Proof. Let $\phi(x)$ be a Σ_i^b-formula in prenex normal form. We want to show that ϕ is in S_2^i equivalent to a formula ψ in prenex form, in which all sharply bounded quantifiers follow after all bounded but not sharply bounded ones. Let

$$\forall i \leq |t| \exists y \leq s, \theta(i, y)$$

be a subformula of ϕ: That is, θ is a Σ_i^b-formula too. By the sharply bounded collection $BB\Sigma_i^b$ available through Lemma 5.2.12 in S_2^i this subformula is equivalent to

$$\exists w \leq t\#s \forall i \leq |t|, \theta(i, (w)_i)$$

This demonstrates how to switch a pair of sharply bounded/bounded quantifiers. Repeating this yields ψ.

To get from ψ a strict Σ_i^b-formula we only have to replace two consecutive occurrences of the same bounded, but not sharply bounded quantifier by one; but this is easily achieved by a pairing function that can be defined by using the function $(w)_i$. Q.E.D.

Lemma 5.2.15. *The theory S_2 is a conservative extension of the theory $I\Delta_0 + \Omega_1$. That is: Any formula in language L_{PA} provable in S_2 is also provable in $I\Delta_0 + \Omega_1$.*

In fact, every model M of $I\Delta_0 + \Omega_1$ can be expanded to a model of S_2.

Proof. The theory S_2 defines the function $\omega_1(x)$ and proves the axiom Ω_1: This follows from a trivial bound

$$\omega_1(x) \leq ((x\#x) + 2)^2$$

This shows that $I\Delta_0 + \Omega_1 \subseteq S_2$.

Let M be a model of $I\Delta_0 + \Omega_1$. By Theorem 3.2.6 there is a Δ_0-definition of the graph of exponentiation and by the remark before Lemma 5.1.5 there is such definition for which $I\Delta_0$ can prove the recursive equations. It follows that $I\Delta_0$ can also Δ_0-define the graph of the function $a\#b$, and from the bound

$$a\#b \leq \omega_1(a \cdot b + 2)$$

it follows that $I\Delta_0 + \Omega_1$ proves the totality of $a\#b$. The axioms of BASIC pose no problem; nor do the Δ_0-definitions of $|x|$ and $\lfloor (x/2) \rfloor$. This demonstrates that M can be extended by functions to obey BASIC, and from the fact that they are Δ_0- definable in M it follows that induction will hold for all bounded formulas in the expanded language. Q.E.D.

5.3. Theory PV

Building on an earlier work of Bennett (1962), Cobham (1965) characterized the class of polynomial time functions in the following "machine independent" way.

We say that a function f is defined from functions g, h_0, h_1, and ℓ by *limited recursion on notation* if:

1. $f(\overline{x}, 0) = g(\overline{x})$,
2. $f(\overline{x}, s_i(y)) = h_i(\overline{x}, y, f(\overline{x}, y))$, for $i = 0, 1$,
3. $f(\overline{x}, y) \leq \ell(\overline{x}, y)$,

where $s_0(y)$ and $s_1(y)$ are two functions adding 0, respectively 1, to the right of the binary representation of y

$$s_0(y) := 2y$$

$$s_1(y) := 2y + 1$$

Theorem 5.3.1 (Cobham 1965). *The class of polynomial time functions is the smallest class of functions containing constant* 0, *functions* $s_0(y)$, $s_1(y)$ *and* $x\#y$, *and closed under:*

1. *permutation and renaming of variables*
2. *composition of functions*
3. *limited recursion on notation*

We might note at this point that it is possible to enlarge basic functions by finitely many polynomial time functions such that requirement 3 becomes redundant in the theorem: That is, the class of polynomial time functions has a *finite basis* (cf. Muchnik 1970).

Building on this theorem Cook (1975) defined formal system PV (for *p*olynomially *v*erifiable). There are two motivations for considering a system like that: One is its relation to the *extended Frege* system (Corollary 9.2.4); another is more philosophical, to define a system in which instances of general proofs can be verified by constructive, computationally feasible procedures.

Definition 5.3.2. *We simultaneously define function symbols of rank k and PV-derivations of rank k,* $k = 0, 1, \ldots$. *The language of PV will then consist of all function symbols of any rank, and a PV-derivation will be a PV-derivation of any rank.*

(a) Function symbols of rank 0 *are constant* 0*; unary* $s_0(y)$, $s_1(y)$, *and* $Tr(x)$*; and binary* $x \frown y$, $x\#y$, *and* $Less(x, y)$.

(b) Defining equations of rank 0 *are:*

$$Tr(0) = 0$$

$$Tr(s_i(x)) = x, \qquad i = 0, 1$$

$$x \frown 0 = x$$

$$x \frown (s_i(y)) = s_i(x \frown y), \qquad i = 0, 1$$

$$x\#0 = 0$$