Friendly Logics, Fall 2015, Lecture Notes 4

Val Tannen

1 Arithmetic vocabularies

In this section we discuss a *FO language for arithmetic* that will subsequently be used for the fundamental results of Gödel et al. Consider first the vocabulary consisting only of the binary function $+$ (we use the syntax $t_1 + t_2$ instead of $+(t_1, t_2)$). As usual, we also assume that equality is in our language. Notice that some of the other arithmetic operations and can already be defined as "shorthand" for formulas with $+$, albeit in predicate form, e.g.:

ordering: $t_1 \leq t_2 \stackrel{\text{def}}{=} \exists x\ t_1 + x = t_2$

strict ordering: $t_1 < t_2 \stackrel{\text{def}}{=} t_1 \leq t_2 \wedge t_1 \neq t_2$

zero: $zero(x) \stackrel{\text{def}}{=} \forall y\ \ x \leq y$

one: $one(x) \stackrel{\text{def}}{=} \forall y, z\ \ y < x \wedge z < x \Rightarrow y = z$

successor: $succ(x, y) \stackrel{\text{def}}{=} x < y \wedge (\forall z\ \ x < z \Rightarrow y \leq z)$

These shorthands are correct with respect to the truth of these FO formulas in the (standard) model $(\mathbb{N}, +)$. For example

$$\mathbb{N}, x \mapsto n \models one(x) \quad \text{iff} \quad n = 1$$

Here is a more general definition:

Definition 1.1 *A k-ary relation $R \subseteq \mathbb{N}^k$ is **definable** in the model $(\mathbb{N}, +)$ if there exists a formula $\varphi(x_1, \ldots, x_k)$ such that*

$$\forall n_1, \ldots, n_k \in \mathbb{N} \quad (n_1, \ldots, n_k) \in R \quad \text{iff} \quad \mathbb{N}, x_1 \mapsto n_1, \ldots, x_k \mapsto n_k \models \varphi(x_1, \ldots, x_k)$$

Therefore, the predicates for ordering, zero, one, and successor are definable from just $+$.

It is awkward to have zero, one, and successor in predicate form. So we will assume that we have available a constant symbol $\underline{0}$ and a unary function symbol $\underline{succ}$ and that their use is allowed by the following shorthands: for any formula $\varphi(x)$

$$\begin{aligned} \varphi(\underline{0}) &\stackrel{\text{def}}{=} \exists x\ zero(x) \wedge \varphi(x) \\ \varphi(\underline{succ}(t)) &\stackrel{\text{def}}{=} \exists y\ succ(t, y) \wedge \varphi(y) \end{aligned}$$

Alternatively, we can assume that we have available two constants $\underline{0}$ and $\underline{1}$ with shorthands as the first one above. Of course we could define $\underline{1} \stackrel{\text{def}}{=} \underline{succ}(\underline{0})$, or, alternatively, $\underline{succ}(t) \stackrel{\text{def}}{=} t + \underline{1}$. Whichever convention we adopt, we need the following:

Definition 1.2 (Numerals) *For each natural number, $n \in \mathbb{N}$, we have in our language for arithmetic a term $\underline{n}$, called the* ***numeral*** *corresponding to n, such that $\mathbb{N}, x \mapsto m \models x = \underline{n}$ iff $m = n$.*

Indeed, we already have $\underline{0}$. For $n > 0$ we have, in the two alternatives we discussed

$$\underline{n} \stackrel{\text{def}}{=} \underline{succ}(\cdots \underline{succ}(\underline{0}) \cdots) \quad (n \text{ successors})$$

or

$$\underline{n} \stackrel{\text{def}}{=} (\cdots (\underline{0} + \underline{1}) \cdots) + \underline{1} \quad (n\ 1'\text{s})$$

Once we have the numerals we can restate definablity in terms of truth of sentences:

$$\forall n_1, \ldots, n_k \in \mathbb{N} \quad (n_1, \ldots, n_k) \in R \quad \text{iff} \quad \mathbb{N} \models \varphi(\underline{n_1}, \ldots, \underline{n_k})$$

and thus we can talk about a predicate on natural numbers being definable in $Th(\mathbb{N}, +)$, the theory consisting of the set of sentences that are *true* in $(\mathbb{N}, +)$.

There is a glaring omission in the examples above: multiplication. Is multiplication is definable in $Th(\mathbb{N}, +)$? The answer is negative, and it follows from the decidability of $Th(\mathbb{N}, +)$ (Presburger) in contrast to the undecidability of $Th(\mathbb{N}, +, \cdot)$ (Gödel-Church-Kleene). [1]

Note that multiplication can be expressed with primitive recursion in terms of addition:

$$\begin{aligned} m \cdot 0 &= 0 \\ m \cdot succ(n) &= (m \cdot n) + m \end{aligned}$$

and that something very similar holds for expressing exponentiation in terms of multiplication:

$$\begin{aligned} m \hat{} 0 &= 1 \\ m \hat{} succ(n) &= (m \hat{} n) \cdot m \end{aligned}$$

Therefore, it is interesting to note that

Proposition 1.1 (Gödel) *Exponentiation is definable in $Th(\mathbb{N}, +, \cdot)$. (In fact, exponentiation is even* provably representable *in a finite axiomatization (no induction) called* Robinson Arithmetic*.)*

For here on, we will assume a language for arithmetic that has $+$, hence numerals, and $\cdot$. We will use $Th(\mathbb{N})$ as a shorter notation for $Th(\mathbb{N}, +, \cdot)$.

[1] $Th(\mathbb{N}, +)$ is also called Presburger Arithmetic, after the Polish mathematician Mojżesz Presburger. For a proof its decidability you can consult either Enderton's textbook or Sipser's textbook. The undecidability of $Th(\mathbb{N}, +, \cdot)$ is shown later in these notes.

2 Arithmetization of syntax (Gödel numbering and representability)

Formulas and proofs are strings and we rely on our knowledge of Computability Theory on strings. So how do these relate to arithmetic?

Gödel showed how the string manipulations manifested in logical systems can be encoded by $(+,\cdot)$-FOL formulas that talk about natural numbers. The encoding trick that he used is the following: if we consider strings over, say, $\{a,b,c\}$, we first associate numbers (>0) to the letters, say $a \longmapsto 1, b \longmapsto 2, c \longmapsto 3$, and then we encode a (non-empty) string by

$$\sharp(d_1 d_2 \cdots d_n) \stackrel{\text{def}}{=} 2^{k_1} 3^{k_2} \cdots p_n^{k_n}$$

where p_n is the n'th prime number and $d_i \longmapsto k_i$. For example $\sharp(bbacabc) = 2^2 3^2 5^1 7^3 11^1 13^2 17^3$. A string u is therefore encoded as a natural number $\sharp(u)$ (the **Gödel number** of u).

In view of the well-known theorem on the unique prime factorization of natural numbers, Gödel's encoding is injective. It is also total computable, the set of natural numbers that are Gödel numbers of some string is decidable, and on that set we can define a computable "decoding" function that inverts Gödel's encoding. This establishes a nice correspondence between strings and numbers.

But not nice enough for the pedagogical needs of these notes. In fact, in Computer Science we are used to the development of Computability Theory using strings and Turing machines. The same concepts can be developed using natural numbers under the preferred name Recursion Theory. To make use in the study of arithmetic theories of the computability results we are familiar with, it is convenient to assume that we have *some* Gödel numbering that is, of course, computable and has an additional property:

$$\forall n \in \mathbb{N} \quad \sharp(\underline{n}) \;=\; n$$

i.e., the Gödel number of a numeral (which is a string) is the number corresponding to the numeral. This makes the encoding surjective (but not injective), with a computable right inverse. [2]

By composing Gödel numbering with numeral representation we obtain the following:

Definition 2.1 (Gödel numerals) *To any string u we associate its* G"odel numeral*, with notation $\ulcorner w \urcorner \stackrel{\text{def}}{=} \underline{\sharp(w)}$, which is a term in our arithmetic language.*

So strings can be represented in arithmetic theories as Gödel numberals. By the way, we can now see that we need arithmetic theories with multiplication. Since Gödel's encoding involves primes, the multiplication function symbol will be used extensively (the same for exponentiation but we saw in Proposition 1.1 that it can be dispensed of). By the way, for a glimpse on how to use multiplication, note the following statement, which says that infinitely many twin primes exist (open, famous conjecture):

$$\forall n \exists p \, \forall x, y \, [p > n \wedge (x, y > 1 \Rightarrow (x \cdot y \neq p \wedge x \cdot y \neq p + 2))]$$

[2] The original Gödel numbering is injective but not surjective. I am aware of alternative Gödel numberings that are bijective, and thus cannot have the property I need. I am assuming that this property can be made true at the same time as all other properties of Gödel numbers needed in the proof of Theorem 2.1 below.

Given a scheme for Gödel numbering, we can now try to capture *sets* of strings using *arithmetic FO theories*, that is, subsets $T \subseteq Th(\mathbb{N})$ that are closed under FO provability, i.e., $\forall \sigma \ T \vdash \sigma \Rightarrow \sigma \in T$. In addition to Presburger Arithmetic and $Th(\mathbb{N})$ itself, arithmetic FO theories of interest include Peano Arithmetic (PA), which is axiomatized by Peano's axioms with the induction axiom *schema* restricted to FO formulas, Robinson Arithmetic (Q), which has a finite axiomatization (no induction axiom schema), and something called Primitive Recursive Arithmetic (PRA) of which we shall speak no more.

For reference, here is a version of Peano's axioms:

$$\forall x \ \ \underline{succ}(x) \neq \underline{0} \qquad \forall x, y \ \ \underline{succ}(x) = \underline{succ}(y) \ \Rightarrow \ x = y$$

$$\forall x \ \ x + \underline{0} = x \qquad \forall x, y \ \ x + (\underline{succ}(y)) = \underline{succ}(x + y)$$

$$\forall x \ \ x \cdot \underline{0} = \underline{0} \qquad \forall x, y \ \ x \cdot (\underline{succ}(y)) = (x \cdot y) + x$$

$$\varphi(\underline{0}) \ \wedge \ (\forall x \ \ \varphi(x) \ \Rightarrow \ \varphi(\underline{succ}(x))) \ \Rightarrow \ (\forall y \ \varphi(y))$$

where $\varphi(x)$ is any FO formula with one free variable.

Definition 2.2 *Let T be an arithmetic FO theory. Let $L \subseteq \Sigma^*$ be a language, and $f : \Sigma^* \to \Sigma^*$ be a total function. Let φ_L be a formula with exactly one free variable, and θ_f be a formula with exactly two free variables.*

L **is definable** *in $Th(\mathbb{N})$ by φ_L if* $\quad \forall w \in \Sigma^* \quad w \in L \ \Leftrightarrow \ \mathbb{N} \models \varphi_L(\ulcorner w \urcorner)$

L **is weakly represented** *in T by φ_L if* $\quad \forall w \in \Sigma^* \quad w \in L \ \Leftrightarrow \ \varphi_L(\ulcorner w \urcorner) \in T$

L **is strongly represented** *in T by φ_L if* $\ \forall w \in \Sigma^* \ \ w \in L \ \Rightarrow \ \varphi_L(\ulcorner w \urcorner) \in T \ \wedge \ w \notin L \ \Rightarrow \ \neg\varphi_L(\ulcorner w \urcorner) \in T$

f **is functionally represented** *in T by θ_f if* $\quad \forall w \in \Sigma^*$

(i) $\quad \theta_f(\ulcorner w \urcorner, \ulcorner f(w) \urcorner) \in T \quad \wedge$

(ii) $\quad \forall y \, [\theta_f(\ulcorner w \urcorner, y) \ \Rightarrow \ y = \ulcorner f(w) \urcorner] \in T$

Since tuples of strings can be encoded as strings we will feel free to assume that the definition generalizes to predicates on tuples of strings and to total functions of multiple variables. Also,

Note that the definition above makes sense only if T is consistent (it does not equal the set of all sentences!), in which case strong representability implies weak representability (by the same formula). Note also that strong representability in $Th(\mathbb{N})$ coincides with definability.

Gödel's insight was to show that the predicate $Bew_{\mathrm{PA}}(\sigma, \pi)$ which holds iff π is a proof of the sentence σ in PA, the function $Concl_{\mathrm{PA}}(\pi)$ that extracts the sentence that π proves, and the closely related predicate $Prov_{\mathrm{PA}}(\sigma)$ which holds iff σ is provable in PA [3] are all representable in PA (respectively, strongly, functionally, and weakly). In fact, Gödel proved that an entire class of decidable predicates, namely those

[3] Actually Gödel used a fragment of the formal system designed by Russel and Whitehead in "Principia Mathematica".

captured by primitive recursion, including $Bew_{\text{PA}}()$, are strongly representable in PA. Of course, Gödel did not have a definition of general computability [4] in 1930. However, Gödel's paper already contained enough ingredients which, when combined with a definition of computability, allowed Church and Kleene to show the following: [5]

Theorem 2.1 (Gödel-Church-Kleene) *Any decidable language/predicate is strongly representable in PA.*

Corollary 2.2 *Any r.e. language/predicate is weakly representable in PA and definable in $Th(\mathbb{N})$ (by the same formula).*

Proof If L is r.e. then there exists a decidable predicate R such that $u \in L$ iff there exists a v s.t. $R(u,v)$. By Theorem 2.1, R is strongly representable by some formula $\varphi_R(x,y)$. Let $\varphi_L(x) \equiv \exists y\, \varphi_R(x,y)$. We show that φ_L both weakly represents L in PA and defines L in $Th(\mathbb{N})$.

If $u \in L$ then $R(u,v)$ for some v hence $PA \vdash \varphi_R(\ulcorner u \urcorner, \ulcorner v \urcorner)$ so $PA \vdash \exists y\, \varphi_R(\ulcorner u \urcorner, y)$ and therefore $\mathbb{N} \models \exists y\, \varphi_R(\ulcorner u \urcorner, y)$.

If $u \notin L$ then for all v it is not the case that $R(u,v)$, i.e., for all v $PA \vdash \neg\varphi_R(\ulcorner u \urcorner, \ulcorner v \urcorner)$. Now we use the fact that our Gödel numbering is surjective. For any $n \in \mathbb{N}$ there is some v_n such that $n = \sharp(v_n)$, [6] therefore $\ulcorner v_n \urcorner = \underline{n}$. It follows that for all $n \in \mathbb{N}$ we have $PA \vdash \neg\varphi_R(\ulcorner u \urcorner, \underline{n})$, hence $\mathbb{N} \models \neg\varphi_R(\ulcorner u \urcorner, \underline{n})$. It follows that $\mathbb{N} \not\models \exists y\, \varphi_R(\ulcorner u \urcorner, y)$ and therefore $PA \not\vdash \exists y\, \varphi_R(\ulcorner u \urcorner, y)$. $\square$

The argument above is straightforward because it is using the soundness of PA in $\mathbb{N}$. To show the weak representability of r.e. sets without any reference to soundness, we would assume that PA is consistent, and, for the last part, that it is ω-consistent. [7] That's what Gödel did.

Next we have a corollary that connects computability to functional representability. We state the corollary for functions of two arguments but it clearly holds for functions of any number of arguments.

Corollary 2.3 *Any total computable function is functionally representable in PA.*

Proof We will prove it for two arguments but the proof clearly generalizes to any number of arguments. Let $f : \Sigma^* \times \Sigma^* \to \Sigma^*$ be a total computable function.

Consider the predicate $F(u,v,w)$ iff $f(u,v) = w$. Since f is total computable, F is decidable. By Theorem 2.1 there exists a formula $\varphi_F(x,y,z)$ that strongly represents F, that is:

$$\forall u,v,w \in \Sigma^* \quad f(u,v) = w \;\Rightarrow\; PA \vdash \varphi_F(\ulcorner u \urcorner, \ulcorner v \urcorner, \ulcorner w \urcorner) \quad \wedge \quad f(u,v) \neq w \;\Rightarrow\; PA \vdash \neg\varphi_F(\ulcorner u \urcorner, \ulcorner v \urcorner, \ulcorner w \urcorner)$$

[4] Logicians were already aware of examples of (total) computable functions that were not primitive recursive (Sudan/Ackermann).

[5] The proof of this theorem is where all the hard work is done so of course I will skip it! I agree with Smorynski who says about such a proof that its details are great fun to work out but very boring to read.

[6] In fact, by the earlier assumption, we can even take $v_n = \underline{n}$.

[7] Namely that PA does not simultaneously prove $\neg\varphi(\underline{0}), \neg\varphi(\underline{1}), \ldots$ and $\exists x \varphi(x)$.

Can we use φ_F to functionally represent f? This would clearly take care of the condition (i), but not of (ii). Why? Intuitively, PA cannot "insure" that to prove universal statements it suffices to verify them for numerals. (In fact, there exist models of PA that are not isomorphic to the natural numbers, the so-called *non-standard models of arithmetic*.)

However, the following trick fixes the problem. Take

$$\theta_f(x,y,z) \quad \equiv \quad \varphi_F(x,y,z) \wedge \forall z'[z' < z \Rightarrow \neg\varphi_F(x,y,z')]$$

We need to show that for all $u, v \in \Sigma^*$ we have

(i) $\quad PA \vdash \varphi_F(\ulcorner u \urcorner, \ulcorner v \urcorner, \ulcorner f(u,v) \urcorner) \wedge \forall z'[z' < \ulcorner f(u,v) \urcorner \Rightarrow \neg\varphi_F(\ulcorner u \urcorner, \ulcorner v \urcorner, z')]$

(ii) $\quad PA \vdash \forall z\,[\varphi_F(\ulcorner u \urcorner, \ulcorner v \urcorner, z) \wedge \forall z'[z' < z \Rightarrow \neg\varphi_F(\ulcorner u \urcorner, \ulcorner v \urcorner, z')] \;\Rightarrow\; z = \ulcorner f(u,v) \urcorner]$

We prove (i). Clearly $PA \vdash \varphi_F(\ulcorner u \urcorner, \ulcorner v \urcorner, \ulcorner f(u,v) \urcorner)$ by the strong representability of F, so we only have to show

(i2) $\quad PA \vdash \forall z'[z' < \ulcorner f(u,v) \urcorner \Rightarrow \neg\varphi_F(\ulcorner u \urcorner, \ulcorner v \urcorner, z')]$

The key observation (which suggested the trick in the definition of θ_f) is the following. For any natural number $n > 0$, we have

$(*) \quad PA \vdash \forall z'\,[z' < \underline{n} \Leftrightarrow z' = \underline{0} \vee \cdots \vee z' = \underline{n-1}]$

We skip the proof of this observation.

Now let $n = \sharp(f(u,v))$. If $n = 0$ the statement (i2) is vacuously provable. Suppose $n > 0$. Then for any $m < n$ we have $\underline{m} \neq f(u,v)$ (here we use $m = \sharp(\underline{m})$) and (i2) follows again from $(*)$ combined with the strong representability of F.

Note that the argument we just made also gives the following stronger fact: for any $n \leq \sharp(f(u,v))$ we have

$PA \vdash \forall z'[z' < \underline{n} \Rightarrow \neg\varphi_F(\ulcorner u \urcorner, \ulcorner v \urcorner, z')]$

From this (ii) follows by contradiction and two cases: $z < \ulcorner f(u,v) \urcorner$ is one case, $z > \ulcorner f(u,v) \urcorner$ is the other. Details are omitted. $\square$

The theorem and the two corollaries above continue to hold if we replace PA with one of the weaker theories, Q or PRA. A more interesting observation is the proofs are constructive (effective). By this we mean that the proof of Theorem 2.1 actually shows how to compute the representing formula φ_L taking as input the axioms and proof rules of PA (or Q, or PRA) together with a description of a TM that decides L. It is then clear, from the proofs of Corollaries 2.2 and 2.3 that the representing formulas there are also computable.

Now given a PA proof π we can compute the sentence that π proves and it is decidable whether that sentence is a given one, σ. Moreover the set of sentences provable in PA is r.e. (as is the case with every theory axiomatized by a decidable set of axioms). Therefore, we have Gödel's insight that what the PA proof system does can be captured "inside" PA:

Corollary 2.4 *$Bew_{\text{PA}}(\sigma, \pi)$ is strongly representable in PA. $Concl_{\text{PA}}(\pi)$ is functionally representable in PA. $Prov_{\text{PA}}(\sigma)$ is weakly representable in PA and the same formula defines it in $Th(\mathbb{N})$.*

We shall see later how to exploit this for Gödel's incompleteness theorems. We end this section with a discussion of the converses of the representability results above.

Proposition 2.5 *Let T be an* r.e. *arithmetic FO theory. Any language weakly representable in T is r.e. If moreover T is* consistent *then any language strongly representable in T is decidable.*

Proof Let L be weakly represented by φ_L in T. Then $L \leq_m T$ via the reduction $w \longmapsto \varphi_L(\ulcorner w \urcorner)$ and since T is r.e. L must be r.e. too.

Now assume that T is consistent. Then, any language strongly represented by a formula is also weakly represented by that formula hence it is r.e. But it follows from the definition of strong representability that if L is strongly representable then so is its complement $\overline{L}$, in fact by the negation of the formula used for L. Therefore, both L and $\overline{L}$ are r.e. so L is decidable. $\square$

Now, because PA is consistent [8] and r.e. we have that a language is decidable iff it is strongly representable in PA and is r.e. iff it is weakly representable in PA, a **logical characterization** of the two main concepts of Computability Theory (and such results can be extended beyond r.e.-ness, leading to the Arithmetic Hierarchy of languages).

3 The computability perspective on Gödel's First

Theorem 2.1 and Corollary 2.2 already lead to a form of Gödel's First Incompleteness Theorem, just by using the tools of computability.

Theorem 3.1 (Computability Formulation of Gödel's First) *$Th(\mathbb{N})$ is not r.e. It follows that there exists an unprovable (in PA) but true sentence (hence PA cannot prove its negation either).*

Proof Recall that $\overline{K}$ is the complement of the Halting Problem. We show that $\overline{K} \leq_m Th(\mathbb{N})$ and the result follows.

Since K is r.e., it follows from Corollary 2.2 that K is definable in $Th(\mathbb{N})$, i.e., there exists a formula $\chi(x, y)$ such that for any TM and any input, M halts on w iff $\mathbb{N} \models \chi(\ulcorner \langle M \rangle \urcorner, \ulcorner w \urcorner)$. Therefore

$$\langle M, w \rangle \quad \longmapsto \quad \neg\chi(\ulcorner \langle M \rangle \urcorner, \ulcorner w \urcorner)$$

provides the desired reduction $\overline{K} \leq_m Th(\mathbb{N})$.

[8] Wait a minute, PA is consistent? Wasn't that Hilbert's Second Problem? And wasn't Hilbert's Program shattered by Gödel? We stated that PA is consistent simply because its proof system is sound in the model $\mathbb{N}$. This a perfectly correct statement in the math that we use to conduct our *meta-study* of logic (presumably Zermelo-Fraenkel set theory). But this is not what Hilbert wanted, he wanted a proof of consistency by "finitistic" means. This was not made clear in his statement of the Second Problem but he explained it later.

Now, since PA (the theory) is r.e. there exists some true (in $Th(\mathbb{N})$) sentence σ that is not provable in PA. But then, $\neg\sigma$ is untrue and therefore cannot be provable either, since PA is sound. $\square$

This theorem implies not only that PA does not provide a complete axiomatization of $Th(\mathbb{N})$, but, moreover, that no reasonable (with a decidable set of axioms) axiomatization can be complete.

The second part of theorem is closer to Gödel's meaning of "incompleteness", but, still, this is not quite the form in which Gödel proved his result. He took a particular proof system and *computably* constructed a sentence which was not provable and whose negation was unprovable too. The meaning of his sentence was "I am not provable"! We will get to Gödel's construction in the next section, however Computability Theory has a way to constructing such a sentence also, using Kleene's Recursion Theorem. As shown in Sipser's textbook, this theorem allows us to program Turing machines to "obtain their own description". Consider the following machine, called S:

S: "on any input
obtain own description $\langle S\rangle$
construct sentence $\rho \equiv \neg\chi(\ulcorner\langle S\rangle\urcorner, \ulcorner w_0\urcorner)$ ($\chi(x,y)$ as in the proof of Theorem 3.1; w_0 arbitrary string)
enumerate all PA proofs
if a proof of ρ is found halt
otherwise, go on forever" (since there are infinitely many proofs in PA)

Proposition 3.2 *The sentence ρ constructed by S is true but unprovable in PA.*

Proof Since χ weakly represents K, $\mathbb{N} \models \rho$ iff S does not halt on w_0. But S halts on w_0 (or any other input) iff $PA \vdash \rho$. Therefore $\mathbb{N} \models \rho$ iff $PA \nvdash \rho$. By the soundness of PA, the only way out of this is when ρ is true but unprovable. $\square$

Notice that ρ does not say "I am not provable", but rather "S does not halt on w_0". However, as we saw, ρ ends up being true iff it is unprovable.

4 Too much self-awareness (fixed points and reflection)

One of the consequences of Kleene's Recursion Theorem is that total computable functions have fixed points. In the computability approach based on lambda calculus, this result has a particularly nice formulation with the existence of a **fixed point combinator** (I am not sure about its history). Gödel had the insight that PA has the following *provable fixed point* property:

Lemma 4.1 (Fixed-point) *Let $\varphi(x)$ be any formula with exactly one free variable x. Then, there exists (can be computed!) a sentence σ such that*

$$PA \vdash \sigma \Leftrightarrow \varphi(\underline{\ulcorner\sigma\urcorner})$$

Proof Let S (for "substitution") be a function such that $S(\varphi(x), t) = \varphi(t)$ for any formula $\varphi(x)$ with one free variable and any term t. When the arguments are not of this form we make sure S returns some fixed string that is not a sentence. Therefore S is total computable and by Corollary 2.3 there exists a formula that functionally represents it. Testing whether the arguments of S are formulas/terms is decidable, hence strongly representable, hence we can construct a formula θ with three free variable that has the following, more precise properties, for any $\psi(x)$ and t

(i) $\theta(\ulcorner\psi(x)\urcorner, \ulcorner t\urcorner, \ulcorner\psi(t)\urcorner)$

(ii) $PA \vdash \forall z\, [\theta(\ulcorner\psi(x)\urcorner, \ulcorner t\urcorner, z) \;\Rightarrow\; z = \ulcorner\psi(t)\urcorner]$

Let $\varphi(x)$ be any formula with exactly one free variable. Using θ as above, consider the formula

$$\chi(x) \quad \equiv \quad \forall z[\theta(x, x, z) \Rightarrow \varphi(z)]$$

We claim that we can take σ, the desired provable fixed point of $\varphi(x)$, to be $\sigma \equiv \chi(\ulcorner\chi(x)\urcorner)$.

To show that it is a provable fixed point we restate (i) and (ii) for $\psi(x) \equiv \chi(x)$ and $t \equiv \ulcorner\chi(x)\urcorner$, therefore $\psi(t) \equiv \sigma$, while using the property $\sharp(\underline{n}) = n$ that our Gödel numbering has to replace $\ulcorner\ulcorner\chi(x)\urcorner\urcorner$ with just $\ulcorner\chi(x)\urcorner$:

(i) $PA \vdash \theta(\ulcorner\chi(x)\urcorner, \ulcorner\chi(x)\urcorner, \ulcorner\sigma\urcorner)$

(ii) $PA \vdash \forall z\, [\theta(\ulcorner\chi(x)\urcorner, \ulcorner\chi(x)\urcorner, z) \;\Rightarrow\; z = \ulcorner\sigma\urcorner]$

And using these one can conclude

$$PA \vdash \sigma \Leftrightarrow \varphi(\ulcorner\sigma\urcorner)$$

Left to right is immediate using (i). Right to left is a bit more subtle, and we skip it. □

Here is an immediate application of the fixed point lemma:

Theorem 4.2 (Tarski's Undefinability) *$Th(\mathbb{N})$ is not definable in $\mathbb{N}$.*

Proof Suppose $\tau(x)$ defines truth, i.e., defines the set of sentences $Th(\mathbb{N})$. Applying Lemma 4.1 to the formula $\neg\tau(x)$ we obtain obtain a sentence σ_T such that (using soundness of PA)

$$\mathbb{N} \models \sigma_T \Leftrightarrow \neg\tau(\ulcorner\sigma_T\urcorner)$$

σ_T says "I am false". Liar's Paradox! Therefore there is no sentence σ_T. Therefore there is no formula τ. □

You noticed that we did not need the "provable" aspect of the fixed points in Lemma 4.1. In fact, the essential idea behind Tarski's Theorem can be expressed using a diagonalization argument. As a warm-up, you should recall Cantor's proof by diagonalization that $2^{\mathbb{N}}$ is not countable. You can also read in

Sipser's textbook how to regard the proof of the undecidability of the Halting Problem as a diagonalization argument.

Alternative Proof by diagonalization of Tarski's Undefinability Theorem (4.2 above).

Actually, what we will prove is that the truth of formulas with one free variable on numerals is not definable. This implies that the truth of sentences is also undefinable (but this last step seems to require the definability of some computable syntactic manipulations).

Suppose there is a formula $\tau(x, y)$ such that for any formula $\varphi(x)$ and any numeral $\underline{n}$ we have

$$\mathbb{N} \models \varphi(\underline{n}) \quad \text{iff} \quad \mathbb{N} \models \tau(\ulcorner \varphi(x) \urcorner, \underline{n})$$

Now take $\kappa(x) \equiv \neg\tau(x, x)$. This is the "diagonal" construct because $\kappa(x)$ "differs" from any given $\varphi(x)$ "on" the numeral $\underline{n_d} = \ulcorner \varphi(x) \urcorner$. Indeed, $\mathbb{N} \models \varphi(\underline{n_d})$ iff $\mathbb{N} \models \tau(\ulcorner \varphi(x) \urcorner, \ulcorner \varphi(x) \urcorner)$ while $\mathbb{N} \models \kappa(\underline{n_d})$ iff $\mathbb{N} \models \neg\tau(\ulcorner \varphi(x) \urcorner, \ulcorner \varphi(x) \urcorner)$ $\square$

Because all r.e. sets are definable (Corollary 2.2) we have:

Corollary 4.3 (Computational form of Gödel's First Again) *$Th(\mathbb{N})$ is not r.e.*

Now we turn to the proof-theoretic form of Gödel's First, and this will allow us to derive also Gödel's Second.

Recall that $Prov_{\text{PA}}(\sigma)$ is r.e. so it is weakly representable in PA, and definable in $\mathbb{N}$ by some formula $\xi(x)$. Let's introduce the notation $\Box\sigma$ for the sentence $\xi(\ulcorner \sigma \urcorner)$. By weak representability we have the following "self-awareness" property of PA:

Lemma 4.4 (Reflection Property)

$$\forall \sigma \quad PA \vdash \sigma \quad \textit{iff} \quad PA \vdash \Box\sigma.$$

(PA proves a sentence iff it "knows" that it proves that sentence!)

Now apply Lemma 4.1 to $\neg\xi(x)$.

Lemma 4.5 (Gödel's sentence) *There exists (can be effectively constructed!) a sentence σ_G such that*

$$PA \vdash \sigma_G \Leftrightarrow \neg\Box\sigma_G$$

A subtlety: the existence of the Gödel's sentence shows that the reflection property cannot hold in the stronger form $\forall\sigma \;\; PA \vdash \sigma \Leftrightarrow \Box\sigma$ as this would imply PA inconsistent.

Theorem 4.6 (Proof-Theoretic Formulation of Gödel's First) *Let σ_G be the sentence constructed above. Then PA proves neither σ_G nor $\neg\sigma_G$.*

Proof Suppose $PA \vdash \sigma_G$. Then $PA \vdash \neg\Box\sigma_G$ by Lemma 4.5 and $PA \vdash \Box\sigma_G$ by Lemma 4.4. Hence PA is inconsistent. Contradiction.

Suppose $PA \vdash \neg\sigma_G$. Then $PA \vdash \Box\sigma_G$ by Lemma 4.5 and further $PA \vdash \sigma_G$ by Lemma 4.4. Again PA is inconsistent. Again contradiction. □

In the first proof of Tarski's Theorem, the emergence of the Liar's paradox was solved by the non-existence of the sentence σ_T. Gödel's sentence σ_G does exist, in fact we will use it for the proof of Gödel's Second. The resolution of the apparent paradox that it creates is that the sentence is not provable. [9]

5 Gödel's Second

Reasoning as in the proof of Theorem 4.6 can actually be performed inside PA. This is because PA is strong enough to have the following "self-awareness" properties (that were abstracted by Hilbert and Bernays from Gödel's proof):

Lemma 5.1 (Hilbert-Bernays derivability conditions) *PA has the following properties:*

D1 *For any sentence σ, if $PA \vdash \sigma$ then $PA \vdash \Box\sigma$ (reflection)*

D2 *For any sentence σ, $PA \vdash \Box\sigma \Rightarrow \Box\Box\sigma$ (PA "knows" it has the reflection property).*

D3 *For any sentences ρ and σ, $PA \vdash \Box(\rho \Rightarrow \sigma) \Rightarrow \Box\rho \Rightarrow \Box\sigma$ (PA "knows" it can do modus ponens).*

We saw reflection before. We skip the proof of the rest.

We can express the consistency of PA as an arithmetic FO sentence:

$Cons_{\text{PA}} \equiv \neg\Box\mathsf{false}$

Theorem 5.2 (Gödel's Second) $PA \nvdash Cons_{\text{PA}}$.

Proof Let σ_G be Gödel's sentence constructed earlier. Recall that $PA \vdash \sigma_G \Leftrightarrow \neg\Box\sigma_G$, in particular $PA \vdash \sigma_G \Rightarrow (\Box\sigma_G \Rightarrow \mathsf{false})$.

By D1, $PA \vdash \Box(\sigma_G \Rightarrow (\Box\sigma_G \Rightarrow \mathsf{false}))$. Then by D3 and (regular) modus ponens $PA \vdash \Box\sigma_G \Rightarrow \Box(\Box\sigma_G \Rightarrow \mathsf{false})$. Next we apply D3 and regular modus ponens on the conclusion of the previous implication obtaining $PA \vdash \Box\sigma_G \Rightarrow \Box\Box\sigma_G \Rightarrow \Box\mathsf{false}$, and since $\Box\mathsf{false}$ is equivalent to $\neg Cons_{\text{PA}}$, $PA \vdash \Box\sigma_G \Rightarrow \Box\Box\sigma_G \Rightarrow \neg Cons_{\text{PA}}$.

By D2 we also have $PA \vdash \Box\sigma_G \Rightarrow \Box\Box\sigma_G$. In boolean logic (even minimal logic!) from $p \Rightarrow (q \Rightarrow r)$ and $p \Rightarrow q$ we can deduce $p \Rightarrow r$. [10] Hence $PA \vdash \Box\sigma_G \Rightarrow \neg Cons_{\text{PA}}$ so $PA \vdash Cons_{\text{PA}} \Rightarrow \neg\Box\sigma_G$.

[9] Does there exist a true but unprovable sentence, as in Theorem 3.1? Yes, because one of σ_G or $\neg\sigma_G$ must be true. Which one?

[10] Inhabited by the S combinator!

By the fixed point equivalence $PA \vdash Cons_{\text{PA}} \Rightarrow \sigma_G$ and since by Gödel's First $PA \nvdash \sigma_G$ we conclude that $PA \nvdash Cons_{\text{PA}}$. $\square$

The following is an apparent generalization of Gödel's Second:

Theorem 5.3 (Löb) *For any sentence σ, if $PA \vdash \Box\sigma \Rightarrow \sigma$ then $PA \vdash \sigma$*

We skip the proof but we show

Corollary 5.4 (Gödel's Second again) $PA \nvdash Cons_{\text{PA}}$.

Proof Take $\sigma \equiv \mathsf{false}$ in Löb's theorem. Obtain that if if $PA \vdash Cons_{\text{PA}}$ then $PA \vdash \mathsf{false}$. Since PA is consistent, $PA \nvdash Cons_{\text{PA}}$. $\square$