# Notations for Exponentiation

Arnold Beckmann [1]

*Mathematical Institute, University of Oxford, 24-29 St. Giles', Oxford OX1 3LB, UK.*

*Institut für Mathematische Logik und Grundlagenforschung, Westfälische Wilhelms-Universität Münster, Einsteinstr. 62, D-48149 Münster, Germany.*

**Abstract**

We define a coding of natural numbers – which we will call *exponential notations* – and interpretations of the less-than-relation, the successor, addition and exponentiation function on exponential notations. We prove that all these interpretations are polynomial time computable. As a corollary we obtain that feasible arithmetic can prove the consistency of the canonical equational theory for the language containing the successor, addition and exponentiation function.[2]

*Key words:* exponentiation, polynomial time computable, course of value recursion, exponential notations, weak arithmetic, consistency
*1991 MSC:* 03D15, 68Q19, 03F25

## Introduction

A necessary condition for a function $f$ to be *feasibly computable* is that it grows at most polynomially (cf. [2] et. al.) which means that there is a polynomial $q_f$ such that $(\forall x)\Big[\log f(x) \le q_f(\log x)\Big]$. This condition is satisfied, e.g., by all functions from the polynomial hierarchy, in particular by the polynomial time computable functions. Therefore, exponentiation is not directly available in weak theories of arithmetic which are related to low-complexity computability, like the bounded arithmetic theories $S^i_2$, $T^i_2$, etc. (cf. [2] or [6] for a definition of these theories).

*Email address:* Arnold.Beckmann@math.uni-muenster.de (Arnold Beckmann).
[1] Supported by the Deutschen Akademie der Naturforscher Leopoldina grant #BMBF-LPD 9801-7 with funds from the Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie.

One possibility of dealing with exponentiation is given for example in [5]. There it is shown that the graph of exponentiation can be defined by a bounded formula, i.e. by a formula in which all quantifiers are bounded.

In this article we will follow another path to handle exponentiation. Our idea is inspired by the proof theoretic analysis of PEANO arithmetic which relates PEANO arithmetic to the ordinal

$$\varepsilon_0 \;=\; \lim_i \omega^{\cdot^{\cdot^{\omega}}} \Big\} \; i\text{-times.}$$

Ordinals less than $\varepsilon_0$ can be represented by terms in the symbols $0$, $+$, $\cdot$, $\lambda\alpha.\omega^\alpha$, which induces a canonical arithmetization $\overline{\varepsilon_0}$ of $\varepsilon_0$ as a subset of the natural numbers. In fact, $\overline{\varepsilon_0}$ and the homomorphic translations of the functions $0, +, \cdot, \lambda\alpha.\omega^\alpha$ onto $\overline{\varepsilon_0}$ are primitive recursive, cf. [7] et. al. We are going – more or less – to replace $\omega$ by 2 in this coding obtaining exponential notations and homomorphic translations on exponential notations of the constant 0, the unary functions successor S, doubling $\mathrm{dbl}(n) = 2 \cdot n$ and exponentiation $\exp(n) = 2^n$, and the binary function addition $+$. They all will be polynomial time computable.

One application of exponential notations can be found in [1]. There they are used for the proof theoretic analysis of weak fragments of arithmetic, called *dynamic ordinal analysis*.

In the last section of this article we will outline another application. BUSS has shown in [2] that arithmetization of metamathematics can be done in bounded arithmetic. But it is difficult to prove consistency even of weak equational theories without the presence of exponentiation. This is because values of closed terms often grow exponentially in their GÖDEL numbers (cf. [3], p.9). In the present article we are going to show that for a certain restriction of the language which does not infer the growth-rate of the functions – for example exponentiation will be in the language – we can feasibly calculate on exponential notations the values of closed terms. Therefore, feasible arithmetic, i.e. BUSS' system $S^1_2$ of bounded arithmetic, can prove the consistency of the equational theory for the language $\{0, \mathrm{S}, +, \mathrm{dbl}, \exp\}$ involving only equations between closed terms, which is axiomatized by the recursive definitions of the function symbols.

In the next section we introduce the polynomial time computable (polytime) functions and repeat a feasible GÖDEL-numbering from [2]. Furthermore, we need some more closure properties of polytime functions, i.e. we will prove that the polytime functions are closed under a limited course-of-values recursion. In section 2 we define the exponential notations and functions manipulating them. Using the closure properties from section 1 we show that most of these functions and predicates are polytime.[2] Finally we show that there exists a polytime function on exponential notations which interprets terms over the language $\{0, \mathrm{S}, +, \mathrm{dbl}, \exp\}$. As a corollary we obtain the above described consistency in Buss' system $S^1_2$.

## 1 Limited course-of-values recursion

We start giving a brief review of the polytime functions and the polytime sequence coding from [2]. We will end this section proving that the polytime functions are closed under a certain (feasible) course-of-value recursion.

Let S, $+$, $\cdot$ be the usual successor, addition and multiplication functions. Let $\mathrm{S}_0$ and $\mathrm{S}_1$ denote the binary successor functions given by $\mathrm{S_i}(n) = 2 \cdot n + i$ with $i \in \{0, 1\}$. The *binary length function* $|n|$, which computes the number of bits in the binary representation of $n$, is given by $|n| = \lceil \log_2(n + 1)\rceil$. For real numbers $r$, $\lceil r\rceil$ is the least integer $z$ which is bigger than or equal to $r$. The smash function $\#$ is given by $m \# n = 2^{|m|\cdot|n|}$. Let dbl and exp denote the *doubling function* $\mathrm{dbl}(n) = 2 \cdot n = \mathrm{S}_0(n)$ and, respectively, the *exponentiation function* $\exp(n) = 2^n$.

The *polytime functions* can be defined as the set of all functions $f$ which can be computed by a Turing machine $M_f$ such that the runtime is bounded by a polynomial $p$ in the length of the input, i.e. $M_f$ needs on input $n$ at most $p(|n|)$ steps to calculate $f(n)$.

There are also algebraic characterizations of the polytime functions. All polytime functions are generated from basic functions 0, S, $+$, $\cdot$, $\lambda n.|n|$, $\#$ using composition and one of the following rules of *limited recursion on notations* (cf. [4] p. 28) or *limited recursion* (cf. [2] p. 8).

The function $f$ is defined from functions $g$, $h_0$, $h_1$ and $k$ by *limited recursion on notation* if

$$\begin{aligned} f(\vec{x}, 0) &= g(\vec{x}) \\ f(\vec{x}, \mathrm{S_i}(y)) &= h_i(\vec{x}, y, f(\vec{x}, y)) \quad (i = 0, 1; \quad i \neq 0 \text{ if } y = 0) \end{aligned}$$

provided that $f(\vec{x}, y) \leq k(\vec{x}, y)$ for all $\vec{x}, y$.

See Rose [8] for a proof that this rule again defines polytime functions.

[2] The results described in the first two sections are part of the author's dissertation [1].

The function $f$ is defined from functions $g$, $h$ and polynomials $p$ and $q$ by *limited recursion* if the following holds:
Let the function $\tau$ be defined as

$$\begin{aligned}\tau(\vec{x},0) &= g(\vec{x})\\ \tau(\vec{x},\mathrm{S}(y)) &= h(\vec{x},y,\tau(\vec{x},y)).\end{aligned}$$

Then let
$$f(\vec{x}) \;=\; \tau(\vec{x},p(|\vec{x}|))$$
provided that $|\tau(\vec{x},y)| \le q(|\vec{x}|)$ for all $\vec{x}$ and $y \le p(|\vec{x}|)$).

See Buss [2] for a proof that this rule again defines polytime functions.

We sometimes use a dyadic notation of the natural numbers: let $i_j \in \{0,1\}$ for $j \le k$, then we define

$$(i_k \dots i_0)_2 \;:=\; \sum_{j=0}^{k} i_j \cdot 2^j.$$

We write $(s.i_k \dots i_0)_2$ for $s \cdot 2^{k+1} + (i_k \dots i_0)_2$.

Let $\langle \dots \rangle$ be a feasible Gödel *numbering* of sequences as defined in [2] p. 8 with the change that we do not reverse the order of the bits. The following equations define such a coding. First we define a function $s * a$ for $s, a \in \omega$ by limited recursion on the notation of $a$. This function adds the value $a$ to the sequence $s$.

$$\begin{aligned} s*0 &= (s.0010)_2 &&= 16\cdot s+2\\ s*1 &= (s.0011)_2 &&= 16\cdot s+3\\ s*(a.i)_2 &= ((s*a).1i)_2 &&= 4\cdot(s*a)+2+i, \quad (i=0,1 \text{ and } a\neq 0).\end{aligned}$$

The Gödel numbers are inductively given by

$$\begin{aligned}\langle\rangle &= 0\\ \langle a_1,\dots,a_k,a_{k+1}\rangle &= \langle a_1,\dots,a_k\rangle * a_{k+1}.\end{aligned}$$

Let `Seq` be the polytime *set of all* Gödel *numbers*.

How does feasible Gödel numbering work? The Gödel number for the sequence $a_1,\dots,a_k$ is constructed as follows. First we write the $a_i$'s in binary notation so that we obtain a string of 0's, 1's and commas. Then we replace each 0 by "10", each 1 by "11" and each comma by "00". The resulting string of zeros and ones is the binary representation of the Gödel number $\langle a_1,\dots,a_k\rangle$. For example the Gödel number of $3, 4, 5$ is $(11110011101000111011)_2$ or 997.947. $\langle\rangle$ is defined to be 0.

In the following we introduce some polytime functions which manipulate GÖDEL numbers (cf. [2]).

$$\begin{aligned}\langle a_1,\dots,a_k\rangle ** \langle b_1,\dots,b_l\rangle &= \langle a_1,\dots,a_k,b_1,\dots,b_l\rangle\\ \beta(0,\langle a_1,\dots,a_k\rangle) &= k\\ \mathrm{lh}(\langle a_1,\dots,a_k\rangle) &= k\\ \beta(i,\langle a_1,\dots,a_k\rangle) &= a_i, \quad \text{for } 1\le i\le k\\ \mathrm{SqBd}(k,l) &= (k \# \mathrm{S}_1(\mathrm{S}_1(l)))^2.\end{aligned}$$

SqBd has the property

$$\forall a_1,\dots,a_k{\le}l\ \Big(\langle a_1,\dots,a_k\rangle \le \mathrm{SqBd}(2^k,l)\Big).$$

In the sequel we will use limited recursion (on notation) to define polytime functions. In doing so we often use $\mathrm{lh}(s)$ to bound recursion. This is allowed since $\mathrm{lh}(s)\le |s|$.

In the following sections we need – beside limited recursion (on notation) – a rule which is similar to course-of-value recursion, and generates polytime functions. The usual course-of-values recursion is equivalent to primitive recursion. Thus, in general, polytime functions are not closed under this rule. Another, more technical, aspect is that $\lambda n.\langle 0,1,\dots,n-1\rangle$ growths exponentially. Therefore, one requirement of limited course-of-values recursion is that the course is given by a polytime function.

In the following let $s \sqsubset t$ mean that $s,t$ are GÖDEL numbers and $s$ is a subsequence of $t$, i.e., if $\mathrm{lh}(s)=k$ and $t=\langle t_0,\dots,t_{l-1}\rangle$ then $k\le l$ and

$$\exists i_0,\dots,i_{k-1}\ \Big(i_0<\dots<i_{k-1}<l \text{ and } s=\langle t_{i_0},\dots,t_{i_{k-1}}\rangle\Big).$$

**Definition 1.1** *An unary function* course *is a* course-function *if it satisfies*

$$\mathrm{course}(s) \sqsubset \langle 0,\dots,s-1\rangle$$

*and*

$$\mathrm{course}(s)=\langle s_0,\dots,s_{k-1}\rangle \quad\Rightarrow\quad \forall i{<}k\ \Big(\mathrm{course}(s_i)\sqsubset\langle s_0,\dots,s_{i-1}\rangle\Big).$$

*The* course-of-values of a function $f$ according to course *is defined by*

$$f^{\mathrm{course}}(s) \ :=\ \langle f(s_0),\dots,f(s_{k-1})\rangle$$

*provided that* $\mathrm{course}(s)=\langle s_0,\dots,s_{k-1}\rangle$.

If $f$ and course are polytime then also $f^{\mathrm{course}}$ is polytime. This can be seen, using limited recursion, by a similar argument as in the following theorem.

**Theorem 1.2 (limited course-of-values recursion)**
*Let* course *be a course-function. Given a function $g$ there exists a uniquely defined function $f$ solving*

$$f(s) = g(s, f^{\text{course}}(s)).$$

*If in addition* course *and $g$ are polytime and there exists another polytime function $h$ satisfying*

$$f(s) \leq h(s)$$

*then this $f$ is polytime, too.*

**Proof**: Existence and uniqueness are proved as usual. For the second part of the theorem we define the function

$$\text{select}(\langle a_0, \dots , a_{k-1}\rangle, \langle a_{i_1}, \dots , a_{i_r}\rangle, \langle b_0, \dots , b_{l-1}\rangle) \ := \ \langle b_{i_1}, \dots , b_{i_r}\rangle$$

for an increasing sequence $\langle a_0, \dots , a_{k-1}\rangle$, $i_1 < \dots < i_r < \min(k, l)$. Using functions

$$\text{b}(x) \ := \ \begin{cases} \langle \alpha, \beta, \gamma, \delta * c\rangle & : \ x = \langle \alpha * a, \beta * a, \gamma * c, \delta\rangle \\ \langle \alpha, \beta * b, \gamma, \delta\rangle & : \ x = \langle \alpha * a, \beta * b, \gamma * c, \delta\rangle \text{ and } a \neq b \\ x & : \text{otherwise} \end{cases}$$

and

$$\begin{aligned} \text{r}(\langle a_1, \dots , a_k\rangle) \ &:= \ \langle a_k, \dots , a_1\rangle \\ \text{select}(\alpha, \beta, s) \ &:= \ \beta(4, \text{b}^{(\text{lh}(\alpha))}(\langle \text{r}(\alpha), \text{r}(\beta), \text{r}(s), \langle\rangle\rangle)) \ \leq \ s \end{aligned}$$

we observe that select is polytime by limited recursion. Here $\text{b}^{(x)}(a)$ is the $x$-fold iteration of $\lambda n.b(n)$ applied to $a$.

In order to prove the assertion it suffices to show that $f^{\text{course}}$ is polytime. Let $t = \text{course}(s) = \langle b_0, \dots , b_{l-1}\rangle$. We define a polytime function $\tilde{\phi}(t, i) = \langle f(b_0), \dots , f(b_{i-1})\rangle$. To this end we observe for $i < l$

$$f(b_i) = g(b_i, f^{\text{course}}(b_i)) = g(b_i, \text{select}(t, \text{course}(b_i), \tilde{\phi}(t, i))).$$

So we define

$$\begin{aligned} \tilde{\phi}(t, 0) \ &:= \ \langle\rangle \\ \tilde{\phi}(t, i + 1) \ &:= \ \tilde{\phi}(t, i) * g(\beta(i + 1, t), \text{select}(t, \text{course}(\beta(i + 1, t)), \tilde{\phi}(t, i))) \\ \phi(t) \ &:= \ \tilde{\phi}(t, \text{lh}(t)) \ \leq \ h^{\text{course}}(t) \\ f^{\text{course}}(s) \ &:= \ \phi(\text{course}(s)). \end{aligned}$$

By limited recursion $f^{\text{course}}$ is polytime. □

## 2 Exponential notations for natural numbers

We start this section by coding natural numbers as formal trees over the alphabet consisting only of the two bracket symbols [, ], and predicates and functions on them. We discuss how these formal predicates and functions can be seen as "real" predicates and functions, i.e. predicates and functions on the natural numbers. Finally we apply the closure properties from the previous section showing that most of the "real" predicates and functions are polytime.

We define

$$\begin{aligned} \hat{0} &:= [\,] \\ \check{2}^{\alpha_1}\check{+}\ldots\check{+}\check{2}^{\alpha_k} &:= [\alpha_1,\ldots,\alpha_k]. \end{aligned}$$

The intended meaning of these terms becomes clear from the evaluation function which is given by

$$\begin{aligned} \Phi(\hat{0}) &= 0 \\ \Phi(\check{2}^{\alpha_1}\check{+}\ldots\check{+}\check{2}^{\alpha_k}) &= 2^{\Phi(\alpha_1)}+\ldots+2^{\Phi(\alpha_k)}. \end{aligned}$$

Now we define the predicates $\mathcal{E}$, $\prec$ and the functions $\Phi_{\mathcal{E}}$, $\mathrm{T}_{\mathcal{E}}$ by the following:

$$\begin{aligned} \alpha\in\mathcal{E} &\Leftrightarrow \alpha=\hat{0} \text{ or there are } \alpha_1,\ldots,\alpha_k\in\mathcal{E} \text{ with} \\ &\phantom{\Leftrightarrow}\ \alpha=\check{2}^{\alpha_1}\check{+}\ldots\check{+}\check{2}^{\alpha_k} \text{ and } \Phi(\alpha_k)<\ldots<\Phi(\alpha_1); \\ \alpha\prec\beta &\Leftrightarrow \alpha,\beta\in\mathcal{E} \text{ and } \Phi(\alpha)<\Phi(\beta); \\ \Phi_{\mathcal{E}} &:= \Phi\restriction\mathcal{E}, \text{ the restriction of } \Phi \text{ to } \mathcal{E}; \\ \mathrm{T}_{\mathcal{E}} &:= {\Phi_{\mathcal{E}}}^{-1}, \text{ the inverse function to } \Phi. \end{aligned}$$

$\mathcal{E}$ will be called the *set of exponential notations*. In the sequel we use small Greek letters representing exponential notations.

We define the functions $\hat{+}$, $\widehat{\mathrm{dbl}}$ and $\widehat{\exp}$ on $\mathcal{E}$ by:

$$\begin{aligned} \alpha\,\hat{+}\,\beta &:= \mathrm{T}_{\mathcal{E}}(\Phi_{\mathcal{E}}(\alpha)+\Phi_{\mathcal{E}}(\beta)) \\ \widehat{\mathrm{dbl}}(\alpha) &:= \mathrm{T}_{\mathcal{E}}(2\cdot\Phi_{\mathcal{E}}(\alpha)) \\ \widehat{\exp}(\alpha) &:= \mathrm{T}_{\mathcal{E}}(2^{\Phi_{\mathcal{E}}(\alpha)}). \end{aligned}$$

We observe that the desired exponentiation function on $\mathcal{E}$, $\widehat{\exp}$, can be written simply as $\lambda\alpha.[\alpha]$.

There are several possibilities of coding syntax. Of course one has to use a feasible sequence coding in order to obtain $\mathcal{E}$, $\prec$, $\hat{+}$, $\widehat{\mathrm{dbl}}$, $\widehat{\exp}$, $\mathrm{T}_{\mathcal{E}}$ as polytime predicates and functions. But even if we fix such one, as we did in the last section, we still have the choice between a flat or unflat coding of trees. On the one hand a flat coding $\mathrm{c_f}$ would look like

$$\mathrm{c_f}([a_1,\dots,a_k]) \;:=\; \langle[\rangle \mathbin{**} \mathrm{c_f}(a_1) \mathbin{**} \dots \mathbin{**} \mathrm{c_f}(a_k) \mathbin{**} \langle]\rangle,$$

where brackets $[,]$ on the right side of the equation sign are identified with suitable GÖDEL numbers. On the other hand an unflat coding $\mathrm{c_{uf}}$ is obtained by

$$\mathrm{c_{uf}}([a_1,\dots,a_k]) \;:=\; \langle \mathrm{c_{uf}}(a_1),\dots,\mathrm{c_{uf}}(a_k)\rangle.$$

The unflat coding usually is less efficient than the flat one, but more feasible, as for example the depth of the tree $t$ is bounded by $||t||$ in case of the unflat coding. This is not true in case of the flat coding.

Comparing different kinds of codings one gets the impression that one should consider under all possible feasible codings the most uneffective one which is still good enough for one's purpose, e.g. proving GÖDEL's incompleteness results for weak theories of arithmetic. Here in this article we will therefore restrict ourselves to the unflat coding of syntax. The author conjectures that the same GÖDEL's incompleteness results as in [2] can also be achieved with this kind of unflat coding.

Using $\mathrm{c_{uf}}$ we can view $\mathcal{E}$, $\prec$, $\hat{+}$, $\widehat{\mathrm{dbl}}$, $\widehat{\exp}$, $\Phi_{\mathcal{E}}$ and $\mathrm{T}_{\mathcal{E}}$ as predicates and functions on natural numbers. We are going to show that all of them but $\Phi_{\mathcal{E}}$ are polytime. This is easy for $\widehat{\exp}$, which can be written as $\lambda\alpha.\langle\alpha\rangle$. The reason why $\Phi_{\mathcal{E}}$ cannot be polytime is simply the following: Let

$$f_i(n) \;:=\; \underbrace{\widehat{\exp}(\dots\widehat{\exp}}_{i\text{-times}}(\mathrm{T}_{\mathcal{E}}(n))\dots)$$

then we compute

$$\Phi_{\mathcal{E}}(f_i(n)) = \left.2^{\cdot^{\cdot^{2^n}}}\right\} i\text{-times}.$$

After having seen that $\mathrm{T}_{\mathcal{E}}$ is polytime this shows that $\Phi_{\mathcal{E}}$ cannot be polytime.

We need some special course functions which compute certain subsequences of exponential notations. They are needed in the definition of $\mathcal{E}$, $\prec$, $\hat{+}$, $\widehat{\mathrm{dbl}}$, $\widehat{\exp}$ and $\mathrm{T}_{\mathcal{E}}$. We start defining

$$\mathrm{sort}(\langle a_1,\dots,a_k\rangle) \;:=\; \langle b_1,\dots,b_l\rangle$$

where $\{a_1,\dots,a_k\} = \{b_1,\dots,b_l\}$ and $b_1 < \dots < b_l$. sort can be computed using one of the commonly known sorting algorithms, e.g., one which runs in time $O(n^2)$ sorting $n$ objects. Thus, $\mathrm{sort}(s)$ is computable in time $O(|s|^2)$, hence polytime.

Now we define

$$\mathrm{U}(\langle\langle a_{1\,1},\dots,a_{1\,i_1}\rangle,\dots,\langle a_{k\,1},\dots,a_{k\,i_k}\rangle\rangle) \;:=\; \langle b_1,\dots,b_l\rangle$$

where $b_1 < \dots < b_l$ and

$$\{b_1,\dots,b_l\} \;=\; \{a_{1\,1},\dots,a_{1\,i_1},\dots,a_{k\,1},\dots,a_{k\,i_k}\}.$$

The following equations may be used to observe that U is polytime. Let $s = \langle s_0,\dots,s_{k-1}\rangle$.

$$\begin{aligned} f(\langle s_0,\dots,s_{k-1}\rangle) &:= s_0 \,{}^{**}\dots{}^{**}\, s_{k-1} \;\le\; \mathrm{SqBd}(s,s) \\ \mathrm{U}(s) &:= \mathrm{sort}(f(s)). \end{aligned}$$

By limited recursion $f$ is polytime, thus also U. We use these functions to see that the *transitive closure*[3] of a sequence can be computed by a polytime function. To this end, we observe that $\mathrm{U}^{|s|}(s) = \langle\rangle$, and we define

$$g(s) \;:=\; s \,{}^{**}\, \mathrm{U}(s) \,{}^{**}\, \mathrm{U}(\mathrm{U}(s)) \,{}^{**}\dots{}^{**}\, \mathrm{U}^{|s|}(s) \;\le\; \mathrm{SqBd}(s \# s, s)$$

then $g$ is polytime by limited recursion. Hence

$$\mathrm{tc}(s) \;:=\; \mathrm{sort}(g(s))$$

is polytime and computes the transitive closure of $s$. By construction tc is a course function.

We need a similar course-function for pairs of sequences.
Let $\mathrm{tc}_2(\langle s,t\rangle) = \langle c_1,\dots,c_k\rangle$ with $c_1 < \dots < c_k$ and

$$\{c_1,\dots,c_k\} \;=\; \{\langle d_i, e_j\rangle \;:\; 1 \le i \le m, 1 \le j \le n\}$$

where $\mathrm{tc}(s) = \langle d_1,\dots,d_m\rangle$ and $\mathrm{tc}(t) = \langle e_1,\dots,e_n\rangle$. The following equations are used to observe that $\mathrm{tc}_2$ is polytime. Let $s = \langle s_0,\dots,s_{k-1}\rangle$ and let $t = \langle t_0,\dots,t_{l-1}\rangle$.

$$\begin{aligned} f(\langle s_0,\dots,s_{k-1}\rangle, a) &:= \langle\langle s_0,a\rangle,\dots,\langle s_{k-1},a\rangle\rangle \;\le\; \mathrm{SqBd}(s, s * a) \\ X(s, \langle t_0,\dots,t_{l-1}\rangle) &:= f(s,t_0) \,{}^{**}\dots{}^{**}\, f(s,t_{l-1}) \;\le\; \mathrm{SqBd}(s \# t, s \,{}^{**}\, t) \\ \mathrm{tc}_2(\langle s,t\rangle) &:= \mathrm{sort}(X(\mathrm{tc}(s),\mathrm{tc}(t))). \end{aligned}$$

By limited recursion both $f$ and $X$ are polytime. Thus, also $\mathrm{tc}_2$ is polytime. By construction $\mathrm{tc}_2$ is a course function.

[3] The *transitive closure* is generated using the obvious element relation on sequences which is given by $a_i$ *is an element of* $\langle a_1,\dots,a_k\rangle$, $0 < i \le k$.

We use $\mathrm{tc}_2$ to show that $\mathcal{E}$ and $\prec$ are polytime.

$$\begin{array}{lcl}
\alpha \in \mathcal{E} & \Leftrightarrow & \alpha = \check{2}^{\alpha_1}\check{+}\ldots\check{+}\check{2}^{\alpha_k} \text{ with } \alpha_1,\ldots,\alpha_k \in \mathcal{E} \text{ and } \alpha_k \prec \ldots \prec \alpha_1. \\
\alpha \prec \beta & \Leftrightarrow & \alpha,\beta \in \mathcal{E}, \alpha = \check{2}^{\alpha_1}\check{+}\ldots\check{+}\check{2}^{\alpha_k}, \beta = \check{2}^{\beta_1}\check{+}\ldots\check{+}\check{2}^{\beta_l} \text{ and} \\
 & & \exists i{<}l\,(i \le k,\ \alpha_1 = \beta_1,\ldots,\alpha_i = \beta_i \text{ and } (i = k \text{ or } \alpha_{i+1} \prec \beta_{i+1})).
\end{array}$$

We cannot apply Theorem 1.2 directly to this simultaneous definition because if we try to compute $\hat{0} \prec \check{2}^{\beta_1}\check{+}\check{2}^{\beta_2} =: \beta$ we need $\beta \in \mathcal{E}$ and for this $\beta_2 \prec \beta_1$. But $\langle \beta_2, \beta_1 \rangle$ does not occur in $\mathrm{tc}_2(\langle \hat{0}, \beta \rangle)$. Surely it is possible to change the definition of $\mathrm{tc}_2$ to overcome this lack, as $\langle \beta_2, \beta_1 \rangle < \langle \hat{0}, \beta \rangle$. But there is another possibility to show that $\mathcal{E}$ and $\prec$ are polytime which uses Theorem 1.2 and $\mathrm{tc}_2$. We define a more general relation $\prec'$. We obtain $\prec'$ by replacing $\mathcal{E}$ by `Seq` (the set of all GÖDEL numbers) in the definition of $\prec$. Let $\chi_{\prec'}$ be the characteristic function of $\prec'$, i.e.,

$$\chi_{\prec'}(\alpha,\beta) \;=\; \begin{cases} 1 : \alpha \prec' \beta \\ 0 : \mathit{otherwise}, \end{cases}$$

and let $h(\langle \alpha, \beta \rangle) := \chi_{\prec'}(\alpha,\beta)$. Rewriting the definition of $\prec'$ we obtain a polytime function $g$ satisfying

$$h(\langle \alpha,\beta \rangle) \;=\; g\Big(\langle \alpha,\beta \rangle, h^{\mathrm{tc}_2}(\langle \alpha,\beta \rangle)\Big) \;\le\; 1,$$

therefore, Theorem 1.2 yields that $h$ is polytime, thus also $\chi_{\prec'}$ and hence $\prec'$ are polytime. Now we define

$$\begin{array}{lcl}
\alpha \in \mathcal{E} & \Leftrightarrow & \mathtt{Seq}(\alpha) \text{ and} \\
 & & \forall i < \mathrm{lh}(\alpha)\Big[\beta(i+1,\alpha) \in \mathcal{E} \text{ and } (i > 0 \rightarrow \beta(i+1,\alpha) \prec' \beta(i,\alpha))\Big] \\
\alpha \prec \beta & \Leftrightarrow & \alpha \in \mathcal{E} \text{ and } \beta \in \mathcal{E} \text{ and } \alpha \prec' \beta.
\end{array}$$

Using Theorem 1.2 with tc we obtain that $\mathcal{E}$ is polytime. Therefore, also $\prec$ is polytime.

Before we can define $\hat{+}$ on the exponential notations we need a successor function $\hat{\mathrm{S}}$ on them. To compute the successor of an exponential notation we need an auxiliary function $F$ to manage carries. Therefore, we simultaneously define for $\alpha = \check{2}^{\alpha_1}\check{+}\ldots\check{+}\check{2}^{\alpha_k} \in \mathcal{E}$

$$F(\alpha) \;:=\; \mu i \le k.\Big(i > 0 \text{ and } \forall j{<}k\,(j \ge i \rightarrow \alpha_j = \hat{\mathrm{S}}(\alpha_{j+1}))\Big)$$

$$\hat{\mathrm{S}}(\alpha) \;:=\; \begin{cases} \check{2}^{\alpha_1}\check{+}\ldots\check{+}\check{2}^{\alpha_{i-1}}\check{+}\check{2}^{\hat{\mathrm{S}}\,\alpha_i} & : \alpha_k = \hat{0} \text{ and } i := F(\alpha) \\ \check{2}^{\alpha_1}\check{+}\ldots\check{+}\check{2}^{\alpha_k}\check{+}\check{2}^{\hat{0}} & : \text{otherwise.} \end{cases}$$

Clearly $F(\alpha) \le k = \mathrm{lh}(\alpha)$ and after proving $|\,\hat{\mathrm{S}}(\alpha)| \le |\alpha * \hat{0}|$ we can use Theorem 1.2 together with tc to see that both functions are polytime.

**Lemma 2.1** $|\,\hat{\mathrm{S}}(\alpha)| \le |\alpha * \hat{0}| \le |\alpha| + 4$.

**Proof**: Remember the definition

$$\begin{array}{rcccl} s * 0 & = & (s.0010)_2 & = & 16 \cdot s + 2, \\ s * 1 & = & (s.0011)_2 & = & 16 \cdot s + 3, \\ s * (a.i)_2 & = & ((s * a).1i)_2 & = & 4 \cdot (s * a) + 2 + i, \quad (i = 0, 1 \text{ and } a \neq 0) \end{array}$$

and

$$\langle a_1, \ldots, a_k, a_{k+1} \rangle = \langle a_1, \ldots, a_k \rangle * a_{k+1}.$$

First we compute some constant notations and some binary lengths. Let $a = (a_1 \ldots a_k)_2$.

$$\begin{array}{rcl} \hat{0} & = & (0)_2 = 0 \\ \hat{\mathrm{S}}(\hat{0}) & = & (10)_2 = 2 \\ s \neq 0 & \Rightarrow & |s * 0| = |\,(s.0010)_2\,| = |s| + 4 \\ a \neq 0 & \Rightarrow & |s * a| = |\,(s.001a_1 1a_2 \ldots 1a_k)_2\,| = |\,(s.00)_2\,| + 2 \cdot |a| \\ & = & \begin{cases} 2 \cdot |a| & : s = 0 \\ |s| + 2 + 2 \cdot |a| & : s \neq 0. \end{cases} \end{array}$$

We prove the assertion by induction on $\alpha = \check{2}^{\alpha_1} \check{+} \ldots \check{+} \check{2}^{\alpha_k} = \langle \alpha_1, \ldots, \alpha_k \rangle$. If $k = 0$, then $\alpha = \hat{0}$, hence $\hat{\mathrm{S}}(\hat{0}) = \check{2}^{\hat{0}} = \hat{0} * \hat{0}$. If $k > 0$ and $\alpha_k \neq \hat{0}$, then $\hat{\mathrm{S}}(\alpha) = \langle \alpha_1, \ldots, \alpha_k, \hat{0} \rangle = \alpha * \hat{0}$. If $k > 0$ and $\alpha_k = \hat{0}$, then let $i := F(\alpha)$. We have to distinguish the following cases. Let $\beta := \langle \alpha_1, \ldots, \alpha_{i-1} \rangle$.
If $i = k$ then we observe $\alpha = \beta * \hat{0}$ and

$$\hat{\mathrm{S}}(\alpha) = \beta * \hat{\mathrm{S}}(\hat{0}) = \beta * (10)_2 = (\beta.001110)_2\,.$$

On the other hand we see

$$\alpha * \hat{0} = (\beta * \hat{0}) * \hat{0} = (\beta.00100010)_2 > \hat{\mathrm{S}}(\alpha).$$

If $i < k$ then we find $\alpha = \beta ** \langle \alpha_i, \ldots, \alpha_k \rangle$. Observe that $\Phi_{\mathcal{E}}(\alpha_j) = k - j$ for $j = i, \ldots, k$, hence $\alpha_i \neq 0$. Now the induction hypothesis produces

$$|\,\hat{\mathrm{S}}(\alpha_i)| \le |\alpha_i * \hat{0}|. \tag{1}$$

This leads to

$$\begin{array}{rcl} |\,\hat{\mathrm{S}}(\alpha)| & = & |\beta * \hat{\mathrm{S}}(\alpha_i)| = |\,(\beta.00)_2\,| + 2 \cdot |\,\hat{\mathrm{S}}(\alpha_i)| \overset{(1)}{\le} |\,(\beta.00)_2\,| + 2 \cdot |\alpha_i * \hat{0}| \\ & = & |\,(\beta.00)_2\,| + 2 \cdot (|\alpha_i| + 4) = |\,(\beta.00)_2\,| + 2 \cdot |\alpha_i| + 8 \end{array}$$

and

$$\begin{array}{rcl} |\alpha * \hat{0}| &=& |(\beta ** \langle \alpha_i, \ldots, \alpha_k \rangle) * \hat{0}| \;\geq\; |\beta ** \langle \alpha_i, \hat{0}, \hat{0} \rangle| \;=\; |((\beta * \alpha_i) * \hat{0}) * \hat{0}| \\ &=& |\beta * \alpha_i| + 8 \;=\; |\,(\beta.00)_2\,| + 2 \cdot |\alpha_i| + 8. \end{array}$$

These two estimations together show $|\,\hat{\mathrm{S}}(\alpha)| \leq |\alpha * \hat{0}|$. □

We define the preaddition pa which computes $\alpha \,\hat{+}\, \check{2}^{\beta}$ by

$$\mathrm{pa}(\alpha, \beta) \;:=\; \begin{cases} \check{2}^{\alpha_1} \check{+} \ldots \check{+} \check{2}^{\alpha_k} \check{+} \check{2}^{\beta} & : k = 0 \text{ or } \beta \prec \alpha_k \\ \check{2}^{\alpha_1} \check{+} \ldots \check{+} \check{2}^{\alpha_{i-1}} \check{+} \check{2}^{\hat{\mathrm{S}}(\alpha_i)} & : \alpha_k = \beta \text{ and } i := F(\alpha) \\ \mathrm{pa}\left(\check{2}^{\alpha_1} \check{+} \ldots \check{+} \check{2}^{\alpha_{k-1}}, \beta\right) * \alpha_k & : \alpha_k \prec \beta \end{cases}$$

where $\alpha = \check{2}^{\alpha_1} \check{+} \ldots \check{+} \check{2}^{\alpha_k}$. In the next lemma we will see that pa is polynomially bounded. Therefore, we can apply Theorem 1.2 together with the following polytime course function initseq to observe that pa is polytime.

$$\mathrm{initseq}(\langle a_1, \ldots, a_k \rangle) \;:=\; \langle \langle \rangle, \langle a_1 \rangle, \ldots, \langle a_1, \ldots, a_{k-1} \rangle \rangle.$$

**Lemma 2.2** *$|\,\mathrm{pa}(\alpha, \beta)| \;\leq\; |\alpha| + 2 \cdot |\beta| + 8.$*

**Proof**: We use induction on $\alpha = \check{2}^{\alpha_1} \check{+} \ldots \check{+} \check{2}^{\alpha_k}$. If $k = 0$ or $\beta \prec \alpha_k$, then

$$|\,\mathrm{pa}(\alpha, \beta)| \;=\; |\alpha * \beta| \;\leq\; |\alpha| + 2 + 2 \cdot |\beta|.$$

If $\alpha_k = \beta$ then let $i := F(\alpha)$ and observe using $\gamma := \check{2}^{\alpha_1} \check{+} \ldots \check{+} \check{2}^{\alpha_{i-1}}$

$$\begin{array}{rcl} |\,\mathrm{pa}(\alpha, \beta)| &=& |\gamma * \hat{\mathrm{S}}(\alpha_i)| \;\leq\; |\,(\gamma.00)_2\,| + 2 \cdot (|\alpha_i| + 4) \\ &=& |\,(\gamma.00)_2\,| + 2 \cdot |\alpha_i| + 8 \;=\; |\gamma * \alpha_i| + 8 \;\leq\; |\alpha| + 8. \end{array}$$

Otherwise, the induction hypothesis (i.h.) shows

$$\begin{array}{rcl} |\,\mathrm{pa}(\alpha, \beta)| &=& |\,\mathrm{pa}(\check{2}^{\alpha_1} \check{+} \ldots \check{+} \check{2}^{\alpha_{k-1}}, \beta) * \alpha_k| \\ &=& |\,\mathrm{pa}(\check{2}^{\alpha_1} \check{+} \ldots \check{+} \check{2}^{\alpha_{k-1}}, \beta)| + 2 + 2 \cdot \max(|\alpha_k|, 1) \\ &\overset{i.h.}{\leq}& |\check{2}^{\alpha_1} \check{+} \ldots \check{+} \check{2}^{\alpha_{k-1}}| + 2 \cdot |\beta| + 8 + 2 + 2 \cdot \max(|\alpha_k|, 1) \\ &=& |\alpha| + 2 \cdot |\beta| + 8. \end{array}$$

□

Now we are able to define by limited recursion

$$\alpha \,\hat{+} (\check{2}^{\beta_1} \check{+} \ldots \check{+} \check{2}^{\beta_l}) \;:=\; \mathrm{pa}(\ldots \mathrm{pa}(\alpha, \beta_1) \ldots, \beta_l)$$

which is limited because

$$\begin{aligned} |\alpha \hat{+} \beta| &= |\,\mathrm{pa}(\ldots \mathrm{pa}(\alpha, \beta_1) \ldots, \beta_l)| \;\leq\; |\alpha| + 2 \cdot |\beta_1| + 8 + \ldots + 2 \cdot |\beta_l| + 8 \\ &\leq\; |\alpha| + |\beta| + 8 \cdot l \;\leq\; |\alpha| + 9 \cdot |\beta|. \end{aligned}$$

Therefore, $\hat{+}$ is polytime.

Now we show that $\widehat{\mathrm{dbl}}$ is polytime. We define by limited recursion on $\alpha = \check{2}^{\alpha_1} \check{+} \ldots \check{+} \check{2}^{\alpha_k}$

$$\widehat{\mathrm{dbl}}(\alpha) \;:=\; \check{2}^{\hat{\mathrm{S}}(\alpha_1)} \check{+} \ldots \check{+} \check{2}^{\hat{\mathrm{S}}(\alpha_k)}$$

and compute

$$\begin{aligned} |\,\widehat{\mathrm{dbl}}(\alpha)| &= 2 \cdot |\,\hat{\mathrm{S}}(\alpha_1)| + 2 + \ldots + 2 + 2 \cdot |\,\hat{\mathrm{S}}(\alpha_k)| \\ &\leq 2 \cdot (|\alpha_1| + 4) + 2 + \ldots + 2 + 2 \cdot (|\alpha_k| + 4) \\ &= |\alpha| + 8 \cdot k \;\leq\; 9 \cdot |\alpha|. \end{aligned}$$

Hence $\widehat{\mathrm{dbl}}$ is polytime.

Finally we want to observe that

$$\mathrm{T}_{\mathcal{E}}(n) \;=\; {\Phi_{\mathcal{E}}}^{-1}(n) \;=\; \text{"the unique } \alpha \in \mathcal{E} \text{ with } \Phi_{\mathcal{E}}(\alpha) = n\text{"}$$

is polytime. Using $\widehat{\mathrm{dbl}}$ we define, this time by limited recursion on notation,

$$\begin{aligned} \mathrm{T}_{\mathcal{E}}(0) &:= \hat{0} \\ \mathrm{T}_{\mathcal{E}}(\,(n.i)_2\,) &:= \begin{cases} \widehat{\mathrm{dbl}}(\mathrm{T}_{\mathcal{E}}(n)) & : i = 0 \\ \hat{\mathrm{S}}(\widehat{\mathrm{dbl}}(\mathrm{T}_{\mathcal{E}}(n))) & : i = 1. \end{cases} \end{aligned}$$

With the next lemma we obtain that $\mathrm{T}_{\mathcal{E}}$ is polytime.

**Lemma 2.3** $|\mathrm{T}_{\mathcal{E}}(n)| \leq 8 \cdot |n|^2$.

**Proof**: We use induction on $n$. If $n = 0$, then $|\mathrm{T}_{\mathcal{E}}(0)| = |\hat{0}| = 0 = 8 \cdot |0|^2$. If $n = 1$, then $|\mathrm{T}_{\mathcal{E}}(1)| = |\,\hat{\mathrm{S}}(\hat{0})| = |2| = 2 \leq 8 \cdot |1|^2$. For the induction step we consider $(n.i)_2$ with $i = 0, 1$ and $n \geq 1$. In general we have $\mathrm{lh}(\alpha) \leq |\,\Phi_{\mathcal{E}}(\alpha)|$, hence $\mathrm{lh}(\mathrm{T}_{\mathcal{E}}(n)) \leq |n|$. Now we estimate

$$\begin{aligned} |\mathrm{T}_{\mathcal{E}}(\,(n.i)_2\,)| &\leq |\,\hat{\mathrm{S}}(\widehat{\mathrm{dbl}}(\mathrm{T}_{\mathcal{E}}(n)))| \;\leq\; |\,\widehat{\mathrm{dbl}}(\mathrm{T}_{\mathcal{E}}(n))| + 4 \\ &\leq |\mathrm{T}_{\mathcal{E}}(n)| + 8 \cdot \mathrm{lh}(\mathrm{T}_{\mathcal{E}}(n)) + 4 \;\leq\; |\mathrm{T}_{\mathcal{E}}(n)| + 8 \cdot |n| + 4 \\ &\overset{i.h.}{\leq} 8 \cdot |n|^2 + 8 \cdot |n| + 4 \;\leq\; 8 \cdot (|n| + 1)^2 \;=\; 8 \cdot |\,(n.i)_2\,|^2. \end{aligned}$$

□

Altogether we have seen that the predicates $\mathcal{E}, \prec$ and the functions $\hat{+}$, $\widehat{\mathrm{dbl}}$, $\widehat{\exp}$ and $\mathrm{T}_{\mathcal{E}}$ are polytime.

We close this section by proving that the predecessor function on the exponential notations

$$\hat{\mathrm{P}}(\alpha) \;:=\; \begin{cases} \hat{0} \;:\; \alpha = \hat{0} \\ \beta \;:\; \text{for that } \beta \text{ with } \beta \hat{+} \hat{\mathrm{S}}(\hat{0}) = \alpha \end{cases}$$

is not a polytime function.

**Proposition 2.4** $\hat{\mathrm{P}}$ *is not polynomially bounded.*

**Proof**: Obviously $|\widehat{\exp}(\mathrm{T}_{\mathcal{E}}(n))| > 1$ for $n > 0$, hence

$$|\hat{\mathrm{P}}(\widehat{\exp}(\mathrm{T}_{\mathcal{E}}(n)))| \;=\; |\mathrm{T}_{\mathcal{E}}(2^n - 1)| \;=\; |\mathrm{T}_{\mathcal{E}}(2^{n-1} + \ldots + 2^0)| \;\geq\; 2 \cdot n \geq 2^{|n|}.$$

Thus $\hat{\mathrm{P}}$ cannot be polynomially bounded, because $\widehat{\exp}$ and $\mathrm{T}_{\mathcal{E}}$ are so as polytime functions. $\square$

## 3 A polytime valuation function

In the previous section we defined interpretations $\hat{f}$ for $f \in \{0, \mathrm{S}, +, \mathrm{dbl}, \exp\}$ as functions on exponential notations. This can be extended inductively in the obviously way to arbitrary terms in the language $\{0, \mathrm{S}, +, \mathrm{dbl}, \exp\}$:

$$t = ft_1 \ldots t_k \quad \Rightarrow \quad \hat{t} = \hat{f}\hat{t}_1 \ldots \hat{t}_k.$$

In the following we identify formal terms and their GÖDELizations. Thus $\lambda t.\hat{t}$ can be seen as a function going from GÖDEL numbers of terms onto exponential notations. We will show that $\lambda t.\hat{t}$ is a polytime function. As a corollary we obtain that feasible arithmetic, i.e. BUSS' system $S^1_2$ of bounded arithmetic, can prove the consistency of the equational theory for the language $\{0, \mathrm{S}, +, \mathrm{dbl}, \exp\}$ involving only equations between closed terms, which is axiomatized by the recursive definitions of the function symbols.

We assume the same kind of unflat coding of syntax as in the definition of the exponential notations. With $\mathrm{tdp}(t)$ we indicate the term depth of $t$, which is inductively given for $t = ft_1 \ldots t_k$ by $\mathrm{tdp}(t) = 0$ if $k = 0$, and $\mathrm{tdp}(t) = 1 + \max_i \mathrm{tdp}(t_i)$ otherwise.

**Lemma 3.1** $\mathrm{tdp}(t) \leq ||t||$.

**Proof**: Let $t = ft_1 \ldots t_k$. If $k = 0$ the assertion is obvious. Otherwise $|t| = |\langle f, t_1, \ldots, t_k\rangle| \geq 2 \cdot \max_i |t_i|$ as $|\langle a_1, \ldots, a_k\rangle| \geq 2 \cdot |a_j|$ for all $j = 1, \ldots, k$.

Hence $||t|| \geq 1 + \max_i ||t_i|| \overset{i.h.}{\geq} 1 + \max_i \operatorname{tdp}(t_i) = \operatorname{tdp}(t)$. □

In the previous section we have computed

$$\begin{array}{rcl@{\qquad\qquad}rcl} |\hat{0}| &=& 0 & |\,\hat{\mathrm{S}}(\alpha)| &\leq& |\alpha| + 4 \\ |\,\widehat{\mathrm{dbl}}(\alpha)| &\leq& 9 \cdot |\alpha| & |\alpha \mathbin{\hat{+}} \beta| &\leq& |\alpha| + 9 \cdot |\beta| \\ |\,\widehat{\exp}(\alpha)| &\leq& 2 \cdot |\alpha| & & & \end{array}$$

hence $|\hat{f}(\vec{\alpha})| \leq c \cdot \max^1 |\vec{\alpha}|$ with $c = 10$ and $\max^1(\vec{a}) = \max\{\vec{a}, 1\}$.

**Lemma 3.2** $|\hat{t}| \leq c^{\operatorname{tdp}(t)}$.

**Proof**: Let $t = f t_1 \dots t_k$. If $k = 0$ the assertion is obvious. Otherwise

$$|\hat{t}| \;\leq\; c \cdot \max_i{}^1 |\hat{t}_i| \;\overset{i.h.}{\leq}\; c \cdot \max_i{}^1 c^{\operatorname{tdp}(t_i)} \;\leq\; c^{1+\max_i \operatorname{tdp}(t_i)} \;=\; c^{\operatorname{tdp}(t)}.$$

□

Using the last two Lemmas we obtain (for $t > 1$)

$$|\hat{t}| \;\leq\; c^{\operatorname{tdp}(t)} \;\leq\; c^{||t||} \;\leq\; |t|^{2 \cdot |c|} \;<\; |t|^{2^{1+||c||}} \;\leq\; |T_{1+||c||}(t)|$$

where $T_0(x) = x$, $T_{c+1}(x) = T_c(x) \mathbin{\#} T_c(x)$, thus $|T_c(x)| \geq |x|^{2^c}$. As $||10|| = 3$ we have $\hat{t} < T_4(t)$. Thus $\lambda t.\hat{t}$ is polytime by course-of-value recursion (Theorem 1.2) using the course function tc from the previous section.

The canonical equational theory $EqT$ for $\{0, \mathrm{S}, +, \mathrm{dbl}, \exp\}$ consists of equations between closed terms, which are inductively defined by instances of the recursive definition of +, dbl, exp given by

$$\begin{array}{c@{\qquad\qquad}c} x + 0 = x & x + \mathrm{S}\, y = \mathrm{S}(x + y) \\ \mathrm{dbl}(0) = 0 & \mathrm{dbl}(\mathrm{S}\, x) = \mathrm{S}\,\mathrm{S}\,\mathrm{dbl}(x) \\ \exp(0) = \mathrm{S}\, 0 & \exp(\mathrm{S}\, x) = \mathrm{dbl}(\exp(x)) \end{array}$$

the definition of equality as an equivalence relation, and the compatibility of equality with the function symbols.

Given an $EqT$ proof $P$ of some (closed) equation $s = t$ we will show by induction on the length of the proof $P$ that the interpretation ˆ makes every equation in $P$ valid, hence $\hat{s} = \hat{t}$. Hence $EqT$ is consistent in the sense that $0 = \mathrm{S}\, 0$ is not derivable in $EqT$, as $\hat{0} \neq \hat{\mathrm{S}}\, \hat{0}$.

If an arbitrary equational theory $EqT'$ for our language for exponentiation is true in the standard model then all instances of axioms from $EqT'$ remain valid under the interpretation $\hat{}$. But we need in addition that this can also be seen inside $S^1_2$. We will sketch below that this is indeed the case for our canonical axiomatization $EqT$ – any non-pathological axiomatization would have this property, too, but would require a new proof inside $S^1_2$. From this the induction step for proving the validity under the interpretation $\hat{}$ is obviously as also the "real" equality on the exponential notations (inside $S^1_2$) is an equivalence relation and compatible with the function symbols. This proof formalizes in $S^1_2$, because $t \mapsto \hat{t}$ is polynomially bounded.

We now sketch how to prove the validity of the axioms of $EqT$ inside $S^1_2$. We start by stating some properties. Let $\hat{n}$ be the exponential notation $\mathrm{T}_{\mathcal{E}}(n)$ for the natural number $n$, hence $\hat{\mathrm{S}}\,\hat{0} = \check{2}^{\hat{0}} = \hat{1}$ is provable in $S^1_2$.

**Proposition 3.3** ($S^1_2$)

*(1)* $\prec$ *fulfills trichotomy.*
*(2)* $\gamma \mathbin{\hat{+}} \hat{1} = \hat{\mathrm{S}}\,\gamma$.
*(3)* $(\gamma \mathbin{\hat{+}} \check{2}^{\hat{n}}) \mathbin{\hat{+}} \check{2}^{\hat{n}} = \gamma \mathbin{\hat{+}} \check{2}^{\widehat{n+1}}$
*(4)* $\hat{\mathrm{S}}(\gamma \mathbin{\hat{+}} (\check{2}^{\widehat{n-1}} \mathbin{\check{+}} \ldots \mathbin{\check{+}} \check{2}^{\hat{0}})) = \gamma \mathbin{\hat{+}} \check{2}^{\hat{n}}$ □

Now it is easy to show the validity of the addition axioms.

**Lemma 3.4** ($S^1_2$) $\alpha \mathbin{\hat{+}} \hat{0} = \alpha$ *and* $\alpha \mathbin{\hat{+}} \hat{\mathrm{S}}\,\beta = \hat{\mathrm{S}}(\alpha \mathbin{\hat{+}} \beta)$.

**Proof**: The first equality is immediately from the definition of $\hat{+}$ and $\hat{0}$. For the second one let $\beta$ be of the form $\check{2}^{\beta_1} \mathbin{\check{+}} \ldots \mathbin{\check{+}} \check{2}^{\beta_k}$.

If $k = 0$ or $\beta_k \neq \hat{0}$ then $\hat{\mathrm{S}}\,\beta = \check{2}^{\beta_1} \mathbin{\check{+}} \ldots \mathbin{\check{+}} \check{2}^{\beta_k} \mathbin{\check{+}} \check{2}^{\hat{0}}$ and

$$\alpha \mathbin{\hat{+}} \hat{\mathrm{S}}\,\beta \overset{(*)}{=} (\alpha \mathbin{\hat{+}} \beta) \mathbin{\hat{+}} \hat{1} \overset{3.3.2}{=} \hat{\mathrm{S}}(\alpha \mathbin{\hat{+}} \beta)$$

where $(*)$ simply uses the definition of $\hat{+}$.

Otherwise $\beta_k = \hat{0}$. Let $i := F(\beta)$, then $\beta$ has the form

$$\check{2}^{\beta_1} \mathbin{\check{+}} \ldots \mathbin{\check{+}} \check{2}^{\beta_{i-1}} \mathbin{\check{+}} \check{2}^{\hat{n}} \mathbin{\check{+}} \ldots \mathbin{\check{+}} \check{2}^{\hat{0}}$$

for $n := k - i$, and $\widehat{n+1} = \hat{\mathrm{S}}\,\hat{n} \prec \beta_{i-1}$ if $i > 1$. Thus

$$\hat{\mathrm{S}}\,\beta = \check{2}^{\beta_1} \mathbin{\check{+}} \ldots \mathbin{\check{+}} \check{2}^{\beta_{i-1}} \mathbin{\check{+}} \check{2}^{\widehat{n+1}}.$$

On the other hand we compute

$$\begin{aligned}\hat{\mathrm{S}}(\alpha \mathbin{\hat{+}} \beta) &\overset{(*)}{=} \hat{\mathrm{S}}\left((\alpha \mathbin{\hat{+}}(\check{2}^{\beta_1} \mathbin{\check{+}} \ldots \mathbin{\check{+}} \check{2}^{\beta_{i-1}})) \mathbin{\hat{+}}(\check{2}^{\hat{n}} \mathbin{\check{+}} \ldots \mathbin{\check{+}} \check{2}^{\hat{0}})\right) \\ &\overset{3.3.4}{=} (\alpha \mathbin{\hat{+}}(\check{2}^{\beta_1} \mathbin{\check{+}} \ldots \mathbin{\check{+}} \check{2}^{\beta_{i-1}})) \mathbin{\hat{+}} \check{2}^{\widehat{n+1}} \\ &\overset{(*)}{=} \alpha \mathbin{\hat{+}}(\check{2}^{\beta_1} \mathbin{\check{+}} \ldots \mathbin{\check{+}} \check{2}^{\beta_{i-1}} \mathbin{\check{+}} \check{2}^{\widehat{n+1}}) = \alpha \mathbin{\hat{+}} \hat{\mathrm{S}}\,\beta.\end{aligned}$$

where $(*)$ again uses the definition of $\hat{+}$. □

The validity of dbl and exp under the interpretation $\hat{}$ is more or less straightforward.

**Lemma 3.5** $(S^1_2)$ *$\widehat{\mathrm{dbl}}(\hat{0}) = \hat{0}$ and $\widehat{\mathrm{dbl}}(\hat{\mathrm{S}}\,\alpha) = \hat{\mathrm{S}}\,\hat{\mathrm{S}}\,\widehat{\mathrm{dbl}}(\alpha)$.*

**Proof**: The first equation is again immediately from the definitions. For the second one let $\alpha$ be of the form $\check{2}^{\alpha_1} \mathbin{\check{+}} \ldots \mathbin{\check{+}} \check{2}^{\alpha_k}$.

If $k = 0$ or $\alpha_k \neq \hat{0}$ then $\hat{\mathrm{S}}\,\alpha = \check{2}^{\alpha_1} \mathbin{\check{+}} \ldots \mathbin{\check{+}} \check{2}^{\alpha_k} \mathbin{\check{+}} \check{2}^{\hat{0}}$, hence

$$\widehat{\mathrm{dbl}}(\hat{\mathrm{S}}\,\alpha) = \check{2}^{\hat{\mathrm{S}}\,\alpha_1} \mathbin{\check{+}} \ldots \mathbin{\check{+}} \check{2}^{\hat{\mathrm{S}}\,\alpha_k} \mathbin{\check{+}} \check{2}^{\hat{1}}.$$

On the other hand $\widehat{\mathrm{dbl}}(\alpha) = \check{2}^{\hat{\mathrm{S}}\,\alpha_1} \mathbin{\check{+}} \ldots \mathbin{\check{+}} \check{2}^{\hat{\mathrm{S}}\,\alpha_k}$ with $\hat{1} \prec \hat{\mathrm{S}}\,\alpha_k$ if $k > 0$, hence

$$\hat{\mathrm{S}}\,\hat{\mathrm{S}}\,\widehat{\mathrm{dbl}}(\alpha) = \hat{\mathrm{S}}(\check{2}^{\hat{\mathrm{S}}\,\alpha_1} \mathbin{\check{+}} \ldots \mathbin{\check{+}} \check{2}^{\hat{\mathrm{S}}\,\alpha_k} \mathbin{\check{+}} \check{2}^{\hat{0}}) = \widehat{\mathrm{dbl}}(\hat{\mathrm{S}}\,\alpha).$$

Otherwise $\alpha_k = \hat{0}$. Let $i$ be $F(\alpha)$, hence $\hat{\mathrm{S}}\,\alpha = \check{2}^{\alpha_1} \mathbin{\check{+}} \ldots \mathbin{\check{+}} \check{2}^{\alpha_{i-1}} \mathbin{\check{+}} \check{2}^{\hat{\mathrm{S}}\,\alpha_i}$. Now we compute $\widehat{\mathrm{dbl}}(\hat{\mathrm{S}}\,\alpha) = \check{2}^{\hat{\mathrm{S}}\,\alpha_1} \mathbin{\check{+}} \ldots \mathbin{\check{+}} \check{2}^{\hat{\mathrm{S}}\,\alpha_{i-1}} \mathbin{\check{+}} \check{2}^{\hat{\mathrm{S}}\,\hat{\mathrm{S}}\,\alpha_i}$. On the other hand $\mathrm{dbl}(\alpha) = \check{2}^{\hat{\mathrm{S}}\,\alpha_1} \mathbin{\check{+}} \ldots \mathbin{\check{+}} \check{2}^{\hat{\mathrm{S}}\,\alpha_{i-1}} \mathbin{\check{+}} \ldots \mathbin{\check{+}} \check{2}^{\hat{\mathrm{S}}\,\hat{0}}$, hence

$$\hat{\mathrm{S}}\,\widehat{\mathrm{dbl}}(\alpha) = \check{2}^{\hat{\mathrm{S}}\,\alpha_1} \mathbin{\check{+}} \ldots \mathbin{\check{+}} \check{2}^{\hat{\mathrm{S}}\,\alpha_{i-1}} \mathbin{\check{+}} \check{2}^{\hat{\mathrm{S}}\,\alpha_i} \mathbin{\check{+}} \ldots \mathbin{\check{+}} \check{2}^{\hat{\mathrm{S}}\,\hat{0}} \mathbin{\check{+}} \check{2}^{\hat{0}}$$

and $F(\hat{\mathrm{S}}\,\widehat{\mathrm{dbl}}(\alpha)) = i$. Thus $\hat{\mathrm{S}}\,\hat{\mathrm{S}}\,\widehat{\mathrm{dbl}}(\alpha) = \check{2}^{\hat{\mathrm{S}}\,\alpha_1} \mathbin{\check{+}} \ldots \mathbin{\check{+}} \check{2}^{\hat{\mathrm{S}}\,\alpha_{i-1}} \mathbin{\check{+}} \check{2}^{\hat{\mathrm{S}}\,\hat{\mathrm{S}}\,\alpha_i} = \widehat{\mathrm{dbl}}(\hat{\mathrm{S}}\,\alpha)$.
□

**Lemma 3.6** $(S^1_2)$ *$\widehat{\exp}(\hat{0}) = \hat{1}$ and $\widehat{\exp}(\hat{\mathrm{S}}\,\alpha) = \widehat{\mathrm{dbl}}(\widehat{\exp}(\alpha))$.*

**Proof**: Let us remind $\widehat{\exp}(\alpha) = \langle\alpha\rangle = \check{2}^{\alpha}$. We compute $\widehat{\exp}(\hat{0}) = \check{2}^{\hat{0}} = \hat{1}$ and $\widehat{\exp}(\hat{\mathrm{S}}\,\alpha) = \check{2}^{\hat{\mathrm{S}}\,\alpha} = \widehat{\mathrm{dbl}}(\check{2}^{\alpha}) = \widehat{\mathrm{dbl}}(\widehat{\exp}(\alpha))$. □

Altogether we have shown the following

**Corollary 3.7** *$S^1_2$ proves the consistency of the canonical equational theory EqT for the language* $\{0, \mathrm{S}, +, \mathrm{dbl}, \exp\}$. □

## References

[1] Arnold Beckmann, *Separating fragments of bounded arithmetic*, PhD thesis, WWU Münster, 1996.

[2] Sam Buss, *Bounded arithmetic*, volume 3 of *Studies in Proof Theory, Lecture Notes*, Bibliopolis, 1986.

[3] Peter Clote and Jan Krajíček, Open problems. In: Peter Clote and Jan Krajíček, editors, *Arithmetic, proof theory, and computational complexity*, pages 1–9, Papers from the conference held in Prague, July 2–5, 1991, Oxford Logic Guides 23, New York, 1993.

[4] A. Cobham, The intrinsic computational difficulty of functions. In: Yehoshua Bar-Hillel, editor, *Logic, Methodology and Philosophy of Science*, pages 24–30, Amsterdam, North-Holland Publishing Company, 1965.

[5] Petr Hajek and Pavel Pudlák, *Metamathematics of First-Order Arithmetic.* Perspectives in Mathematical Logic, Springer-Verlag, 1993.

[6] Jan Krajíček, *Bounded Arithmetic, Propositional Logic, and Complexity Theory.* Cambridge University Press, Heidelberg/New York, 1995.

[7] Wolfram Pohlers, *Proof Theory. An Introduction.* Number 1407 in Lecture Notes in Mathematics. Springer-Verlag, Berlin/Heidelberg/New York, 1989.

[8] Harvey E. Rose, *Subrecursion. Functions and hierarchies*, Oxford Logic Guides, Oxford, 1984.

[9] Gaisi Takeuti, Sharply bounded arithmetic and the function $a - 1$. In: Wilfried Sieg, editor, *Logic and Computation*, pages 281–288, number 106 in Contemporary Mathematics, Providence, American Mathematical Society, 1990.