Bounded Arithmetic S_2

Recall that our ultimate goal is to come up with a theory T so that given a relation $P(x, y)$ for which $T \vdash \forall x \exists y P(x, y)$ one can find an efficient algorithm which on input x computes y so that $P(x, y)$.

The first reasonable candidate theory was $I\Delta_0$ with Δ_0-definable relation $P(x, y)$, since by Parikh's theorem

$$I\Delta_0 \vdash \forall x \exists y P(x, y) \implies I\Delta_0 \vdash \forall x \exists y \leq t(x) P(x, y),$$

with $t(x)$ - an arithmetical term, i.e. a polynomial.

As we have discussed, the algorithm for computing y from x just loops through all numbers in increasing order until one is found satisfying $P(x, y)$ and $t(x)$ serves as an upper bound on the length of the cycle.

Of course, since x is represented as a string of length $\approx \log(x)$ and there are $t(x)$ numbers to check, the total runtime of the algorithm is $\geq t(x) \approx 2^{C \log(x)}$ for a suitable constant C, since $t(x)$ is a polynomial.

Notice that, even if one comes up with a substantial reduction of the search space, in principle even verifying whether $P(x, y)$ holds for a Δ_0-definable relation $P(x, y)$ still takes time exponential in lengths of x, y.

Can we do better?

Exercise 1. * Show that, if one is allowed to use nested non-determinism, the task of computing y given x as above can be done in time linear in the length of x.

Formally, the above task can be solved by an algorithm from the functional analog of the *linear-time hierarchy*.

Fact 2. Any total relation $P(x, y)$ computable by a function from the linear-time hierarchy is definable by a suitable Δ_0-formula $\varphi(x, y)$. Moreover, for any such $P(x, y)$ and $\varphi(x, y)$ $I\Delta_0$ proves $\forall x \exists y \varphi(x, y)$.

In particular, $I\Delta_0$ cannot be the theory we seek, unless we first derive new groundbreaking results. For the exact same reason, we cannot show that $I\Delta_0$ does not give the witnessing we want.

Before we proceed further, we first need to inspect our main question once again.

Recall that we have started with PA, which is obviously too strong for any reasonable witnessing, and then moved to a weaker theory $I\Delta_0$ which again turned too strong. A natural choice is to move to an even weaker theory. Notice, however, that all this time we actually had the weakest theory of them all right before us - the *empty theory* E.

If one believes that some efficient witnessing is achievable and is dependent on the strength of T, then there is no better choice than to let T be E. One then needs to inspect what happens if $E \vdash \forall x \exists y P(x, y)$ and whether we do get an algorithm of any kind.

As we will later see, there is a fundamental witnessing result for E which is also known as Herbdrand's theorem. We will prove it later. At this point, however, we need to refine the original question we started with.

It is true that the weaker the theory, the stronger the proofs, and proofs in E are as strong as possible. However, such theories are actually too weak, in the sense that for any reasonable $P(x, y)$ E would probably not be able to prove $\forall x \exists y P(x, y)$. Witnessing results, which are of course interesting on their own, are primarily used as a tool for further investigations in the complexity theory. And so we want our T to be weak to derive efficient witnessing, but at the same time, we want it to be strong enough to actually prove $\forall x \exists y P(x, y)$ for non-trivial $P(x, y)$.

And so we refine our original question and get the following.

Question 3. Is there a theory T so that for a total relation $P(x, y)$ there is an efficient algorithm computing y from the given x satisfying $P(x, y)$ if and only if $P(x, y)$ is definable by a formula $\varphi(x, y)$ so that $T \vdash \forall x \exists y \varphi(x, y)$?

We have already seen that $I\Delta_0$ can give only (nested) non-deterministic linear-time witnessing and proving whether or not it is equivalent to efficient (i.e. poly-time deterministic) witnessing is beyond the current state of knowledge.

However, we can show that $I\Delta_0$ fails to be the T we seek by investigating the other implication of 3.

Exercise 4. In the following, we assume $P(x, y)$ is already represented as a suitable Δ_0-formula. Show that for all of the below relations $I\Delta_0 \nvdash \forall x \exists y P(x, y)$. Furthermore, argue that there is an efficient algorithm which on input x outputs y satisfying $P(x, y)$.

- x represents two binary n-dimensional vectors v, w and y represents their *outer product* $v * w$, i.e. a matrix of shape (n, n) such that $(v * w)_{i,j} = v_i w_j$.
- x represent a formula $\psi(x)$ together with a term t and y represent $\psi(t)$.
- For a fixed efficient super-linear-time algorithm A, x represents an input to A and y represents finished accepting/rejecting computation of A on x.
- * For a fixed *secure pseudorandom number generator* G with super-linear stretch, x represents an input to G and y represents the output of G on x.

And so it is not enough just to weaken $I\Delta_0$, we first need to bypass the above problems. This will be done by expanding the language L_{PA} and adding suitable axioms for newly introduced symbols.

Recall L_{PA} is the language $0, 1, +, \cdot, <$ and PA^- is the theory in L_{PA} axiomatizing positive parts of discreetly-ordered rings. The axioms are as follows.

PA^-

- $\forall x, y, z\, ((x + y) + z = x + (y + z))$
- $\forall x, y\, (x + y) = (y + x)$
- $\forall x, y, z\, ((x \cdot y) \cdot z = x \cdot (y \cdot z))$
- $\forall x, y\, (x \cdot y) = (y \cdot x)$
- $\forall x, y, z\, (x \cdot (y + z)) = x \cdot y + x \cdot z$
- $\forall x\, ((x + 0 = x) \wedge (x \cdot 0 = 0))$
- $\forall x\, (x \cdot 1 = x)$
- $\forall x, y, z\, ((x < y \wedge y < z) \to x < z)$
- $\forall x\, \neg x < x$
- $\forall x, y\, (x < y \vee x = y \vee y < x)$
- $\forall x, y, z\, (x < y \to x + z < y + z)$
- $\forall x, y, z\, (0 < z \wedge x < y \to x \cdot z < y \cdot z)$
- $\forall x, y\, (x < y \to \exists z\, x + z = y)$
- $0 < 1 \wedge \forall x\, (x > 0 \to x \geq 1)$
- $\forall x\, (x \geq 0)$

Below $\mathbb{N}$ is the standard model interpreting L_{PA} symbols in the usual way. Of course, $\mathbb{N}$ models PA^-.

As a first step we expand L_{PA} by an additional unary function symbol $\lfloor \frac{x}{2} \rfloor$ together with the axiom

- $\forall x, y\, (x = \lfloor \frac{y}{2} \rfloor \leftrightarrow (2 \cdot x = y \vee 2 \cdot x + 1 = y))$

Exercise 5. Show that there is a unique interpretation of $\lfloor \frac{x}{2} \rfloor$ in $\mathbb{N}$ satisfying the above axiom.

From now on $\mathbb{N}$ is assumed to interpret $\lfloor \frac{x}{2} \rfloor$, as well.

As a second step, we add a unary function symbol $|x|$ together with the following axioms

- $|0| = 0$
- $|1| = 1$
- $\forall x, y\, (x < y \to |x| < |y|)$
- $\forall x\, (x \neq 0 \to (|2 \cdot x| = |x| + 1 \wedge |2 \cdot x + 1| = |x| + 1))$

- $\forall x\,(x \neq 0 \rightarrow |x| = ||\lfloor \frac{x}{2} \rfloor| + 1)$

Exercise 6. Show that there is a unique interpretation of $|x|$ in $\mathbb{N}$ satisfying the above axioms.

From now on $\mathbb{N}$ is assumed to interpret $|x|$, as well.

Finally, we add a binary function symbol $x\#y$ with the following axioms

- $\forall x\,(0\#x = 1)$
- $\forall x, y\,(x\#y = y\#x)$
- $\forall x\,(1\#(2 \cdot x) = 2 \cdot (1\#x) \wedge 1\#(2 \cdot x + 1) = 2 \cdot (1\#x))$
- $\forall x, y\,(|x\#y| = |x| \cdot |y| + 1)$
- $\forall x, y, z\,(|x| = |y| \rightarrow x\#z = y\#z)$
- $\forall x, y, z, w\,(|x| = |y| + |z| \rightarrow x\#w = (y\#w) \cdot (z\#w))$

Exercise 7. Show that there is a unique interpretation of $x\#y$ in $\mathbb{N}$ satisfying the above axioms.

The motivation behind $x\#y$ is the following simple but very important observation.

Exercise 8. Let x, y be numbers representing binary strings in the standard way. Then, the bit-length of y is poly-size bounded in the bit-length of x if and only if y as a number is bounded by a term resulting from applying $\#$ to x iteratively.

Concretely

$$|y| < |x|^c \iff y < x\# \cdots \#x$$

with c a fixed constant and $\#$ applied exactly c-times.

From now on $\mathbb{N}$ is assumed to interpret $x\#y$, as well.

Remark 9. * It is possible to solve Exercises 5 and 6 with $\mathbb{N}$ being replaced by an arbitrary $I\Delta_0$ model $\mathbb{M}$.

Exercise 7 is a bit tricky. First of all one needs to be sure that the operation $x\#y$ is even definable by a Δ_0-formula. This is true, although not trivial, i.e. there is a Δ_0-formula $\varphi(x, y, z)$ so that in $\mathbb{N}$ $\forall x, y, z\,(x\#y = z \leftrightarrow \varphi(x, y, z))$.

By choosing $\varphi(x, y, z)$ well enough, one can show that $I\Delta_0$ does indeed prove the uniqueness of the interpretation of $x\#y$.

However, $I\Delta_0$ is not able to prove $\forall x, y \exists z \varphi(x, y, z)$ and so there exist models of $I\Delta_0$ where $x\#y$ can only be interpreted as a *partial* operation.

The language L_{PA} with newly introduced symbols is denoted as L_{S_2} and the corresponding theory is called $BASIC$.

The notion of a bounded L_{S_2}-formula is defined in the same way as before and so we can overload Δ_0. Finally, the overloaded $I\Delta_0$ is denoted as S_2.

Remark 10. * The number 2 in S_2 indicates the presence of # in the language. The theory without such a symbol is called S_1, while at the same time, it is possible to iteratively define $\#_k$ symbols (the usual # here is $\#_2$). Such operations are all super-polynomial (quasi-polynomial and faster) but are still not as fast as the exponential function.

Fact 11. Theorem of Parikh still applies in the current context, i.e. for any Δ_0-formula $\varphi(x, y)$

$$S_2 \vdash \forall x \exists y \varphi(x, y) \implies S_2 \vdash \forall x \exists y \leq t(x) \varphi(x, y),$$

with $t(x)$ - an L_{S_2}-term, i.e. a quasi-polynomial.

Exercise 12. What kind of deterministic/non-deterministic witnessing do we get for the theory S_2 and Δ_0-definable total relation $P(x, y)$? Compare it to the witnessing for $I\Delta_0$.