PV_1 consists of all equations $t = u$ provable in PV but has also a form of induction axiom: For an open formula $\psi(x)$ define a function $h(b, u)$ by

(a) $h(b, 0) = (0, b)$

(b) if $h(b, \lfloor (u/2) \rfloor) = (x, y)$ and $u > 0$ then set

$$h(b,u) := \begin{cases} (\lceil (x + y/2) \rceil, y) & \text{if } \lceil (x + y/2) \rceil < y \wedge \psi(\lceil (x + y/2) \rceil) \\ (x, \lceil (x + y/2) \rceil) & \text{if } x < \lceil (x + y/2) \rceil \wedge \neg\psi(\lceil (x + y/2) \rceil) \\ (x, y) & \text{otherwise} \end{cases}$$

Then PV_1 contains the axiom

$$(\psi(0) \wedge \neg\psi(b) \wedge h(b, b) = (x, y)) \rightarrow (x + 1 = y \wedge \psi(x) \wedge \neg\psi(y))$$

This axiom simulates the binary search and is thus related to the PIND scheme. The logic of PV_1 is the usual first order predicate calculus.

Theory PV_{i+1} contains PV_1 and has inductively defined characteristic functions of all Σ_i^b- predicates in its language, in particular universal axioms of the form

$$\exists y \leq t(\bar{x}),\ f(\bar{x}, y) = 1 \rightarrow g(\bar{x}) \leq t(\bar{x}) \wedge f(\bar{x}, g(\bar{x})) = 1$$

where g are new function symbols introduced inductively for all formulas of the form

$$\exists y \leq t(\bar{x}),\ f(\bar{x}, y) = 1$$

Furthermore, PV_{i+1} is closed under definition by cases, composition, and limited recursion on notation (i.e., has function symbols for all functions introduced by these processes) and also contains the preceding axiom for every open ψ in its language. The logic of PV_{i+1} is also the first order predicate calculus.

Note that each PV_i is a universal theory. We state a theorem that will be proved in Section 6.1 (cf. Corollary 6.1.3).

Theorem 5.3.5. *For every $i \geq 1$, PV_{i+1} is fully conservative over the theory T_2^i. The theory PV_1 is conservative over PV.*

5.4. Coding of sequences

In this section we sketch a way to code finite sets and sequences in S_2^1. This is necessary in order to be able to formalize various syntactic and logical notions and computations of machines in subsystems of S_2.

For the language L_{PA} (and $I\Delta_0 + \Omega_1$) this is quite nontrivial as one must first find a well-behaved Δ_0-definition of the graph of exponentiation, in order to speak about lengths of numbers and their bits. The existence of a Δ_0-definition of the graph of exponentiation follows from Bennett (1962) (cf. Theorem 3.2.6), but the theorem does not imply that there is such a definition about which $I\Delta_0$ or $I\Delta_0 + \Omega_1$

could prove the basic recursive properties

1. $x^0 = 1$
2. $x^{y+1} = x^y \cdot x$

Such a bounded definition was constructed by J. Paris (in Dimitracopulos 1980) (see also Pudlák 1983).

Another crucial function whose well-behaved bounded definition is needed is Numones(x): the number of ones in the binary expansion of x. The function is clearly computable in logarithmic space, and thus its graph is Δ_0-definable by Corollary 3.2.9, but proving the basic recurrence properties

1. Numones$(0) = 0$
2. Numones$(2x) =$ Numones(x)
3. Numones$(2x + 1) =$ Numones$(x) + 1$

again requires some work. In $I\Delta_0 + \Omega_1$ this is easier. Theorem 3.2.7 is a general tool showing that all usual concepts defined by inductive properties can be defined in a well-behaved way in $I\Delta_0 + \Omega_1$.

With these two definitions in hand one defines the basic relations and functions on finite words, identifying a number with its dyadic representation and the coding of sequences then follows the development of rudimentary sets in Bennett (1962). In $I\Delta_0 + \Omega_1$ the formalization of syntax and logic is then smooth.

If one wants to formalize logical notions in $I\Delta_0$ only, there are other complications. For example, the term resulting from substitution of a term u into a term v for a variable x will not in general have length proportional to $|u| + |v|$; hence its code will not be bounded by a polynomial in u, v and by Theorem 5.1.4 $I\Delta_0$ cannot prove that the substitution is always defined.

In some situations one can restrict the syntax, for example, to terms and formulas with only one occurrence of each variable (or to their representation with this property), but the formalizations obtained in this way are unnatural.

We refer the reader to Paris and Wilkie (1987b) or Hájek and Pudlák (1993) for detailed development of coding, sequences, and syntax in $I\Delta_0$ and $I\Delta_0 + \Omega_1$.

With the language L of S_2 the situation is much simpler because we have the length function $|x|$ in language allowing us to define the graph of exponentiation immediately by

$$2^x = y \equiv \exists z < y, z + 1 = y \wedge |z| = x \wedge |y| = x + 1$$

which is equivalent to

$$2^x = y \equiv \forall z < y, z + 1 = y \rightarrow |z| = x \wedge |y| = x + 1$$

We want to define the basic notions of rudimentary sets and of coding of sequences by means of Δ_1^b-formulas in S_2^1, in such a way that S_2^1 can prove the basic properties and, in particular, the properties of Lemma 5.1.5. This is done in great detail in Buss (1986). Another approach is to follow the development of Paris and Wilkie (1987b) and Hájek and Pudlák (1993) in S_2^1 and to verify that all notions that are

only Δ_0 there are Δ_1^b in S_2^1. To illustrate these topics we outline a way of coding sequences, but we shall proceed rather swiftly, leaving details to the reader.

First we define *the pairing function*

$$\langle a, b\rangle := \left\lfloor \frac{(a+b)(a+b+1)}{2} \right\rfloor + a$$

It is defined by a term (hence is Δ_1^b) and S_2^1 can prove the basic property

$$\langle a, b\rangle = \langle u, v\rangle \equiv (a = u \wedge b = v)$$

Then we define the predicate "*a is a power of* 2"

$$\mathrm{Pow}(a) \equiv \exists x \leq a, x + 1 = a \wedge |x| + 1 = |a|$$

which is provably in S_2^1 equivalent to

$$\forall x \leq a, x + 1 = a \rightarrow |x| + 1 = |a|$$

Next define the function *the i*th *bit of a*

$$\mathrm{bit}(a, i) := \begin{cases} 1 & \text{if } \exists u, v, w \leq a, u + v + 2vw = a \wedge \mathrm{Pow}(v) \wedge |v| = i + 1 \\ & \qquad \wedge u < v \\ 0 & \text{otherwise} \end{cases}$$

which is also Δ_1^b as $\mathrm{bit}(a, i) = 1$ is also equivalent to

$$\forall u, v, w \leq a, u + v + 2vw = a \wedge \mathrm{Pow}(v) \wedge |u| \leq i \rightarrow |v| = i + 1$$

Using this function we define *the elementhood predicate*

$$i \in a \equiv \mathrm{bit}(a, i) = 1$$

Claim 1. *Functions and predicates* $\langle a, b\rangle$, *Pow*(a), *bit*(a, i), *and* $i \in a$ *are* Δ_1^b *in* S_2^1.

We want to code arbitrary 0–1 words. This cannot be done just by binary expansions of numbers that always start with 1. So we think of a *word* as pair $\langle u, v\rangle$, coding the word consisting of first right $|v|$ bits of u. With this interpretation in mind define the *equality of words* $a =_w b$ by

$$\exists x, y \leq a \exists u, v \leq b, \langle x, y\rangle = a \wedge \langle u, v\rangle = b \wedge$$

$$\wedge (\forall i \leq |y|, i \in x \equiv (i \in u \wedge i \leq |v|)) \wedge (\forall i \leq |v|, i \in u \equiv (i \in x \wedge i \leq |y|))$$

which is again Δ_1^b as x, y and u, v are unique. We also define the function the ith *letter in word a*

$$\mathrm{Letter}(a, i) := \begin{cases} 1 & \text{if } \exists u, v \leq a, \langle u, v\rangle = a \wedge i \in u \wedge i \leq |v| \\ 0 & \text{otherwise} \end{cases}$$

The idea of coding sequences of words is that a sequence will be coded by a pair $\langle a, b\rangle$, where the ith bit 1 in b marks the end of the ith subword of a: That is, a sequence $w_1, \ldots, w_t$ of t words will be coded by number a whose binary expansion is $w_t \frown \ldots \frown w_1$ and number b, which has bit 1 in positions $|w_1|$, $|w_1| + |w_2|, \ldots, |w_1| + \cdots + |w_t| = |a|$.

This idea requires that we must be able to define the function *counting the number of ones among the first i bits of a*

$$\text{Numones}(a, i) := |\{j \mid j \leq i \wedge j \in a\}|$$

Define

$$\begin{aligned}&\text{Numones}(a, i) = k \text{ iff}\\ &\exists x \leq (a\#a)\#a \forall u \leq |a|, \langle 1, u\rangle \in x \equiv (u \in a \wedge \forall v < u, v \notin a)\\ &\quad \wedge \forall t, u, v \leq |a|, \langle t, u\rangle \in x \wedge u < v \wedge v \in a \wedge (\forall s < v, u < s \rightarrow s \notin a)\\ &\quad \rightarrow \langle t + 1, v\rangle \in x \wedge \exists u \leq i, \langle k, u\rangle \in x \wedge \forall u \leq i, \langle k + 1, u\rangle \notin x\end{aligned}$$

In words: x codes an increasing map from $\{1, \ldots, k\}$ onto the 1's of a. Such an x is unique; hence the definition is Δ_1^b, and the inductive character of the definition of x allows us to prove basic inductive properties of Numones(a) (see previous discussion).

We are ready to define *sequences* and the function $(w)_i$

$$\text{Seq}(w) \equiv \exists x, y \leq w, \langle x, y\rangle = w \wedge |x| = |y|$$

and for w a sequence

$$(w)_i = u \equiv \exists x, y \leq w, \langle x, y\rangle = w \wedge \forall t \leq |u| \forall j, \qquad k \leq |x|,$$

$$\begin{aligned}\text{Numones}(y, j) = i - 1 \wedge \text{Numones}(y, k) = i \rightarrow\ & k = j + |u|\\ \wedge\ & t \in u \equiv (j + t) \in x\end{aligned}$$

Lemma 5.4.1. *The function $(w)_i = u$ is Δ_1^b-definable in S_2^1 and S_2^1 proves the conditions of Lemma 5.1.5.*

We shall conclude this section by a lemma stating that some predicates can be in a sense coded in S_2^i. It extends Lemma 5.2.12.

Lemma 5.4.2. *Let $A(a)$ be a $\Sigma_0^b(\Sigma_i^b)$ -formula, that is, a formula obtained from Σ_i^b-formulas by logical connectives and sharply bounded quantification. Let $i \geq 1$.*

The theory S_2^i proves

$$\forall x \exists y \forall t \leq |x|, A(t) \equiv (t \in y)$$

That is: Any bounded set of lengths defined by a $\Sigma_0^b(\Sigma_i^b)$-formula can be coded by a number.

Proof. It is enough to show that for $A \in \Sigma_i^b$, which also implies that any $\Sigma_0^b(\Sigma_i^b)$-predicate can be (on any interval $[0, |x|]$) expressed as Δ_1^b whose coding follows from the case $i = 1$.

Consider a Σ_i^b-formula $B(s)$ with parameter x

$$\exists y \leq x \forall t \leq |x|, (t \in y \rightarrow A(t)) \wedge |y| = s$$

Clearly $B(0)$ holds as y corresponds to the empty set, and so by the Σ_i^b-LENGTH-MAX principle available in S_2^i by Lemma 5.2.7 there is maximal $s \leq |x|$ satisfying B. It is straightforward to verify that y corresponding to this s codes A on interval $[0, |x|]$. Q.E.D.

Corollary 5.4.3. *For $i \geq 1$ the theory S_2^i proves the $\Sigma_0^b(\Sigma_i^b)$-PIND scheme.*

An interesting topic related to coding are partial truth definitions; see Paris and Dimitracopoulos (1982).

5.5. Second order systems

In this section we shall introduce some second order systems of bounded arithmetic, most from Buss (1986). We shall, however, proceed by model-theoretic reasoning rather than by direct proof-theoretic investigations. This will allow us to give simple model-theoretic proofs for the so-called RSUV isomorphism and translate several results from the previous section directly to these systems. It also allows us to relate the use of a second order object to the limited use of exponentiation.

Consider M a nonstandard model of S_2 and define a particular cut $I \subseteq_e M$ by

$$\text{for } a \in M: \ a \in I \quad \text{iff} \quad M \models \exists x, a = |x|$$

For the obvious reason we shall denote this cut by $\mathrm{Log}(M)$. This cut is closed under addition and multiplication (as $|a| \cdot |b| + 1 = |a\#b|$), but it is not necessarily closed under # (that would require that M is closed under $\omega_2(x)$, which we do not assume).

Take a collection $\mathcal{X}_M$ of those subsets of $\mathrm{Log}(M)$ *coded* in model M, that is, those $\alpha \subseteq \mathrm{Log}(M)$ such that for some $a \in M$

$$\forall i \in \mathrm{Log}(M), (M \models i \in \alpha) \equiv (\mathrm{bit}(a, i) = 1)$$

We shall denote such an α by $\tilde{a}$.

Consider now the two-sorted first order structure $(\mathrm{Log}(M), \mathcal{X}_M)$ with all symbols of $L \setminus \{x\#y\}$ defined for elements of $\mathrm{Log}(M)$ by restricting the operations and relations from M, with $=$ defined on $\mathcal{X}_M$ and with the relation $i \in \alpha$ defined for pairs from $\mathrm{Log}(M) \times \mathcal{X}_M$ by the preceding condition. We call elements of the second sort $\mathcal{X}_M$ *sets*.