CHAPTER 5. THE INCOMPLETENESS THEOREM

1. Introduction

Gödel's famous Incompleteness Theorem states that there are sentences true on $\mathcal{N}$ which are not consequences of **PA**. In fact, this holds for any reasonable set of axioms in place of **PA**, as we will explain later. Analogous results hold for arithmetic on many other number systems, such as $\mathbb{Z}$ and $\mathbb{Q}$, but we will not discuss them. The proof depends on the following three separate developments:

- The Completeness Theorem, which implies that consequences of **PA** are in fact deducible from **PA**.
- Gödel Numbering, by which formulas and deductions are coded by integers in such a way that the usual syntactical functions and relations on them become recursive functions and relations of the associated numbers.
- The Theorem that all recursive functions and relations are definable in **PA**, which enables us to talk about (the numbers coding) formulas and deductions using formulas and sentences of $\mathcal{L}_{\mathbb{N}}$.

With all of these pieces in place the proof of Incompleteness is quite easy — we will give a sentence σ which is true on $\mathcal{N}$ iff it is not provable from **PA**. In fact the sentence σ will *say* "I am not provable from **PA**". But before we do this we need to discuss Gödel Numbering, which we do in the next section.

2. Gödel Numbers

The idea behind Gödel numbers is very simple – first, assign non-negative integers to the symbols of a language; then every finite sequence of symbols, such as a formula, is assigned the sequence number of the sequence of numbers assigned to the symbols in the sequence. This sequence number is then called the *Gödel number* of the sequence. Since the usual operations on sequence numbers are recursive, the corresponding operations on formulas, for example, become recursive as operations on their Gödel numbers. This process may then be repeated, assigning Gödel numbers to finite sequences of formulas, for example deductions from some set of axioms.

The symbols of the language $\mathcal{L}_{\mathbb{N}}$ are the variables v_n for $n \in \mathbb{N}$, the other logical symbols $\neg$, $\rightarrow$, $\forall$, $=$, the punctuation symbols (,), and ,, and the non-logical symbols s, $+$, $\cdot$, $\bar{0}$, and $<$. We define $g(v_n) = 2n$ for all $n \in \mathbb{N}$, and we define $g(\neg) = 1$, $g(\rightarrow) = 3$, $\ldots$, $g(<) = 23$. The precise numbers assigned are not important. We only require that different symbols are assigned

different numbers and that the function Var defined by $Var(n) = g(v_n)$ is recursive.
Note: We use the notation of Chapter 2 for terms and formulas, thus we have $+(v_1, v_2)$, not $(v_1 + v_2)$, and $< (v_1, v_2)$, not $(v_1 < v_2)$.

Definition 2.1. Let $\epsilon_0 \dots \epsilon_n$ be a sequence of symbols. The *Gödel number* of the sequence, notation $\ulcorner \epsilon_0 \dots \epsilon_n \urcorner$, is the sequence number $\langle g(\epsilon_0), \dots, g(\epsilon_n)\rangle$.

For example $\ulcorner \bar{1} \urcorner = \ulcorner s(\bar{0}) \urcorner = \langle g(s), g((), g(\bar{0}), g())\rangle = \langle 15, 9, 21, 11\rangle$.

Definition 2.2. Suppose α_i is a sequence of symbols for each $i \le n$. Then the *Gödel number* of the sequence $\alpha_0, \dots, \alpha_n$ is $\ulcorner \alpha_0, \dots, \alpha_n \urcorner = \langle \ulcorner \alpha_0 \urcorner, \dots, \ulcorner \alpha_n \urcorner \rangle$.

For example $\ulcorner < (v_1, v_2), \forall v_1 \neg < (v_1, v_1) \urcorner = \langle \ulcorner < (v_1, v_2) \urcorner, \ulcorner \forall v_1 \neg < (v_1, v_1) \urcorner \rangle$.
The terms $\bar{n}$ are called *numerals*. We show below that the set of all Gödel numbers of numerals is recursive.

Definition 2.3. The function Num is defined by $Num(n) = \ulcorner\ \bar{n}\ \urcorner$.

Lemma 2.1. Num *is recursive.*

Proof. The equations $Num(0) = \ulcorner \bar{0} \urcorner$ and $Num(n+1) = \ulcorner s(\urcorner * Num(n) * \ulcorner) \urcorner$ define Num by primitive recursion. □

Note that $n < Num(n)$ for all n.

Lemma 2.2. *The set* N *of all Gödel numbers of numerals is recursive.*

Proof. By the remark preceding the statement of the Lemma we have $k \in N$ iff $\exists n(n < k \wedge Num(n) = k)$. Therefore N is recursive since Num is recursive and bounded quantification preserves recursivity. □

We give a second proof of Lemma 2.2 which does not use the fact that we have a recursive function listing the Gödel numbers of the numerals. This second proof uses Course-of-Values Recursion and will also be used when we show, for example, that the set of all Gödel numbers of terms is recursive.

We first give a definition by recursion of the set of all numerals as follows: α is a numeral iff either α is $\bar{0}$ or α is $s(\beta)$ for some numeral β. Since $\ulcorner\ \bar{n}\ \urcorner < \ulcorner \overline{n+1} \urcorner$ we can translate this definition to Gödel numbers of numerals in the following way. $k \in N$ iff $k = \ulcorner \bar{0} \urcorner \vee \exists l \big(l < k \wedge l \in N \wedge k = \ulcorner s(\urcorner * l * \ulcorner) \urcorner\big)$. Note that $l \in N$ is allowed in Course-of-Values Recursion since it is the same as $\big(\overline{K_N}(n)\big)_l = 1$.

We may reformulate the recursive definition of the set of terms as follows. t is a term iff either t is $\bar{0}$ or t is v_n for some n or there is a term t_1 such that t is $s(t_1)$ or there are terms t_1 amd t_2 such that t is $+(t_1, t_2)$ or t is $\cdot(t_1, t_2)$.

Theorem 2.1. *The set* Tm *of all Gödel numbers of terms of* $\mathcal{L}_{\mathbb{N}}$ *is recursive.*

Proof. We follow the recursive definition of the set of terms just given to see that $k \in Tm$ iff $k = \ulcorner \bar{0} \urcorner \vee \exists n (n < k \wedge k = \langle Var(n) \rangle) \vee \exists l \big(l < k \wedge l \in Tm \wedge k = \ulcorner s(\urcorner * l * \ulcorner) \urcorner\big) \vee \exists l_1 \exists l_2 \big(l_1 < k \wedge l_2 < k \wedge l_1 \in Tm \wedge l_2 \in Tm \wedge [k = \ulcorner +(\urcorner * l_1 * \ulcorner , \urcorner * l_2 * \ulcorner) \urcorner \vee k = \ulcorner \cdot(\urcorner * l_1 * \ulcorner , \urcorner * l_2 * \ulcorner) \urcorner]\big)$. □

In exactly the same way we see that the set of Gödel numbers of formulas is recursive. We define the set of of Gödel numbers of deductions from **PA** as follows: n is the Gödel number of a deduction iff n is a sequence number of has length $l \geq 1$ and for every $i < l$, $(n)_i$ is the Gödel number of a formula and this formula is either a logical axiom or an element of **PA** or there are $j, k < i$ such that
$(n)_k = \ulcorner(\urcorner * (n)_j * \ulcorner\rightarrow\urcorner * (n)_i * \ulcorner)\urcorner$.
Since the set of Gödel numers of the logical axioms and the set of Gödel numbers of sentences in **PA** are both recursive it follows that the set of Gödel numbers of deductions from **PA** is recursive.

We will need the following results in the proof of the Incompleteness Theorem.

Theorem 2.2. (a) *There is a recursive function* S *of two arguments such that whenever* $l = \ulcorner\varphi\urcorner$ *for some formula* $\varphi(v_0)$ *then* $S(l, k) = \ulcorner\varphi(\bar{k})\urcorner$.

(b) *Let the relation* Pf *be defined by* $Pf(n, m)$ *holds iff* $m = \ulcorner\psi\urcorner$ *for some formula* ψ *and* n *is the Gödel number of a deduction from* **PA** *of* ψ*. Then* Pf *is recursive.*

3. Proof of the Incompleteness Theorem

We can now put all of the pieces together to give the proof of Gödel's Incompleteness Theorem. First note that if the relation R is definable in **PA** by the formula θ then θ defines R in $\mathcal{N}$ (since $\mathcal{N} \models$ **PA**).

Theorem 3.1. *(Gödel's Incompleteness Theorem) There is a sentence* σ *of* $\mathcal{L}_{\mathbb{N}}$ *such that* $\mathcal{N} \models \sigma$ *iff* **PA** $\not\models \sigma$*. Thus* $\mathcal{N} \models \sigma$, **PA** $\not\models \sigma$*, and* **PA** $\not\models \neg\sigma$*.*

Proof. Define $R(n, k)$ to be $Pf(n, S(k, k))$. Then R is recursive by Theorem 2.2, and $R(n, k)$ holds iff $k = \ulcorner\varphi\urcorner$ for some formula $\varphi(v_0)$ and n is the Gödel number of some deduction from **PA** of the sentence $\varphi(\bar{k})$. R is definable in **PA** by some formula $\theta(z, v_0)$ and thus $R(n, k)$ holds iff $\mathcal{N} \models \theta(\bar{n}, \bar{k})$.

Now let $\varphi(v_0)$ be $\neg\exists z\theta(z, v_0)$, let $k = \ulcorner\varphi(v_0)\urcorner$, and let σ be $\varphi(\bar{k})$. So σ is $\neg\exists z\theta(z, \bar{k})$ where $k = \ulcorner\neg\exists z\theta(z, v_0)\urcorner$.

Then the following are equivalent:

(1) $\mathcal{N} \models \sigma$
(2) $\mathcal{N} \models \neg\exists z\theta(z, \bar{k})$
(3) $\mathcal{N} \models \neg\theta(\bar{n}, \bar{k})$ for all $n \in \mathbb{N}$
(4) $R(n, k)$ fails for all $n \in \mathbb{N}$
(5) for every $n \in \mathbb{N}$, n is not the Gödel number of a deduction from **PA** of $\varphi(\bar{k})$
(6) **PA** $\not\models \sigma$

Since $\mathcal{N} \models$ **PA** we can't have $\mathcal{N} \not\models \sigma$, and so $\mathcal{N} \models \sigma$ and therefore **PA** $\not\models \sigma$. □

PA could be replaced in Theorem 3.1 by any set Γ of sentences true on $\mathcal{N}$ such that $\{\ulcorner\varphi\urcorner : \varphi \in \Gamma\}$ is recursive. In particular, Incompleteness still holds if we add finitely many new axioms or finitely many new axiom schemes to **PA**. We thus obtain the following:

Theorem 3.2. $\{\ulcorner\theta\urcorner : \theta \in Sn_{L_{\mathbb{N}}}$ *and* $\mathcal{N} \models \theta\}$ *is not recursive.*

This means that there is no "effective" procedure to decide, given any sentence θ, whether or not $\mathcal{N} \models \theta$. One can also show that there is no effective procedure to decide, given any θ, whether or not $\mathbf{PA} \models \theta$ – in fact there is no effective procedure to decide whether or not $\mathbf{Q} \models \theta$. This last fact can be used to derive A. Church's negative solution to the Decision Problem.

Theorem 3.3. $\{\ulcorner\theta\urcorner : \models \theta\}$ *is not recursive, that is, there is no effective procedure to decide, given any* θ*, whether or not* θ *is valid.*