A Note on the Exponential Relation in Peano Arithmetic II

Paul J. Voda
Institute of Informatics, Comenius University Bratislava,
Mlynská dolina, 842 15 Bratislava, Slovakia,
email: voda@fmph.uniba.sk
homepage: www.fmph.uniba.sk/∼voda

November 2001

0.1 Introduction. The development of Peano arithmetic or of μ-recursive functions proceeds smoothly once the graph of the exponentiation function $2^x = y$ is defined as a Σ_1-predicate because one can code finite sequences as digits of 2^k-ary representation of numbers. Gödel did the development with the help of his β-function using the Chinese remainder theorem (see [Sho67]). A slightly simpler development is possible via p-adic concatenation (p is a prime) as suggested by Kripke (see [BJ80]). Smullyan in [Smu92] uses an idea of Quine and derives $2^x = y$ as a Σ_1-predicate by a detour through the general relation $x^y = z$. He poses as a problem a direct definition of $2^x = y$.

That the relation $x^y = z$ is even a Σ_0-predicate was proved by Bennett [Ben62]. For a formal proof of this in the fragment $I\Sigma_0$ of Peano arithmetic see [HP93]. All approaches require an auxiliary pairing function and in $I\Sigma_0$ also difficult estimates of quantifier bounds.

We present a simple derivation of the relation $2^x = y$ as a Σ_0-predicate which extends to $p^x = y$ for any prime p in a straightforward way. The derivation is based on elementary facts about binary representation of numbers without any recourse to pairing, p-adic concatenation, or β functions. We think that we have largely succeeded in supressing the metamathematical character of the coding by expressing it in purely arithmetical terms. Our derivation easily formalizes in $I\Sigma_0$-arithmetic. This note is a simplified presentation of the idea discussed in our first note [Vo98].

0.2 Σ_0-predicates. Σ_0-predicates are defined by formulas constructed from 0, 1, +, $\cdot$, =, propositional connectives, and bounded quantifiers. The last are $\exists x{\le}\tau\,\phi$ and $\forall x{\le}\tau\,\phi$ where τ is a term. We permit also sharp bounds in quantifiers, for instance, $\exists x{<}\tau\,\phi$ as abbreviation for $\exists x{\le}\tau\,(x \neq \tau \wedge \phi)$. We abbreviate by x^2 the term $x{\cdot}x$ and by 2 the term $1+1$.

0.3 Some Σ_0-predicates. The predicate $x \mid y$ of *divisibility* holding if x divides y is a Σ_0-predicate by:

$$x \mid y \leftrightarrow \exists z{\le}y\, y = x{\cdot}z \ .$$

The relation $<$ is Σ_0 by:

$$x < y \leftrightarrow \exists z{\le}y\, y = x + z + 1$$

Myhill has observed that a positive number p is a power of two, in symbols $Pow_2(p)$, iff all its divisors $d > 1$ are themselves divisible by 2. This is captured by the following Σ_0-definition of the predicate:

$$Pow_2(p) \leftrightarrow 0 < p \wedge \forall d{\le}p\, (d \mid p \rightarrow d = 1 \vee 2 \mid d) \ .$$

For powers of two p and q such that $p > q$ the Σ_0-predicate $Sgm(x, p, q, s)$ holds iff the number s is a segment in the binary representation of x between p and q:

$$Sgm(x, p, q, s) \leftrightarrow \exists a{\le}x \exists b{<}q (x = a{\cdot}p + s{\cdot}q + b \wedge s{\cdot}q + b < p) \ .$$

The Σ_0-predicate $p \in x$ holds if p is a power of two set to one in the binary representation of x:

$$p \in x \leftrightarrow Pow_2(p) \wedge Sgm(x, 2{\cdot}p, p, 1) \ .$$

Note that we have $1 \in x$ iff x is odd.

bitno.	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$y{+}s$	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	1	1
$x{\cdot}y{+}t$	1	1	1	0	0	0	0	0	1	1	1	0	0	1	1	0	1	0

Figure 1: Course of values sequences $y + s$ and $x{\cdot}y + t$ encoding the computation of $2^x = y$ for $x = 14$ and $y = 16,384$.

0.4 Computing 2^x by recursion on notation. Consider the following definition of the exponentiation function 2^x by recursion on notation which is known to computer programmers as the *fast (logarithmic) program for* 2^x:

$$\begin{aligned} 2^0 &= 1 \\ 2^{2\cdot x+0} &= (2^x)^2 \qquad\qquad \text{if } x > 0 \\ 2^{2\cdot x+1} &= 2{\cdot}(2^x)^2 \ . \end{aligned}$$

If $2^x = y$ then there is a finite course of values sequence of powers of two $p_n, p_{n-1}, \ldots, p_0$ which are needed in the computation of 2^x. The sequence is such that $p_n = y = 2^x$, $p_0 = 1 = 2^0$, and $p_{i+1} = p_i^2$ or $p_{i+1} = 2{\cdot}p_i^2$ for all $n > i \geq 0$. Note that $p_{i+1} = (k{+}1){\cdot}p_i^2$ for a unique $k = 0$ or $k = 1$. There are numbers x_i such that $2^{x_i} = p_i$, $x_n = x$, $x_0 = 0$, and $x_{i+1} = 2{\cdot}x_i{+}k$ where $p_{i+1} = (k{+}1){\cdot}p_i^2$ for all $n > i \geq 0$.

For some $s < y$ we then have $y + s = \sum_{i\leq n} p_i = \sum_{i\leq n} 2^{x_i}$ and for some $t < y$ we have $x{\cdot}y + t = \sum_{i\leq n} x_i{\cdot}p_i = \sum_{i\leq n} x_i{\cdot}2^{x_i}$. This suggests a simple arithmetical (as opposed to metamathematical) encoding of the course of values sequences for the powers p_i and exponents x_i. Figure 1 shows the course of values sequences for the computation of 2^{14}.

0.5 Graph of exponentiation $2^x = y$ is a Σ_0-predicate. The following Σ_0-predicate $\mathit{Seqs}(s,t)$ holds iff $s = \sum_{i\leq n} 2^{x_i}$ and $t = \sum_{i\leq n} x_i{\cdot}2^{x_i}$, i.e. if s and t encode the course of values sequences of powers and exponents for the computation of 2^{x_n}:

$$\begin{aligned}&\mathit{Seqs}(s,t) \leftrightarrow \forall p{\leq}s(p \in s \rightarrow p = 1 \wedge 1 \notin t \vee \exists q{<}p\exists k{\leq}1\exists x{<}q(\\ &\qquad p = (k{+}1){\cdot}q^2 \wedge \mathit{Sgm}(s,p,q,1) \wedge \mathit{Sgm}(t,p,q,x) \wedge \mathit{Sgm}(t,p^2,p,2{\cdot}x{+}k)))\ .\end{aligned}$$

The graph of exponentiation $2^x = y$ is now defined as a Σ_0-predicate as follows:

$$2^x = y \leftrightarrow x < y \wedge \mathit{Pow}_2(y) \wedge \exists s{<}y\exists t{<}y\ \mathit{Seqs}(y + s, x{\cdot}y + t)\ .$$

0.6 Historical remark. We have simplified the presentation (but not the idea) of our note [Vo98] after we have seen a similar (but not identical) encoding of course of values sequences for 2^x in [Imm98]. Immerman attributes the idea to Lindell [Lind9?]. Lindell is dealing with the general graph of exponentiation $x^y = z$ in the context of finite models and he was inspired by Nelson's coding of sequences [Nel86] as presented in [HP93]. Both Lindell and the present author arrived at the key idea of encoding the computation without an auxiliary pairing function independently. For the purposes outlined in Par. 0.1 we do need the graph of $x^y = z$, the graph of $2^x = y$ suffices. This means that the three course of values sequences of Lindell can be replaced by our two where the sequence of powers $y + s$ plays also the role of markers separating the segments in the sequence of exponents $x{\cdot}y + t$. Morever, the quantifier bounds on the sequences are much simpler in our approach than in the approach of Lindell, and both bounds are significantly simpler than those in [HP93].

References

[Ben62] J. H. Bennett. *On Spectra.* Ph.D. dissertation. Princeton, 1962.

[BJ80] G. Boolos, R. Jeffrey. *Computability and Logic.* Cambridge Univ. Press, 1980.

[HP93] P. Hájek and P. Pudlák. *Metamathematics of First-Order Arithmetic.* Springer Verlag, 1993.

[Imm98] N. Immerman. *Descriptive Complexity.* Springer 1998.

[Nel86] E. Nelson. *Predicative Arithmetic.* Mathematical Notes. Princeton Univ. Pres, 1986.

[Lind9?] S. Lindell. *How to define exponentiation from addition and multiplication in first-order logic on finite structures.* Manuscript 199?

[Sho67] J. R. Shoenfield. *Mathematical Logic.* Addison-Wesley, 1967.

[Smu92] R. M. Smullyan. *Gödel's Incompleteness Theorem.* Oxford University Press, 1992.

[Vo98] P. J. Voda. *A Note On the Exponentiation Relation in Peano Arithmetic,* www.fmph.uniba.sk/∼voda, 1998.