Bounded Arithmetic and Polynomial-time Hierarchy

Chung Tong Lee

October 31, 2007

1 Introduction

An input for a Turing Machine (TM) can be considered as a sequence of numeric values in a n-adic number system where n is the size of alphabet. A TM computation for decision problem can then be transformed as a evaluation of a numeric function whose range is $\{0,1\}$. With some basic functions, Buss [1] built a hierarchy of bounded arithmetic theories S_2^i which characterizes the polynomial hierarchy (PH) [1]. We will discuss S_2^i briefly (up to chapter 3 of [1]), with focus on S_2^1, and complete the discussion with Herbrand analyse of S_2^i [3] as an simpler alternative to witness theorem (chapter 5 in [1]).

2 Limited Iteration and Polynomial Hierarchy

Definition 1.

- *A function $f(\vec{x})$ is defined by limited recursion from $g(\vec{x})$ and $h(\vec{x},y,z)$ with time bound p and space bound q iff the followings hold*

$$\begin{aligned}\tau(\vec{x},0) &= g(\vec{x},0)\\ \tau(\vec{x},y') &= h(\vec{x},y,\tau(\vec{x},y))\\ f(\vec{x}) &= \tau(\vec{x},p(|\vec{x}|))\end{aligned}$$

  and

$$(\forall n \le p(|\vec{x}|))\,[|\tau(\vec{x},n)| \le q(|\vec{x}|)]$$

  where p and q are polynomials with non-negative integer coefficients.

- *The collection* **B** *of numeric functions contains the following functions:*

0	*The constant zero function*
$2\cdot x$	*The left-shift function*
$\lfloor \frac{x}{2} \rfloor$	*The right-shift function*
$leq(x,y)$	$= \begin{cases} 1 & \text{if } x \le y \\ 0 & \text{otherwise} \end{cases}$
$Choice(x,y,z)$	$= \begin{cases} y & \text{if } x > 0 \\ z & \text{otherwise} \end{cases}$

- *For a collection of functions $\mathfrak{F}$, $Cl(\mathfrak{F})$ is closure of $\mathfrak{F}$ under composition and limited iteration.*
- $\mathfrak{P} \stackrel{df}{=} Cl(\mathbf{B})$.
- $\square_1^p$ *is the collection of functions which are computable by a polynomial-time TM.*

Theorem 1. $\mathfrak{P} = \square_1^p$

Proof. This is the Theorem 2 in [1]. We will only give a brief sketch of the proof.

Case 1: $\mathfrak{P} \subseteq \square_1^p$

By induction on complexity of definition of $f \in \mathfrak{P}$. All functions in $\mathbf{B}$ are p–time computable. If an oracle function is p–time computable by a TM, adding the oracle to the TM will not change its ability for p-time computation. Composition is just an oracle consultation. It is not difficult to see that *limited iteration* captures the idea of polynomial-time computation of a TM with oracles for previously-defined functions. Hence we have this side of inclusion.

Case 2: $\square_1^p \subseteq \mathfrak{P}$

By encoding the instant description (ID) of a TM. This is very similar to the class proof of SAT being a NP-complete problem [2]. The coding scheme used in this section of [1] is different from the one about S_2^1 in the later chapter of the same book. Nonetheless, all functions necessary for encoding/decoding the ID's of a TM are definable from $\mathbf{B}$ using composition and limited iteration.

□

3 Bounded Arithmetic

Definition 2.

- *The language of bounded arithmetic, $\mathcal{L}_{BA}$, is given as*

$$\mathcal{L}_{BA} = \{0, x', +, \cdot, |x|, \lfloor \tfrac{x}{2} \rfloor, \#, =, \leq\}$$

Symbols	*Meanings*
0	*the zero constant (function)*
x'	*successor function*
$x + y$	*addition*
$x \cdot y$	*multiplication*
$\|x\|$	*length of x in binary representation, i.e.* $\lceil \log_2(x+1) \rceil$
$\lfloor \frac{x}{2} \rfloor$	*largest integer smaller than or equal to $x/2$*
$x\#y$	$x\#y = 2^{\|x\| \cdot \|y\|}$

- $\mathfrak{B} \stackrel{df}{=} \{0, x', +, \cdot|x|, \lfloor \frac{x}{2} \rfloor, \#\}$, *i.e. the collection of all functions in $\mathfrak{L}_{BA}$.*

- *$BASIC$ is the set of open formulas that defines the functions in $\mathfrak{B}$. Details can be found in chapter 2 of* [1].
- *Φ–$PIND$ axioms are of the form $(\varphi(0) \wedge \forall x[\varphi(\lfloor \frac{x}{2} \rfloor) \rightarrow \varphi(x)]) \rightarrow \forall x \varphi(x)$ where $\varphi \in \Phi$.*

Φ–$PIND$ can be incorporated into a Tait-style calculus as an inference rule as follows:

$$(\Phi\text{–}PIND) \quad \frac{\Gamma, \varphi(0) \qquad \Gamma, \neg\varphi(\lfloor \frac{x}{2} \rfloor), \varphi(x)}{\Gamma, \varphi(t)}$$

where x does not occur free in Γ and t is any term in the language.

Definition 3.

- *Sharply bounded quantifiers are of the form $\exists x \leq |t|$ or $\forall x \leq |t|$ where t is a term in the language.*
- *$QF(\mathfrak{L})$ is the set of all open formulas in the language $\mathfrak{L}$. They are quantifier-free.*
- *$\Sigma^b_0(\mathfrak{L})$ is the closure of $QF(\mathfrak{L})$ under connectives and sharply bounded quantifications.*
- $\Delta^b_0(\mathfrak{L}) \stackrel{df}{=} \Sigma^b_0(\mathfrak{L}) \stackrel{df}{=} \Pi^b_0(\mathfrak{L})$
- *$\Sigma^b_{i+1}(\mathfrak{L})$ is the closure of $\Pi^b_i(\mathfrak{L})$ under $\wedge$, $\vee$, sharply bounded quantifications and bounded existential quantification, i.e. $\exists x \leq t$.*
- *$\Pi^b_{i+1}(\mathfrak{L})$ is the closure of $\Sigma^b_i(\mathfrak{L})$ under $\wedge$, $\vee$, sharply bounded quantifications and bounded universal quantification, i.e. $\forall x \leq t$.*
- *With respect to a theory $\mathbf{T}$ in $\mathfrak{L}$, a formula $\delta \in \Delta^b_i(\mathfrak{L})$ iff $\mathbf{T} \vdash (\delta \leftrightarrow \varphi) \wedge (\delta \leftrightarrow \psi)$ where $\varphi \in \Sigma^b_i(\mathfrak{L})$ and $\psi \in \Pi^b_i(\mathfrak{L})$.*
- *The theory $S^1_2 \stackrel{df}{=} BASIC + \Sigma^b_1$–$PIND$ in* [1]. *Using notation in* [3], *$S^1_2 \stackrel{df}{=} \Sigma^b_1(\mathfrak{B})$–$PIND$.*

4 Definable Functions and Conservative Extension

Definition 4. *For a theory $\mathbf{T}$ and a class of formula Φ,*

- *A function $f(\vec{x})$ is Φ–definable in $\mathbf{T}$ iff*

$$\begin{aligned} &\mathbf{T} \vdash \forall \vec{x} \exists y \varphi(\vec{x}, y) \\ &\mathbf{T} \vdash \forall \vec{x} y z \, [(\varphi(\vec{x}, y) \wedge \varphi(\vec{x}, z)) \rightarrow (y = z)] \\ &\mathbb{N} \models \forall \vec{x} \varphi(\vec{x}, f(\vec{x})) \end{aligned}$$

  where $\varphi \in \Phi$.

- *The collection of all Φ–definable functions in $\mathbf{T}$ is denoted by Φ–$DF(\mathbf{T})$.*

Definition 5. *Let $\mathbf{T}_1$ be a theory in a language $\mathfrak{L}_1$ and $\mathbf{T}_2$ in $\mathfrak{L}_2$ where $\mathfrak{L}_1 \subseteq \mathfrak{L}_2$. If $\mathbf{T}_2 \vdash \varphi$ implies $\mathbf{T}_1 \vdash \varphi$ for every formula φ in $\mathfrak{L}_1$, we say $\mathbf{T}_2$ is a conservative extension of $\mathbf{T}_1$.*

Theorem 2. *For a theory $\mathbf{T}$ with Σ^b_{i+1}–$PIND$, extending $\mathbf{T}$ with symbols for Σ^b_{i+1}–definable functions and Δ^b_i–definable predicates yields a conservative extension.*

Proof. Let the original language be $\mathfrak{L}$ and $\varphi_f \in \Sigma^b_{i+1}(\mathfrak{L})$ be the defining formula of a function f in the extended language . We have $\mathbf{T} \vdash_d \exists! y \varphi_f(y)$. For a derivation of Γ in the extended theory where Γ is in $\mathfrak{L}$, we can replace $\psi(f)$ with $\psi(y) \wedge \varphi_f(y)$ and combine a suitable subderivation of d to get a valid derivation of Γ in $\mathbf{T}$. Note that if $\psi(f)$ is used in Σ^b_{i+1}–$PIND$ in the extended system, $\psi(y) \wedge \varphi_f(y) \in \Sigma^b_{i+1}(\mathfrak{L})$ and Σ^b_{i+1}–$PIND$ is applicable to the formula. Similar arguments apply for predicates, by replacing the predicates with the defining formulas. To ensure sure Σ^b_{i+1}–$PIND$ is applicable to the replaced formula, predicate should be Δ^b_i–definable, since the predicates may be within the scope of negation. □

By theorem (2), we can use symbols for functions and predicates which are Σ^b_1–definable functions and Δ^b_0–definable respectively in S^1_2 to simplify our discussion. In [1], it is called "bootstrapping". The objective is to define functions which are similar to Gödel's coding/decoding functions for a sequence of numbers in S^1_2. We won't go into details about all these definitions. Instead, we illustrate the meanings of these functions by example.

To encode a number, we turn it into binary representation, insert a "1" before every binary digit, e.g. the number 10 is 1010_b in binary and is coded as 11101110_b, i.e. 238. Similarly, the number 3 is coded as 1111_b, i.e. 15. The number 0 is coded as 10_b, i.e 2. To code the sequence of numbers, we first code the individual numbers and add "00' as the comma. Thus the code representing the sequence $< 0, 10, 3 >$ is 100011101110001111_b, i.e. 146319. The functions used for decoding are also definable in S^1_2. Using the sequence $< 0, 10, 3 >$ as an example, we have

$$\begin{aligned} len(146319) &= 3 \\ \beta(146319, 1) &= 0 \\ \beta(146319, 2) &= 10 \\ \beta(146319, 3) &= 3 \end{aligned}$$

Not every number is a valid sequence but every (finite) sequence can be coded into a unique number. Suppose the largest number in a sequence is a and there are total b numbers in it. The code w for the sequence is bound by

$$SqBd(a, b) = (2 \cdot b + 1) \# (4 \cdot (2 \cdot a + 1)^2)$$

Theorem 3. $\mathfrak{P} \subseteq \Sigma^b_1$–$DF(S^1_2)$

Proof. By induction on complexity of definition of $f \in \mathfrak{P}$. It is clear that the functions in $\mathbf{B}$ is Σ^b_1–definable in S^1_2. Induction step for the composition case relies on the existence of the code w which encodes the sequence of all component functions in the arguments. It is left to the reader to work out the details.

For limited iteration, it is enough to find a number w that encodes the sequence of results of each iteration. The $SqBd$ function gives us the bound of w so we can express the defining formula with a Σ^b_1–formula. Suppose $f \in \mathfrak{P}$ is defined from g and h by limited iteration, the following formula is equivalent to $f(\vec{x}) = y$ and is provable in S^1_2

$$\exists! y\big(\exists w \leq SqBd(2^{q(|\vec{x}|)}, 2^{p(|\vec{x}|)})\big) \left[\begin{array}{ll} & \beta(1,w) = g(\vec{x}) \\ \wedge & (\forall i < len(w) - 1)\big[\beta(i'',w) = h(\vec{x}, i, \beta(i',w))\big] \\ \wedge & y = \beta(len(w), w) \end{array}\right]$$

and we can use subformula of the above as the defining formula for $f(\vec{x})$. Strictly speaking, the defining formula in this form is not a Σ^b_1–formula. Looking up the definition of len, we can replace $(\forall i < len(w) - 1)$ with its equivalence in term of $|w|$, and get a equivalent sharply bounded quantification. Hence there is a defining Σ^b_1–formula for $f(\vec{x})$. □

Note: The proofs about definable functions are really sketchy. They are presented in such a way in order to illustrate the idea without going too deep into details. Readers are strongly encouraged to refer to [1].

5 Herbrand Analysis for S^1_2

We will show the other way of inclusion of theorem (3) by technique used in [3], as an simpler alternative to chapter 5 of [1].

Theorem 4. *(Herbrand Theorem) For a theory* $\mathbf{T}$ *which is specified by open formulas as axioms, if* $\mathbf{T} \vdash \exists y \varphi(y)$*, we have a finite number of term* $t_1, \ldots, t_k$ *s.t.*

$$\mathbf{T} \vdash \varphi(t_1), \ldots, \varphi(t_k)$$

Proof. By induction on the length of a derivation of $\Gamma, \exists y \varphi(y)$ where Γ contains only open formula. □

Definition 6.

- $\mathfrak{L}_{\mathfrak{P}}$ *is the language obtained from* $\mathfrak{L}_{BA}$ *by adding symbol for each function in* $\mathfrak{P}$*.*
- $\Phi(\mathfrak{P})$*–*$PIND$ *is the theory obtained from theory which defines all functions in* $\mathfrak{P}$ *with (*Φ*–*$PIND$*) inference rule.*[1]

[1]It should be noticed that all axioms of $\Phi(\mathfrak{P})$–$PIND$ are open formulas.

Theorem 5. $\mathfrak{P}$ *is closed under definition by (finite) cases where the conditions are open formulas.*

Proof. Firstly, we show that the following functions for each atomic formula and connectives are in $\mathfrak{P}$

$f_{=}(x,y)$	$choice(leq(x,y), leq(y,x), 0)$
$f_{\leq}(x,y)$	$leq(x,y)$
$f_{\neg}(x)$	$choice(x,0,1)$
$f_{\wedge}(x,y)$	$choice(x, leq(1,y), 0)$
$f_{\vee}(x,y)$	$choice(x,1,leq(1,y))$

Then,we have the characteristic function $f_\varphi \in \mathfrak{P}$ for every open formula φ in $\mathfrak{L}_{\mathfrak{P}}$ by induction on the complexity of φ.

If a function f is defined by case s.t.

$$f(\vec{x}) = \begin{cases} t_1 & \text{if } \varphi_1(\vec{x}) \\ t_2 & \text{else if } \varphi_2(\vec{x}) \\ \vdots & \vdots \end{cases},$$

it is easy to give the same definition by choice function:

$$choice(f_{\varphi_1}(\vec{x}), t_1, choice(f_{\varphi_2}(\vec{x}), t_2, choice(\ldots)))$$

□

Corollary 6. $QF\text{–}DF(QF(\mathfrak{P})\text{–}PIND) \subseteq \mathfrak{P}$

Proof. Immediate consequence of theorems (4) and (5). □

Theorem 7. *(Term Extraction) If* $QF(\mathfrak{P})\text{–}PIND \vdash_d \Gamma, \exists y \varphi(\vec{x}, y)$ *where* Γ *contain no universal quantification*[2]*,* φ *is open and* $\vec{x}$ *is the list of all parameters of the derivation* d*, then we have* $QF(\mathfrak{P})\text{–}PIND \vdash \Gamma, \varphi(\vec{x}, f(\vec{x}))$ *where* $f \in \mathfrak{P}$*.*

Proof. By modifying the proof for theorem (4) for purely existential formulas and using Corollary (6). It should be noted that the argument does not work if Γ contains universal quantified formulas since $\forall$–inversions introduce extra variables not in $\vec{x}$. □

Theorem 8. *For an open formula* φ *and a term* t *in* $\mathfrak{L}_{\mathfrak{P}}$*, there is a function* f_φ *in* $\mathfrak{P}$ *s.t.*

$$\Sigma^b_1(\mathfrak{P})\text{–}PIND \vdash (\exists y \leq |t|)\varphi(\vec{x}, y) \leftrightarrow \varphi(\vec{x}, f_\varphi(\vec{x}))$$

Proof. By construction of a thorough search function, begin with 0 up to $|t|$. If no such witness is found, the search function returns $|t|+1$. Such function is Σ^b_1–definable in $\Sigma^b_1(\mathfrak{P})\text{–}PIND$, using the sequence coding technique. □

[2]In [3], it is called purely existential.

Corollary 9. *For any formula $\varphi \in \Sigma_0^b(\mathfrak{L}_\mathfrak{P})$ there is a formula $\varphi^* \in QF(\mathfrak{L}_\mathfrak{P})$ s.t.*

$$\Sigma_1^b(\mathfrak{P})\text{–}PIND \vdash \varphi \leftrightarrow \varphi^*.$$

Theorem 10. *(Σ_1^b–replacement)*
For a Σ_1^b–formula φ and a term s_1 in $\mathfrak{L}_{BA}$, there is a Σ_1^b–formula ψ and a term s_2 s.t. S_2^1 proves

$$(\forall x \le |t|)(\exists y \le s_1)\varphi(x,y) \leftrightarrow (\exists y \le s_2)(\forall x \le |t|)\psi(x,y)$$

Proof. This is Theorem 14 in [1]. The basic idea is to code the sequence of y's for each x into w.

$$S_2^i \vdash (\forall x \le |t|)(\exists y \le s_1)\varphi(x,y) \leftrightarrow \\ (\exists w \le SqBd(s_1,t))(\forall x \le |t|)[\varphi(x,\beta(x',w)) \wedge \beta(x',w) \le s_1]$$

□

The above theorem enables us to "push" the sharply bounded quantification into the scope of bounded quantifier.

Definition 7. *The collection of strict Σ_1^b-formulas in a language $\mathfrak{L}$, denoted by $s\text{-}\Sigma_1^b(\mathfrak{L})$, is the smallest set of formulas which begin with exactly one bounded existential quantifier, i.e. $\exists x \le t$, followed by an open formula.*

Corollary 11. *For any formula $\varphi \in \Sigma_1^b(\mathfrak{L}_\mathfrak{P})$, there is a formula $\varphi^* \in s\text{-}\Sigma_1^b(\mathfrak{L}_\mathfrak{P})$ s.t.*

$$\Sigma_1^b(\mathfrak{P})\text{–}PIND \vdash \varphi \leftrightarrow \varphi^*.$$

Proof. Pushing any sharply bounded quantifier inside the scope of a bounded quantifier by Σ_1^b–replacement (theorem (10)), combing any two bounded existential quantifiers into one by pairing function (definable in S_2^1) and replacing the sharply-bounded formula with its equivalence by corollary (9) gives the desired $s\text{-}\Sigma_1^b$ formula. □

Theorem 12. $\Sigma_1^b\text{–}DF(\Sigma_1^b(\mathfrak{P})\text{–}PIND) = QF\text{–}DF(QF(\mathfrak{P})\text{–}PIND)$

Proof. The $\forall$–inversion holds as none of the axioms is specified using $\forall$. Together with corollary (11), it is suffice to show that $s\text{-}\Sigma_1^b(\mathfrak{P})\text{–}PIND$ is conservative over $QF(\mathfrak{P})\text{–}PIND$ for $s\text{-}\Sigma_1^b$–formula. We examine a normal derivation of Γ which contains no universal quantification. Obvious lines in the following derivations are skipped and we may optionally illustrate the skipped part/inference by $\vdots$ and double lines.

Consider the first application of ($s\text{-}\Sigma_1^b\text{–}PIND$) with p.f. $\exists y[y \le t(s) \wedge \varphi(s,y)]$ where φ is an open formula. Let $\psi(x,y) \stackrel{df}{=} y \le t(x) \wedge \varphi(x,y)$. The instance is as follows:

$$\frac{\Delta, \exists y[\psi(0,y)] \qquad \Delta, \neg\exists y[\psi(\lfloor \tfrac{b}{2} \rfloor, y)], \exists y[\psi(b,y)]}{\Delta, \exists y[\psi(s,y)]}$$

Because Γ is purely existential and the derivation is normal, Δ must also be purely existential. Inverting the $\forall$–quantifier in $\neg\exists y[\psi(\lfloor\frac{b}{2}\rfloor, y)]$, we obtain an eigenvariable c which does not occur in Δ. By Theorem (7), there are $f_0, f_1 \in \mathfrak{P}$ s.t.

$$QF(\mathfrak{P})\text{–}PIND \vdash \Delta, \psi(0, f_0(0))$$
$$QF(\mathfrak{P})\text{–}PIND \vdash \Delta, \neg\psi(\lfloor\tfrac{b}{2}\rfloor, c)), \psi(b, f_1(b, c))$$

Now we define

$$f(0) = f_0(0)$$
$$f(x') = f_1(x', f(\lfloor\tfrac{x'}{2}\rfloor))$$

Function defined in this manner belongs to $\mathfrak{P}$ and will be shown later in the proof. We choose this form to facilitate our argument for conservativeness.

A formal way to perform substitution in derivation is by using equality axioms and cuts. Substituting $f(\lfloor\frac{b}{2}\rfloor)$ for c, f for f_1 when $b \neq 0$ (i.e. $b = x'$), we have the following derivation in $QF(\mathfrak{P})\text{–}PIND$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\neg\psi(0, f(0)), \psi(0, f(0))}}{b \neq 0, \neg\psi(\lfloor\frac{b}{2}\rfloor, f(\lfloor\frac{b}{2}\rfloor)), \psi(b, f(b))} \quad \dfrac{\dfrac{\vdots}{\Delta, \neg\psi(\lfloor\frac{b}{2}\rfloor, c)), \psi(b, f_1(b, c)}}{b = 0, \neg\psi(\lfloor\frac{b}{2}\rfloor, f(\lfloor\frac{b}{2}\rfloor)), \psi(b, f(b))}}{\neg\psi(\lfloor\frac{b}{2}\rfloor, f(\lfloor\frac{b}{2}\rfloor)), \psi(b, f(b))}\,(C)$$

With this, we can derive the same conclusion of $s\text{-}\Sigma_1^b\text{–}PIND$ as follows:

$$\dfrac{\dfrac{\dfrac{\vdots}{\Delta, \psi(0, f(0))} \quad \dfrac{\vdots}{\Delta, \neg\psi(\lfloor\frac{b}{2}\rfloor, f(\lfloor\frac{b}{2}\rfloor)), \psi(b, f(b))}}{\Delta, \psi(s, f(s))}\,(QF\text{–}PIND)}{\Delta, \exists y[y \leq t(s) \wedge \varphi(s, y)]}\,(\exists)$$

We need to show $f \in \mathfrak{P}$ to complete the proof. Let's consider the function $frontbits(x, y)$ which gives the number with binary representation identical to the leading y bits of x. We skip the limited iteration definition but it is not difficult to see $frontbits(x, y) \in \mathfrak{P}$. Then $f_2(b, c, d) \stackrel{df}{=} f_1(frontbits(b, c), d)$ belongs to $\mathfrak{P}$ as $\mathfrak{P}$ is closed under composition. The function f can be defined by limited iteration from f_0 and f_2 as follows:

$$\tau(b, 0) = f_0(0)$$
$$\tau(b, c') = f_2(b, c, \tau(b, c))$$
$$f(b) = \tau(b, |b|)$$

where $p(|b|) = |b|$ and $q(|b|) = t(b)$. Thus, $f \in \mathfrak{P}$.

Hence, we can reduce the number of $s\text{-}\Sigma_1^b\text{–}PIND$ application by one. By induction, the conservative property is shown. □

6 Generalization: S_2^i and Σ_i^P

Definition 8.

- *A function f is a predicate if its range is $\{0,1\}$.*
- *For a collection of functions $\mathfrak{F}$, $PRED(\mathfrak{F})$ is the collection of predicates in $\mathfrak{F}$.*
- *For two functions f and g, we define*

$$(\exists y \le f(\vec{x}))g(\vec{x},y) \stackrel{df}{=} \begin{cases} 1 & \text{if } (\exists y \le f(\vec{x}))\,[g(\vec{x},y) > 0] \\ 0 & \text{otherwise} \end{cases}$$

$$(\forall y \le f(\vec{x}))g(\vec{x},y) \stackrel{df}{=} \begin{cases} 1 & \text{if } (\forall y \le f(\vec{x}))\,[g(\vec{x},y) > 0] \\ 0 & \text{otherwise} \end{cases}$$

- *For a collection of functions $\mathfrak{F}$ which is closed under composition:*

$$PB\exists(\mathfrak{F}) = \{(\exists y \le 2^{p(|\vec{x}|)})R(\vec{x},y) \mid R \in PRED(\mathfrak{F})\}$$
$$PB\forall(\mathfrak{F}) = \{(\forall y \le 2^{p(|\vec{x}|)})R(\vec{x},y) \mid R \in PRED(\mathfrak{F})\}$$

where p is a suitable polynomial.

Definition 9. *With these, we can define the hierarchy of predicates which corresponds to polynomial hierarchy:*

$$\Delta_1^p \stackrel{df}{=} PRED(\ \square_1^p).$$
$$\Sigma_i^p \stackrel{df}{=} PB\exists(\Delta_i^p).$$
$$\Pi_i^p \stackrel{df}{=} PB\forall(\Delta_i^p).$$
$$\square_{i+1}^p \stackrel{df}{=} Cl(\Sigma_i^p).$$
$$PH \stackrel{df}{=} \bigcup_{k\in\mathbb{N}} \Sigma_k^p.$$

Δ_1^p, Σ_1^p and Π_1^p are essentially the computational complexity classes P, NP and co–NP respectively.

Theorem 13. $\square_i^p \subseteq \Sigma_i^b$–$DF(S_2^i)$

Proof. The base case is Theorem (3). Here we will show the case for $i+1$. Modify the proof for theorem (3), we only need to include the case when a function $f \in \square_{i+1}^p$ is defined from $g \in \Sigma_i^b$–$DF(S_2^i)$ by $PB\exists$ s.t.

$$f(\vec{x}) = \begin{cases} 1 & \text{if } (\exists y \le 2^{p(|\vec{x}|)})[g(\vec{x},y) > 0] \\ 0 & \text{otherwise} \end{cases}$$

and the defining formula for g is $\varphi_g(\vec{x}, y, z) \in \Sigma_i^b$ s.t. $(g(\vec{x}, y) = z) \leftrightarrow \varphi_g(\vec{x}, y, z)$. Consider the following formula $\varphi_f \stackrel{df}{=}$

$$\Big((u = 1) \wedge (\exists y \leq 2^{p(|\vec{x}|)})\,[(z > 0) \wedge \varphi_g(\vec{x}, y, z)]\Big)$$
$$\vee \Big((u = 0) \wedge (\forall y \leq 2^{p(|\vec{x}|)})\,[(z \neq 0) \vee \neg\varphi_g(\vec{x}, y, z)]\Big)$$

It is obvious that $(f(\vec{x}) = u) \leftrightarrow \varphi_f(\vec{x}, u)$ and $\varphi_f \in \Sigma_{i+1}^b$. □

Theorem 14. Σ_i^b–$DF(S_2^i) \subseteq \ \Box_i^p$

Proof. By extending the language $\mathfrak{L}_{BA}$ with symbols for functions in $\Box_i^p$. It can be shown that for every formula in $\Sigma_i^b(\mathfrak{L}_{BA})$, there is an equivalence in s-Σ_1^b–formula in the extended language, provable in the extended theory as a form of Skolemization or operator theory in [3]. The form of defining formulas for functions in $\Box_i^p$ is carefully stated to preserve $\wedge$–, $\vee$–, and $\forall$–inversions as well as Theorem (7), i.e. we will avoid any use of quantification or logical connectives. For example, to define a Σ_1^b–Skolem function f for an open formula φ s.t.

$$(\exists y \leq t(\vec{x}))\varphi(\vec{x}, y) \leftrightarrow \varphi(\vec{x}, f(\vec{x})),$$

the axioms are given as two sets of formula

$$\{\neg y \leq t(\vec{x}), \neg\varphi(\vec{x}, y), \varphi(\vec{x}, f(\vec{x}))\}, \{f(\vec{x}) \leq t(\vec{x})\}.$$

Σ_{i+1}^b-Skolem functions are defined with open formulas which consist of $\Box_i^p$ function symbols, instead of using Σ_i^b–formulas directly. With these measures, the argument follows the proof of Theorem (12). □

References

[1] Samuel R. Buss. *Bounded Arithmetic*. Bibliopolis, 1986.

[2] David S. Johnson Miacheal R. Gary. *Computers and Intractability, A guid to the theoryof NP-completeness*. W. H. Freeman, 1979.

[3] Wilfried Sieg. Herbrand analyses. *Archive for Mathatical Logic*, 30:409–441, 1991.