Memo : Undecidability of Validity for Peano Arithmetic

1 Peano Arithmetic

We treat Peano Arithmetic (PA) in first-order logic. Here is a very short recap of the language of PA. It contains the following symbols:

$$\begin{array}{rl}
\textbf{function symbols :} & 0 \quad S \quad + \quad \cdot \\
\textbf{predicate symbols :} & \equiv \\
\textbf{logical symbols :} & \bot \quad \wedge \quad \vee \quad \rightarrow \\
\textbf{quantifiers :} & \forall \quad \exists
\end{array}$$

Next, we inductively define **terms** $\mathbb{T}$ and **formulas** $\mathbb{F}$.

Definition 1.1: t is of type $\mathbb{T}$ iff

- $t = v_n$ for some number $n \in \mathbb{N}$.
- $t = f t_1 \dots t_n$ for $t_1, \dots, t_n : \mathbb{T}$ and a function symbol f of arity n.

Definition 1.2: ϕ is of type $\mathbb{F}$ iff

- $\phi = \bot$
- $\phi = t_1 \equiv t_2$ for $t_1, t_2 : \mathbb{T}$
- $\phi = L\phi_1\phi_2$ for $\phi_1, \phi_2 : \mathbb{F}$ and a logical symbol L.
- $\phi = Q.\ \varphi$ for $\varphi : \mathbb{F}$ and a quantifier Q.

Remark 1.3: The binding of variables v_n by quantifiers is realised by using de Bruijn indices [1]. As can be seen in the last rule for building a formula $Q.\ \varphi$, we do not specify the variable that Q binds. Instead, a variable v_n is bound by Q if Q is the n-th quantifier shadowing v_n. This approach to binding greatly simplifies formalisation.

To ease reading, we will however also use the customary notation for binding variables, by adding the variable name behind the quantifier. So in this text, it will the usage of de Bruijn indices will not be noticeable.

In the following, we will use the usual brackets plus infix notations for 2-ary function and logical symbols. We will also use variable names x, y, z.

1.1 Axioms

Here is a list of formulas we will take as axioms.

- **Disjointness** : $\forall x.\ 0 \not\equiv Sx$
- **Injectivity of** S : $\forall xy.\ Sx \equiv Sy \ \rightarrow\ x \equiv y$
- **Zero addition** : $\forall x.\ 0 + x \equiv 0$
- **Recursion for addition** : $\forall xy.\ (Sx) + y \equiv S(x + y)$
- **Zero multiplication** : $\forall x.\ 0 \cdot x \equiv 0$
- **Recursion for multiplication** : $\forall x.\ (Sx) \cdot y \equiv y + x \cdot y$

We will call the list containing these formulas α_{PA}. Additionally, for any formula $\phi(x)$ with free variable x we also add the induction axiom

$$\phi(0) \ \rightarrow\ (\forall x.\ \phi(x) \ \rightarrow\ \phi(Sx)) \ \rightarrow\ \forall x.\ \phi(x)$$

Remark 1.4: Note that, because of the induction axioms, there are infinitely many axioms. For any formula however, it is possible to decide whether it is an axiom or not. This also makes it possible to enumerate all axioms.

2 Tarski Semantics

[5] A (Tarski) model $\mathcal{M}$ for a first order language like PA is given by fixing a domain D and an interpretation function I, which assigns to any k-ary function symbol f a function $f^I : D^k \rightarrow D$ and to any k-ary predicate symbol P a k-ary predicate P^I on D. For any **assignment** $\rho : \mathbb{N} \rightarrow D$ we can then recursively define a **term evaluation** $\hat{\rho} : \mathbb{T} \rightarrow D$ by

$$\hat{\rho}(v_i) := \rho(i) \quad \text{and} \quad \hat{\rho}(f v_1 \dots v_n) := f^I \hat{\rho}(v_1) \dots \hat{\rho}(v_n).$$

Let $a : D$ and ρ be an assignment, then by $a;\rho$ we will designate the function that starts with a and follows up by ρ:

$$a;\rho(0) := a \quad , \quad a;\rho(k+1) := \rho(k)$$

With this, we extend term evaluations $\hat{\rho}$ to formula evaluations $\mathcal{M} \vDash_\rho \phi$ by

- $\mathcal{M} \vDash_\rho \bot \ :=\ \bot$
- $\mathcal{M} \vDash_\rho P t_1 \dots t_n \ :=\ P^I \hat{\rho}(t_1) \dots \hat{\rho}(t_n)$
- $\mathcal{M} \vDash_\rho L \phi_1 \dots \phi_n \ :=\ \hat{L}(\mathcal{M} \vDash_\rho \phi_1) \dots (\mathcal{M} \vDash_\rho \phi_n)$
- $\mathcal{M} \vDash_\rho Q\ \varphi \ :=\ \hat{Q}\ d : D.\ \mathcal{M} \vDash_{d;\rho} \varphi$

where we note that $\hat{L}, \hat{Q}$ are the interpretations of the logical symbols and quantifiers in our meta-logic. If Γ is a list of formulas, we will write $\mathcal{M} \vDash_\rho \Gamma$ iff $\mathcal{M} \vDash_\rho \phi$ for every formula ϕ in Γ and $\mathcal{M} \vDash \Gamma$ if we have $\mathcal{M} \vDash_\rho \Gamma$ for every ρ.

We then call a list of formulas Γ **valid** iff for every model $\mathcal{M}$, we have $\mathcal{M} \vDash \Gamma$.

Remark 2.1: In the following, the only predicate symbol we have will be the equality symbol $\equiv$ for PA. To ease discussion, we will further restrict our attention to extensional models, i.e. models in which the interpretation $\equiv^I$ of equality is syntactic equality $=$. Any model we consider from now on will be assumed to be extensional.

2.1 PA models

We want to work with models of PA, i.e. models in which the axioms of PA hold.

Definition 2.2: A model $\mathcal{M}$ is a **PA model** iff $\mathcal{M} \vDash \alpha_{\mathsf{PA}}$ and $\mathcal{M} \vDash \alpha$ for any induction axiom α. We will also write $\mathcal{M} \vDash \mathsf{PA}$ for this.

In this context, we also adopt the following meaning for validity:

$$\mathsf{PA} \vDash \phi \ := \ \forall \mathcal{M}.\ \mathcal{M} \vDash \mathsf{PA} \to \mathcal{M} \vDash \phi$$

It expresses that some formula holds in every model in which the axioms of PA hold. Our goal will be to show that it is (in some sense we will make more precise) not possible to have a general way of deciding this for every formula.

3 Diophantine Equations

Informally, a `h10c` **problem** is a list of **atomic equations**. The latter contain variables $x_1, x_2, x_3, \ldots$ and have one of the following three forms:

$$x_i = 1 \quad | \quad x_i + x_j = x_k \quad | \quad x_i \cdot x_j = x_k$$

Given a function $\sigma : \mathbb{N} \to \mathbb{N}$ we define **evaluations** of atomic equations by

- $[x_i = 1]_\sigma :=\ \sigma(i) = 1$
- $[x_i + x_j = x_k]_\sigma :=\ \sigma(i) + \sigma(j) = \sigma(k)$
- $[x_i \cdot x_j = x_k]_\sigma :=\ \sigma(i) \cdot \sigma(j) = \sigma(k)$

If there exists a σ which makes the evaluation true, we say that the equation is **satisfiable**.

Example 3.1: The following is an example of a `h10c` problem

$$[\ x_1 = 1\ ,\ x_1 + x_1 = x_2\ ,\ x \cdot x = x_2\ ]$$

By substituting the equations into each other this yields the equation $x^2 = 2$, which is a more familiar Diophantine equation. It is now also obvious that the above problem cannot be satisfied, since there is no natural number x satisfying $x^2 = 2$.

Remark 3.2: We mention here (but will not show), that for every Diophantine equation it is possible to write down an equivalent `h10c` problem [2]. We choose to treat satisfiability of the latter, since the translation into an adequate PA sentence is easier.

For problems L, we now define the predicate

$$\mathsf{sat}_{\mathsf{h10c}}(L) \ := \ \exists\, \sigma : \mathbb{N} \to \mathbb{N}, \ \ \forall\, a \in L, \ [a]_\sigma$$

and call L **satisfiable** if the above holds, i.e. when there is some σ satisfying all its atomic equations simultaneously.

4 Undecidability

We say that some predicate P over a type X is **decidable** iff there is a function f from X to the Boolean type $\mathbb{B}$ with $fx = \texttt{true} \ \leftrightarrow \ Px$ and we call it **undecidable** when the negation holds.

Remark 4.1: Inside of the proof assistant Coq, nothing internally tells us that any function from some type to another type is in any sense computable. From the outside however we can argue that every function we can define in Coq is indeed a computable function.

This is why in our definition of *decidable* we simply ask for *any* function f on the level of Coq. Looked at from the outside we can recognise f as being computable, which is what we want for the concept of a decider.

It has been shown [3] [4] that Hilbert's 10th problem is undecidable and, more importantly in this context, $\mathsf{sat}_{\mathsf{h10c}}$ is undecidable [2].

We now want to use this result to show that $\mathsf{PA} \vDash$ is undecidable as well. To do this, we will establish a many-one reduction. This means we translate `h10c` problems to formulas by a function ε, such that

Theorem 4.2: $\forall\, L, \ \mathsf{sat}_{\mathsf{h10c}}\ L \ \leftrightarrow \ \mathsf{PA} \vDash \varepsilon(L)$

By this result then, undecidability of $\mathsf{PA} \vDash$ follows:

Proof: Assume it were decidable, then there is a function $f : \mathbb{F} \to \mathbb{B}$ establishing the respective equivalence. But with this, $(f \circ \varepsilon)$ would show decidability of $\mathsf{sat}_{\mathsf{h10c}}$:

$$f(\varepsilon(L)) = \texttt{true} \ \leftrightarrow \ \mathsf{PA} \vDash \varepsilon(L) \ \leftrightarrow \ \mathsf{sat}_{\mathsf{h10c}}(L) \qquad \square$$

5 Embedding of H10 into PA

The idea behind the embedding ε will be very straight forward, the essential part will be proving the wished for equivalence of Theorem 4.2.

We first define a function ${}_{\mathsf{pre}}\varepsilon$ translating atomic equations to the most canonical PA formula possible:

- ${}_{\mathsf{pre}}\varepsilon(x_i = 1) := \ (v_i \equiv S0)$
- ${}_{\mathsf{pre}}\varepsilon(x_i + x_j = x_k) := \ (v_i + v_j \equiv v_k)$
- ${}_{\mathsf{pre}}\varepsilon(x_i \cdot x_j = x_k) := \ (v_i \cdot v_j \equiv v_k)$

Remark 5.1: In the above, the choice of our notation almost makes it look like ${}_{\mathsf{pre}}\varepsilon$ does not do anything. Note however, that ${}_{\mathsf{pre}}\varepsilon$ is applied to elements of type `h10c` which then end up (on the right) as formulas in PA.

We extend the definition of ${}_{\mathsf{pre}}\varepsilon$ to problems, by recursively defining

- ${}_{\mathsf{pre}}\varepsilon[] := (0 \equiv 0)$
- ${}_{\mathsf{pre}}\varepsilon(a :: L) := {}_{\mathsf{pre}}\varepsilon(a) \wedge {}_{\mathsf{pre}}\varepsilon(L)$

Example 5.2: Calculating the embedding of the list from Example 3.1, we get

$$\begin{aligned} & {}_{\mathsf{pre}}\varepsilon\left[\ x_1 = 1\ ,\ x_1 + x_1 = x_2\ ,\ x_3 \cdot x_3 = x_2\ \right] \\ = \ & (v_1 \equiv S0)\ \wedge\ (v_1 + v_1 \equiv v_2)\ \wedge\ (v_3 \cdot v_3 \equiv v_2)\ \wedge\ (0 \equiv 0) \end{aligned}$$

This is a formula with free variables v_1, v_2, v_3. However, it does not yet express the satisfiability of some equations. We can fix this by adding existential quantifiers, to get

$$\exists\, v_1\, \exists\, v_2\, \exists\, v_3.\ (v_1 \equiv S0)\ \wedge\ (v_1 + v_1 \equiv v_2)\ \wedge\ (v_3 \cdot v_3 \equiv v_2)\ \wedge\ (0 \equiv 0)$$

which is now expressing what we want.

So learning from the above example we define

$$\varepsilon(L) := \exists\, v_1 \ldots \exists\, v_N.\ {}_{\mathsf{pre}}\varepsilon(L)$$

where N stands for the biggest index of a variable v_i appearing in ${}_{\mathsf{pre}}\varepsilon(L)$. In this way we make sure that $\varepsilon(L)$ is a closed formula expressing the existence of a solution for ${}_{\mathsf{pre}}\varepsilon(L)$.

6 Verification of Reduction

Definition 6.1: Let $\mathcal{M} \vDash \mathsf{PA}$, then we recursively define the function $\nu : \mathbb{N} \to D$ by

$$\nu(0) := 0^I \quad , \quad \nu(k+1) := \nu(k) +^I S^I 0^I$$

and we call $\nu(k)$ the **numeral** of k in $\mathcal{M}$.

Lemma 6.2: *Let $\mathcal{M}$ be any PA model and $x, y : \mathbb{N}$, then we have*

$$\nu(x + y) = \nu(x) +^I \nu(y) \quad \textit{and} \quad \nu(x \cdot y) = \nu(x) \cdot^I \nu(y)$$

Proof: Both equalities can be shown by induction over $x : \mathbb{N}$ and using the fact that the axioms of PA hold in the model. $\square$

Lemma 6.3: *Let $\mathcal{M}$ be some PA model, L some* `h10c` *problem and $\sigma : \mathbb{N} \to \mathbb{N}$ an environment, then*

$$\forall\, a \in L. [a]_\sigma \;\to\; \mathcal{M} \vDash_{\nu \circ \sigma} {}_{\mathsf{pre}}\varepsilon(L)$$

showing that the numerals given by σ satisfy the pre-embedding of a.

Proof: We prove this by induction on the list L. If L is empty, this is trivial, so assume $L = a :: L'$. We do a case analysis on the form of a, only showing the second case since all others are similar. So assume a is of the form $x_i + x_j = x_k$ and σ satisfies it. Then

$$[x_i + x_j = x_k]_\sigma \;\leftrightarrow\; \sigma(i) + \sigma(j) = \sigma(k)$$

Lemma 6.2 and the above then imply

$$\nu(\sigma(k)) = \nu\Big(\sigma(i) + \sigma(j)\Big) = \nu(\sigma(i)) +^I \nu(\sigma(j))$$

which is equivalent to $\mathcal{M} \vDash_{\nu \circ \sigma} (v_i + v_j \equiv v_k)$. Using the induction hypothesis this gives us

$$\mathcal{M} \vDash {}_{\mathsf{pre}}\varepsilon(a) \;\wedge\; \mathcal{M} \vDash {}_{\mathsf{pre}}\varepsilon(L') \;\leftrightarrow\; \mathcal{M} \vDash {}_{\mathsf{pre}}\varepsilon(a :: L') \qquad \square$$

The next result tells us that, when we are in the standard model, we can even show that Lemma 6.3 holds as an equivalence.

Lemma 6.4: *Let $\mathbb{N}$ be the standard model of PA, a :* `h10c` *and $\sigma : \mathbb{N} \to \mathbb{N}$ some environment, then*

$$\forall\, a \in L.\ [a]_\sigma \;\leftrightarrow\; \mathbb{N} \vDash_\sigma {}_{\mathsf{pre}}\varepsilon(L)$$

Proof: If $\mathcal{M} = \mathbb{N}$, then ν is the identity function on $\mathbb{N}$ which makes it possible to run all steps from the prove of Lemma 6.3 in reverse. $\square$

Lemma 6.5: *Let $\mathcal{M}$ be some model with domain D, $\phi : \mathbb{F}$ and ρ some environment. Then we have*

$$\mathcal{M} \vDash_\rho (\exists^N \phi) \;\leftrightarrow\; \exists\, d_1 : D \ldots \exists\, d_N : D.\ \mathcal{M} \vDash_{d_1;\ldots;d_N;\rho} \phi$$

Proof: We do an induction over N, leaving out the trivial base case. In the following we first apply the definition of $\mathcal{M} \vDash_\rho$, then the induction hypothesis.

$$\begin{aligned}
\mathcal{M} \vDash_\rho (\exists^{N+1}\phi) &\leftrightarrow \mathcal{M} \vDash_\rho (\exists v\ \exists^N \phi) \\
&\leftrightarrow \exists\, d : D.\ \mathcal{M} \vDash_\rho (\exists^N \phi) \\
&\leftrightarrow \exists\, d : D, \exists\, d_1 : D \ldots \exists\, d_N : D.\ \mathcal{M} \vDash_{d;d_1;\ldots;d_N;\rho} \phi
\end{aligned} \qquad \square$$

Lemma 6.6: *Let $\mathcal{M}$ be some model with domain D, t a term largest variable appearing is smaller than v_N and $\rho, \sigma : \mathbb{N} \to D$ two environments which agree up to N. Then we have $\hat{\rho}(t) = \hat{\sigma}(t)$.*

Proof: Note that when a term $t = ft_1 \dots t_n$ has smaller variables than v_N , its constituents t_j must also have variables smaller than v_N.

Knowing this, the result then follows by structural induction on the term t. □

Lemma 6.7: *Let $\mathcal{M}$ be some model with domain D, ϕ a formula whose largest variable appearing is smaller than v_N and $\rho, \sigma : \mathbb{N} \to D$ two environments which agree up to N. Then we have*

$$\mathcal{M} \vDash_\rho \phi \;\leftrightarrow\; \mathcal{M} \vDash_\sigma \phi$$

Proof: Similarly to terms, note that whenever a formula ϕ with smaller variables than v_N is composed of several formulas, either by logical connectives or quantifiers, then the constituents also have variables smaller than v_N.

We show the equivalence (where we quantify over ρ, σ) by structural induction on the formula ϕ. For $\phi = \bot$ this is trivial. If $\phi = Pt_1 \dots t_n$ for some predicate symbol P and terms $t_1 \dots t_n$, we have

$$\mathcal{M} \vDash_\rho Pt_1 \dots t_n \;\leftrightarrow\; P^I \hat{\rho}(t_1) \dots \hat{\rho}(t_n) \;\leftrightarrow\; P^I \hat{\sigma}(t_1) \dots \hat{\sigma}(t_n) \;\leftrightarrow\; \mathcal{M} \vDash_\sigma Pt_1 \dots t_n$$

by using Lemma 6.6 on the terms.

For $\phi = L\phi_1\phi_2$, where L is a logical connective and ϕ_1, ϕ_2 formulas, we have

$$\begin{aligned} \mathcal{M} \vDash_\rho L\phi_1\phi_2 \;&\leftrightarrow\; \hat{L}(\mathcal{M} \vDash_\rho \phi_1)(\mathcal{M} \vDash_\rho \phi_2) \\ &\leftrightarrow\; \hat{L}(\mathcal{M} \vDash_\sigma \phi_1)(\mathcal{M} \vDash_\sigma \phi_2) \;\leftrightarrow\; \mathcal{M} \vDash_\sigma L\phi_1\phi_2 \end{aligned}$$

where we used the induction hypothesis on ϕ_1, ϕ_2. In the quantifier case $\phi = Q\ v, \alpha$ we similarly have

$$\begin{aligned} \mathcal{M} \vDash_\rho Q\ \alpha \;&\leftrightarrow\; \hat{Q}\ d : D.\ \mathcal{M} \vDash_{d;\rho} \alpha \\ &\leftrightarrow\; \hat{Q}\ d : D.\ \mathcal{M} \vDash_{d;\sigma} \alpha \;\leftrightarrow\; \mathcal{M} \vDash_\sigma Q\ \alpha \end{aligned}$$

by the induction hypothesis applied to α. □

We can now put everything together to verify the reduction.

Proof: (of Theorem 4.2) We show both implications of the equivalence. First, assume $\mathsf{PA} \vDash \varepsilon(L)$. By Lemma 6.5 this is equivalent to

$$\forall \mathcal{M}\ \rho.\ \exists d_1 : D \dots \exists d_N : D.\ \mathcal{M} \vDash_{d_1;\dots;d_N;\rho}\ {}_{\mathsf{pre}}\varepsilon(L)$$

So in particular, for the standard model $\mathbb{N}$ of PA and $\rho = \mathrm{id}$ we get natural numbers $d_1 \dots d_N : \mathbb{N}$ such that $\mathbb{N} \vDash_{d_1;\dots;d_N;\mathrm{id}}\ {}_{\mathsf{pre}}\varepsilon(L)$. Lemma 6.4 then tells us that L

is satisfied by $d_1; \dots; d_N; \mathrm{id}$.

For the other implication, assume that $\mathsf{sat}_{\mathsf{h10c}}(L)$ holds; meaning there is a $\sigma : \mathbb{N} \to \mathbb{N}$ such that all atomic equations in L are satisfied. Now let $\mathcal{M}$ be any model of PA. We want to show $\mathcal{M} \vDash \varepsilon(L)$, which by Lemma 6.5 is equivalent to

$$\forall \rho.\ \exists d_1 : D \dots \exists d_N : D.\ \mathcal{M} \vDash_{d_1;\dots;d_N;\rho} {}_{\mathsf{pre}}\varepsilon(L)$$

For any ρ, we set $d_k := \nu \circ \sigma(k)$ for every $k = 1, \dots, N$ and now need to show

$$\mathcal{M} \vDash_{d_1;\dots;d_N;\rho} {}_{\mathsf{pre}}\varepsilon(L).$$

By Lemma 6.7, the environments $d_1; \dots; d_N; \rho$ and $\nu \circ \sigma$ are interchangeable, since they agree up to N and by the definition of ε there is no larger variable than v_N in ${}_{\mathsf{pre}}\varepsilon(L)$. So the above is equivalent to

$$\mathcal{M} \vDash_{\nu \circ \sigma} {}_{\mathsf{pre}}\varepsilon(L)$$

which now follows from Lemma 6.3 and the fact that σ was such that it satisfied the problem L. $\square$

7 Observations

Going through the proofs, there are some observations we can make with regards to what we need in α_{PA} in order to establish the result.

It turns out that, if we take the reduced list of axioms α_R only containing the four axioms: zero addition, recursion for addition, zero multiplication and recursion for multiplication, all proofs we did still work out, mainly because they are only really needed in Lemma 6.2. So we can show

$$\forall L.\ \mathsf{sat}_{\mathsf{h10c}}\ L \ \leftrightarrow\ \forall \mathcal{M}.\ \mathcal{M} \vDash \alpha_R\ \to\ \mathcal{M} \vDash \varepsilon(L)$$

Bibliography

[1] N. G. de Bruijn. "*Lambda calculus notation with nameless dummies, a tool forautomatic formula manipulation, with application to the Church-Rosser theorem*". In: *Indagationes Mathematicae* 34 (1972), pp. 381–392.

[2] Yannick Forster Dominique Larchey-Wendling. "*Hilbert's Tenth Problem in Coq*". 2019.

[3] Yuri V. Matijasevič. "*Enumerable sets are Diophantine*". In: *Soviet Mathematics: Doklady* 11 (1970), pp. 354–357.

[4] Yuri V. Matiyasevich. On Hilbert's Tenth Problem. Expository Lectures 1. 2000. URL: https://mathtube.org/sites/default/files/lecture-notes/Matiyasevic (accessed: 11.06.2020).

[5] Dominik Wehr Yannick Forster Dominik Kirst. "*Completeness Theorems for First-Order Logic Analysed in Constructive Type Theory*".