_1(x)) \vee (\forall z_2)[2^{t_2(x,z_1)} \neq z_2 \vee \phi_M(x,s_2(x,z_1)) \vee$$
$$\cdots \vee (\forall z_\ell)[2^{t_2(x,z_1,\ldots,z_{\ell-1})} \neq z_\ell \vee \phi_M(x,s_2(x,z_1,\ldots,z_{\ell-1}))]\cdots]]].$$

Since $\neg\phi \supset \neg\phi_M(x,s_i(x,\vec{z}))$, we immediately have that $S_2$ also proves

$$(\forall x)[\phi(x) \vee (\forall z_1)(2^{t_1(x)} \neq z_1 \vee (\forall z_2)(2^{t_2(x,z_1)} \neq z_2 \vee$$
$$\cdots \vee (\forall z_\ell)(2^{t_\ell(x,z_1,\ldots,z_{\ell-1})} \neq z_\ell)\cdots))]$$

where each $t_i$ is a function with polynomial growth rate with graph defined by a $\Delta_0$-formula. Thus, $S_2$ proves that $\phi(x)$ holds, provided there exists $z_1 = 2^{t_1(x)}$, $z_2 = 2^{t_2(x,z_1)},\ldots,$ $z_\ell = 2^{t_\ell(x,z_1,\ldots,z_{\ell-1})}$. Since each $t_i$ has polynomial growth rate, the values of the $z_i$'s are bounded by $2^x_{\ell+1}$ for sufficiently large $x \in \mathbb{N}$; therefore, $S_2$ proves that if $2^x_{\ell+1}$ exists, then $\phi(x)$ holds. Taking $k = \ell+1$, Lemma 4.3.2 is proved.

**4.3.3. Theorem.** (Solovay [1976]) *For each $n,k \geq 0$, there is a $S^1_2$-proof $P$ of $(\exists x)(superexp(\underline{k},\underline{n},x))$ with size polynomially bounded in terms of $|n|$ and $k$. In addition, $P$ is a $\Sigma_{2k+1}$-proof.*

**Proof.** The proof is based on using formulas that define inductive cuts. The particular ones we need are formulas $J_i(x)$ and $K_i(x)$ defined as:

$$\begin{aligned}
J_0(x) &\Leftrightarrow 0=0 \quad \text{(always true)}\\
K_0(x) &\Leftrightarrow (\exists y)(2^x = y)\\
J_{i+1}(x) &\Leftrightarrow (\forall z)(K_i(z) \supset K_i(z+x))\\
K_{i+1}(x) &\Leftrightarrow (\exists y)(2^x = y \wedge J_{i+1}(y))
\end{aligned}$$

**Lemma.**
(a) $S^1_2 \vdash J_k(0)$
(b) $S^1_2 \vdash J_k(x) \supset J_k(x+1)$

(c) $S^1_2 \vdash J_k(x) \wedge u < x \supset J_k(u)$

(d) $S^1_2 \vdash J_k(x) \supset J_k(x+x)$

(e) $S^1_2 \vdash K_k(0)$

(f) $S^1_2 \vdash K_k(x) \wedge u < x \supset K_k(u)$

(g) $S^1_2 \vdash K_k(x) \supset K_k(x+1)$

(h) $S^1_2 \vdash K_k(x) \supset (\exists z) superexp(\underline{k+1}, x, z)$.

Parts (a)-(g) are proved simultaneously by induction on $k$. Part (h) is likewise proved using induction on $k$. Moreover, it is easy to verify that the $S_2$-proofs of formulas (a)-(g) are polynomial size in $k$, and involve only $\Sigma_{2k+1}$-formulas.

By using (d) and (c) of the lemma, it is straightforward now to give find an $S^1_2$-proof of $J_k(\underline{n})$ of size polynomial in $|n|$ and $k$; from this, (e) and (h) give the desired proof of $P$ of $(\exists z) superexp(\underline{k}, \underline{n}, z)$.

**4.3.4. Lemma.** *Suppose $\phi(x) \in \Sigma_1$ and $S_2 + \exp \vdash (\forall x)\phi(x)$. Then, there is a $k \geq 0$ such that*

$$S^1_2 \ \vdash\ (\forall x)\left(S_2 \mathrel{\vdash_{\overline{\Sigma_k}}} \phi(\underline{x})\right).$$

Lemma 4.3.4 is proved from Lemma 4.3.2 by formalizing the argument of Lemma 4.3.3 in $S^1_2$.

**4.3.5. Lemma.** *Let $\phi(x)$ be a $\forall\Pi^b_1$-formula, which is without loss of generality of the form $(\forall y)\phi_M(x,y)$ where $\phi_M \in \Pi^b_1$. Then there is a term $t$ such that*

$$S^1_2 \ \vdash\ \neg\phi(x) \longrightarrow \left(S_2 \mathrel{\vdash_{\overline{\Delta_0}}} (\exists y \leq t(x))\neg\phi_M(\underline{x}, y)\right).$$

This lemma is a special case of Theorem 2.1.2.

**4.3.6. Lemma.** *Let $\phi$ be a $\forall\Pi^b_1$-sentence such that $S_2 + \exp \vdash \phi$. Then there is a $k \geq 0$ such that*

$$S^1_2 \ \vdash\ \neg\phi \longrightarrow \neg Con_{\Sigma_k}(S_2).$$

**Proof.** Without loss of generality, $\phi$ is of the form $(\forall x)\phi_M(x)$ with $\phi_M$ a $\Pi^b_1$-formula. By Lemma 4.3.4, $S_2$ proves $(\forall x)(S_2 \mathrel{\vdash_{\overline{\Sigma_k}}} \phi_M(\underline{x}))$. On the other hand, Lemma 4.3.5 implies that $S_2$ proves

$$\neg\phi_M(x) \supset \left(S_2 \mathrel{\vdash_{\overline{\Delta_0}}} \neg\phi_M(\underline{x})\right).$$

These two facts suffice to prove Lemma 4.3.6.

**4.3.7. Lemma.** *Let $k > 0$. Then $S_2 + \exp$ proves $Con_{\Sigma_k}(S_2)$.*

**Proof.** (Sketch). The proof of this has two main steps:

(1) Firstly, one shows that $S_2 + \exp$ proves $BdCon(S_2) \supset Con_{\Sigma_k}(S_2)$. This is done, by formalizing the following argument: (a) Assume that $P$ is a $\Sigma_k$-proof of $0 = 1$ in the theory $S_2$. (b) By using sequence encoding to collapse adjacent like quantifiers, we may assume w.l.o.g. that each formula in $P$ has at most $k+1$ unbounded quantifiers. (c) By applying the process used to prove the Cut-Elimination Theorem 2.4.2 of Chapter I, there is a bounded $S_2$-proof of $0 = 1$ of size at most $2^{||P||}_{2k+4}$. Since only finitely many iterations of exponentiation are needed, the last step can be formalized in $S_2 + \exp$.

(2) Secondly, one shows that $S_2 + \exp$ can prove the bounded consistency of $S_2$. The general idea is that if there is a bounded $S_2$-proof $P$ of $0 = 1$, then there is a a fixed value $\ell$ so that all variables appearing in $P$ can be implicitly bounded by $L = 2^{size(P)}_{\ell}$ where $size(P)$ is the number of symbols in $P$. (In fact, $\ell = 3$ works.) Once all variables are bounded by $L$, a truth definition can be given based on the fact that $2^{L^{size(P)}}$ exists. With this truth definition, $S_2 + \exp$ can prove that every sequent in the $S_2$-proof is valid.

**4.3.8. Corollary.** *The theory* $S_2 + \exp$ *is conservative over the theory* $S_2 \cup \{Con_{\Sigma_k}(S_2) : k \geq 0\}$ *with respect to* $\forall\Pi^b_1$*-consequences.*

**Proof.** The fact that the first theory includes the second theory is immediate from Theorem 4.3.7. The conservativity is immediate from Lemma 4.3.6.

Incidentally, since $S_2$ is globally interpretable in $Q$, we also have that the theories $S_2 + \{Con_{\Sigma_k}(S_2) : k \geq 0\}$ and $S_2 + \{Con_{\Sigma_k}(Q) : k \geq 0\}$ are equivalent.

**4.3.9. Theorem.** $S_2 \cup \{Con_{\Sigma_k}(S_2) : k \geq 0\} \nvdash Con(S_2)$.

It is an immediate consequence of Theorem 4.3.9 and Corollary 4.3.8 that $S_2 + \exp \nvdash Con(S_2)$, which is the main result we are trying to establish. So it remains to prove Theorem 4.3.9:

**Proof.** Let $k > 0$ be fixed. Use Gödel's Diagonal Lemma to choose an $\exists\Sigma^b_1$-sentence $\phi_k$ such that

$$S^1_2 \vdash \phi_k \leftrightarrow \left(S_2 + Con_{\Sigma_k}(S_2) \vdash_{\overline{\Delta_0}} \neg\phi_k\right).$$

Now $\phi_k$ is certainly false, since otherwise $S_2 + Con_{\Sigma_k}(S_2)$ proves $\neg\phi_k$, which would be false if $\phi_k$ were true. Furthermore, $S_2 + \exp$ can formalize the previous sentence, since as sketched above in the part (2) of the proof of Theorem 4.3.7, $S_2 + \exp$ can prove the validity of every formula appearing in a bounded proof in the theory $S_2 + Con_{\Sigma_k}(S_2)$. Therefore, $S_2 + \exp$ proves $\neg\phi_k$.

Since $\neg\phi_k$ is a $\forall\Pi^b_1$-sentence, Corollary 4.3.8 implies that there is some $m > 0$ such that $S_2 + Con_{\Sigma_m}(S_2) \vdash \neg\phi_k$. It is evident that $S_2 + Con_{\Sigma_k}(S_2)$ cannot prove $Con_{\Sigma_m}(S_2)$ since this would contradict the fact that $\phi_k$ is false, which implies that

$S_2 + Con_{\Sigma_k}(S_2)$ does not prove $\neg\phi_k$. Therefore $S_2 + Con_{\Sigma_k}(S_2)$ also cannot prove $Con(S_2)$.

Since this argument works for arbitrary $k$, the Compactness Theorem implies that $S_2 + \{Con_{\Sigma_k}(Q) : k \geq 0\}$ cannot prove $Con(S_2)$.

The proofs above actually proved something slightly stronger than Theorem 4.3.9; namely,

**4.3.10. Theorem.** *There is an $m = O(k)$ such that $S_2 + Con_{\Sigma_k}(S_2) \nvdash Con_{\Sigma_m}(S_2)$.*

We conjecture that $m = k + 1$ also works, but do not have a proof at hand.

**4.3.11.** A related result, which was stated as an open problem by Wilkie and Paris [1987] and was later proved by Hájek and Pudlák [1993,Coro. 5.34], is the fact that there is a $\forall\Pi_1^b$-sentence $\phi$, such that $S_2^1 + \exp \vdash \phi$ but such that $S_k^1 \nvdash \phi$ for all $k \geq 0$.

**Acknowledgements.** We are grateful to J. Avigad, C. Pollett, and J. Krajíček for suggesting corrections to preliminary versions of this chapter. Preparation of this article was partially supported by NSF grant DMS-9503247 and by cooperative research grant INT-9600919/ME-103 of the NSF and the Czech Republic Ministry of Education.

## References

W. Ackermann
[1941] Zur Widerspruchsfreiheit der Zahlentheorie, *Mathematische Annallen*, 117, pp. 162–194.

J. Avigad and R. Sommer
[1997] A model-theoretic approach to ordinal analysis, *Bulletin of Symbolic Logic*, 3, pp. 17–52.

J. Barwise
[1977] *Handbook of Mathematical Logic*, North-Holland, Amsterdam.

J. H. Bennett
[1962] *On Spectra*, PhD thesis, Princeton University.

G. Boolos
[1989] A new proof of the Gödel incompleteness theorem, *Notices of the American Mathematical Society*, 36, pp. 388–390.
[1993] *The Logic of Provability*, Cambridge University Press.

S. R. Buss
[1986] *Bounded Arithmetic*, Bibliopolis, Napoli. Revision of 1985 Princeton University Ph.D. thesis.
[1990] Axiomatizations and conservation results for fragments of bounded arithmetic, in: *Logic and Computation, proceedings of a Workshop held Carnegie-Mellon University, 1987*, W. Sieg, ed., vol. 106 of Contemporary Mathematics, American Mathematical Society, Providence, Rhode Island, pp. 57–84.
[1992] A note on bootstrapping intuitionistic bounded arithmetic, in: *Proof Theory: A selection of papers from the Leeds Proof Theory Programme 1990*, P. H. G. Aczel, H. Simmons, and S. S. Wainer, eds., Cambridge University Press, pp. 149–169.

[1994] The witness function method and fragments of Peano arithmetic, in: *Proceedings of the Ninth International Congress on Logic, Methodology and Philosophy of Science, Uppsala, Sweden, August 7-14, 1991*, D. Prawitz, B. Skyrms, and D. Westerståhl, eds., Elsevier, North-Holland, Amsterdam, pp. 29–68.

[1995] Relating the bounded arithmetic and polynomial-time hierarchies, *Annals of Pure and Applied Logic*, 75, pp. 67–77.

[1997] Bounded arithmetic and propositional proof complexity, in: *Logic of Computation*, H. Schwichtenberg, ed., Springer-Verlag, Berlin, pp. 67–121.

S. R. Buss and A. Ignjatović

[1995] Unprovability of consistency statements in fragments of bounded arithmetic, *Annals of Pure and Applied Logic*, 74, pp. 221–244.

S. R. Buss and J. Krajíček

[1994] An application of Boolean complexity to separation problems in bounded arithmetic, *Proceedings of the London Mathematical Society*, 69, pp. 1–21.

G. J. Chaitin

[1974] Information-theoretic limitations of formal systems, *J. Assoc. Comput. Mach.*, 21, pp. 403–424.

P. Clote

[1985] Partition relations in arithmetic, in: *Methods in Mathematical Logic*, C. A. Di Prisco, ed., Lecture Notes in Computer Science #1130, Springer-Verlag, Berlin, pp. 32–68.

A. Cobham

[1965] The intrinsic computational difficulty of functions, in: *Logic, Methodology and Philosophy of Science, proceedings of the second International Congress, held in Jerusalem, 1964*, Y. Bar-Hillel, ed., North-Holland, Amsterdam.

S. A. Cook

[1975] Feasibly constructive proofs and the propositional calculus, in: *Proceedings of the Seventh Annual ACM Symposium on Theory of Computing*, Association for Computing Machinery, New York, pp. 83–97.

S. A. Cook and A. Urquhart

[1993] Functional interpretations of feasibly constructive arithmetic, *Annals of Pure and Applied Logic*, 63, pp. 103–200.

S. Feferman

[1960] Arithmetization of metamathematics in a general setting, *Fundamenta Mathematicae*, 49, pp. 35–92.

H. Gaifman and C. Dimitracopoulos

[1982] Fragments of Peano's arithmetic and the MRDP theorem, in: *Logic and Algorithmic: An International Symposium held in honour of Ernst Specker*, Monographie #30 de L'Enseignement Mathématique, pp. 187–206.

G. Gentzen

[1936] Die Widerspruchsfreiheit der reinen Zahlentheorie, *Mathematische Annalen*, 112, pp. 493–565. English translation in: Gentzen [1969], pp. 132-213.

[1938] Neue Fassung des Widerspruchsfreiheitbeweis für der reinen Zahlentheorie, *Forschungen zur Logik end zur Grundlegung der exacten Wissenschaften, New Series*, 4, pp. 19–44. English translation in: Gentzen [1969], pp. 252-286.

[1969] *Collected Papers of Gerhard Gentzen*, North-Holland, Amsterdam. Edited by M. E. Szabo.

J.-Y. Girard

[1987] *Proof Theory and Logical Complexity*, vol. I, Bibliopolis, Napoli.

K. Gödel

[1958] Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes, *Dialectica*, 12, pp. 280–287.

P. Hájek and P. Pudlák

[1993] *Metamathematics of First-order Arithmetic*, Perspectives in Mathematical Logic, Springer-Verlag, Berlin.

D. Hilbert and P. Bernays

[1934-39] *Grundlagen der Mathematik, I & II*, Springer, Berlin.

W. A. Howard

[1970] Assignment of ordinals to terms for primitive recursive functionals of finite type, in: *Intuitionism and Proof Theory: Proceedings of the Summer Conference at Buffalo N.Y. 1968*, A. Kino, J. Myhill, and R. E. Vesley, eds., North-Holland, Amsterdam, pp. 443–458.

D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis

[1988] How easy is local search?, *Journal of Computer and System Science*, 37, pp. 79–100.

R. W. Kaye

[1991] *Models of Peano arithmetic*, Oxford Logic Guides #15, Oxford University Press.

[1993] Using Herbrand-type theorems to separate strong fragments of arithmetic, in: *Arithmetic, Proof Theory and Computational Complexity*, P. Clote and J. Krajíček, eds., Clarendon Press (Oxford University Press), Oxford.

C. F. Kent and B. R. Hodgson

[1982] An arithmetic characterization of NP, *Theoretical Computer Science*, 21, pp. 255–267.

J. Ketonen and R. M. Solovay

[1981] Rapidly growing Ramsey functions, *Annals of Mathematics*, 113, pp. 267–314.

J. Krajíček

[1995] *Bounded Arithmetic, Propositional Calculus and Complexity Theory*, Cambridge University Press.

J. Krajíček, P. Pudlák, and G. Takeuti

[1991] Bounded arithmetic and the polynomial hierarchy, *Annals of Pure and Applied Logic*, 52, pp. 143–153.

M. W. Krentel

[1988] The complexity of optimization problems, *Journal of Computer and System Sciences*, 36, pp. 490–509.

H. Lessan

[1978] *Models of Arithmetic*, PhD thesis, Manchester University.

P. Lindström

[1997] *Aspects of Incompleteness*, Lecture Notes in Logic #10, Springer-Verlag, Berlin.

R. J. Lipton

[1978] Model theoretic aspects of computational complexity, in: *Proceedings of the 19th Annual Symposium on Foundations of Computer Science*, IEEE Computer Society, Piscataway, New Jersey, pp. 193–200.

M. H. Löb

[1955] Solution of a problem of Leon Henkin, *Journal of Symbolic Logic*, 20, pp. 115–118.

E. Mendelson

[1987] *Introduction to Mathematical Logic*, Wadsworth & Brooks/Cole, Monterey.

G. E. Mints
[1973] Quantifier-free and one-quantifier systems, *Journal of Soviet Mathematics*, 1, pp. 71–84.

E. Nelson
[1986] *Predicative Arithmetic*, Princeton University Press.

V. A. Nepomnjaščii
[1970] Rudimentary predicates and Turing calculations, *Kibernetika*, 6, pp. 29–35. English translation in *Cybernetics* 8 (1972) 43-50.

R. Parikh
[1971] Existence and feasibility in arithmetic, *Journal of Symbolic Logic*, 36, pp. 494–508.

J. B. Paris and C. Dimitracopoulos
[1982] Truth definitions for $\Delta_0$ formulae, in: *Logic and Algorithmic, Monographie no 30 de L'Enseignement Mathematique*, University of Geneva, pp. 317–329.

J. B. Paris and L. Harrington
[1977] A mathematical incompleteness in Peano arithmetic, in: *Handbook of Mathematical Logic*, North-Holland, Amsterdam, pp. 1133–1142.

J. B. Paris and L. A. S. Kirby
[1978] $\Sigma_n$-collection schemes in arithmetic, in: *Logic Colloquium '77*, North-Holland, Amsterdam, pp. 199–210.

C. Parsons
[1970] On a number-theoretic choice schema and its relation to induction, in: *Intuitionism and Proof Theory: Proceedings of the Summer Conference at Buffalo N.Y. 1968*, A. Kino, J. Myhill, and R. E. Vesley, eds., North-Holland, Amsterdam, pp. 459–473.
[1972] On $n$-quantifier induction, *Journal of Symbolic Logic*, 37, pp. 466–482.

W. Pohlers
[1980] *Proof Theory: An Introduction*, Lecture Notes in Mathematics #1407, Springer-Verlag, Berlin.

P. Pudlák
[1983] Some prime elements in the lattice of interpretability types, *Transactions of the American Mathematical Society*, 280, pp. 255–275.
[1990] A note on bounded arithmetic, *Fundamenta Mathematicae*, 136, pp. 85–89.

A. A. Razborov
[1994] *On provably disjoint NP-pairs*, Tech. Rep. RS-94-36, Basic Research in Computer Science Center, Aarhus, Denmark, November. http://www.brics.dk/index.html.
[1995] Unprovability of lower bounds on the circuit size in certain fragments of bounded arithmetic, *Izvestiya of the RAN*, 59, pp. 201–224.

A. A. Razborov and S. Rudich
[1994] Natural proofs, in: *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing*, Association for Computing Machinery, New York, pp. 204–213.

J. B. Rosser
[1936] Extensions of some theorems of Gödel and Church, *Journal of Symbolic Logic*, 1, pp. 87–91.

K. Schütte
[1977] *Proof Theory*, Grundlehren der mathematischen Wissenschaften #225, Springer-Verlag, Berlin.

W. Sieg
[1985] Fragments of arithmetic, *Annals of Pure and Applied Logic*, 28, pp. 33–71.

C. Smorynski
[1977] The incompleteness theorems, in: Barwise [1977], pp. 821–865.

R. M. Smullyan
[1992] *Gödel's Incompleteness Theorems*, Oxford Logic Guides #19, Oxford University Press.

R. M. Solovay
[1976] *Letter to P. Hájek* Unpublished.

R. Sommer
[1990] *Transfinite Induction and Hierarchies Generated by Transfinite Recursion within Peano Arithmetic*, PhD thesis, U.C. Berkeley.

L. J. Stockmeyer
[1976] The polynomial-time hierarchy, *Theoretical Computer Science*, 3, pp. 1–22.

G. Takeuti
[1987] *Proof Theory*, North-Holland, Amsterdam, 2nd ed.
[1990] Some relations among systems for bounded arithmetic, in: *Mathematical Logic, Proceedings of the Heyting 1988 Summer School*, P. P. Petkov, ed., Plenum Press, New York, pp. 139–154.

A. Tarski, A. Mostowski, and R. M. Robinson
[1953] *Undecidable Theories*, North-Holland, Amsterdam.

A. J. Wilkie and J. B. Paris
[1987] On the scheme of induction for bounded arithmetic formulas, *Annals of Pure and Applied Logic*, 35, pp. 261–302.

C. Wrathall
[1976] Complete sets and the polynomial-time hierarchy, *Theoretical Computer Science*, 3, pp. 23–33.

D. Zambella
[1996] Notes on polynomially bounded arithmetic, *Journal of Symbolic Logic*, 61, pp. 942–966.