Arithmetic and the Gödel Incompleteness Theorem

11/30/2016

1. Arithmetic

We give now a more or less standard axiomatization for "the natural numbers." That is, we write down axioms which intend to capture the intuitive properties we ascribe to the natural numbers. As we will see below, however, we must be careful in using a phrase such as "the natural numbers." The following axiom scheme is referred to as the Peano axioms for the natural numbers. It consists of a finite set of axioms (sometimes called the Frege subsystem) together with an infinite schema of induction axioms. We present the axiom scheme in the language $\mathcal{L} = \{+, \cdot, E, S, <, 0\}$, but note that the language consisting of just $+$ and $\cdot$ would suffice (the other functions, relations, and 0 can be defined from $+$ and $\cdot$ within the version of the Peano axiom scheme mentioning only the axioms for these two functions). It simplifies things a little to have these extra symbols in the language, however.

Definition 1.1. The Peano axiom scheme is the following set of sentences in the language of number theory.

(Successor Axioms)

$\forall x\ \neg(S(x) \approx \mathbf{0})$.

$\forall x\ \forall y\ (S(x) \approx S(y) \to x \approx y)$

(Order axioms)

$\forall x\ \neg(x < \mathbf{0})$

$\forall x\ \forall y(x < y \vee x \approx y \vee y < x)$.

$\forall x\ \forall y\ (x < S(y) \leftrightarrow x \leqslant y)$

(Addition axioms)

$\forall x\ (x + \mathbf{0} \approx x)$

$\forall x\ \forall y\ x + S(y) \approx S(x + y)$.

(Multiplication axioms)

$\forall x\ x \cdot \mathbf{0} \approx \mathbf{0}$.

$\forall x\ \forall y\ x \cdot S(y) \approx x \cdot y + x$

(Exponentiation axioms)

$\forall x\ xE\mathbf{0} \approx S(\mathbf{0})$

$\forall x\ \forall y\ xES(y) \approx (xEy) \cdot x$

(Induction axioms)

For every formula $\phi(x)$ the axiom $[\phi(\mathbf{0}) \wedge \forall x\ (\phi(x) \to \phi(x+1))] \to \forall x\ \phi(x)$

We let PA denote the Peano axioms scheme, and let F denote PA minus the induction axioms. Thus, F is a finite set of axioms. Note that F just says that the various functions and relations are correctly computed at $S(y)$ from their values at y. As we said above, we could get by, in defining the Peano axioms, with just the addition and multiplication axioms for F.

Remark 1.2. To show that PA $\vdash \forall x\, \varphi(x)$, it does not suffice to show that PA $\vdash \varphi(S^k(0))$ for all $k \in \omega$. That is, $\Gamma = \{\varphi(S^k(0))\}_{n\in\omega}$ does not logically imply $\forall x\, \varphi(x)$. The existence of non-standard models explains why this is not necessarily the case.

As an exercise in using PA we show the following.

Lemma 1.3. *PA proves that $<$ is a strict linear ordering.*

Proof. Assume PA. We first show

Claim 1. $\forall x\, (S(x) \not\approx x)$.

Proof. Let $\varphi(x)$ be the formula $S(x) \not\approx x$. We have $\varphi(0)$ from the first successor axiom. Assume $\varphi(x)$ and we show $\varphi(S(x))$. suppose towards a contradiction that $S(S(x)) \approx S(x)$. Then by the second successor axiom we have $S(x) \approx x$, a contradiction. By the induction schema we have that $\forall x\, \varphi(x)$. □

Claim 2. $\forall x\, \neg(x < x)$.

Proof. Consider $\forall x\, \psi(x)$, where $\psi(x) = \forall y\, (y \leqslant x \rightarrow \neg(y < y))$. We show $\forall x\, \psi(x)$ by induction, that is, using the induction scheme in PA.

First we show $\psi(0)$. If $y \leqslant 0$, then $y \approx 0$ by the first order axiom. Also, $\neg(0 < 0)$ by this axiom. so, $\psi(0)$. Assume now $\psi(x)$, and we show $\psi(S(x))$. Assume towards a contradiction that $y \leqslant S(x)$ and $y < y$. If $y < S(x)$ then $y \leqslant x$ by the third order axiom. Then we are done by $\psi(x)$. So, assume $y = S(x)$. So, we have $S(x) < S(x)$. By the third order-axiom, $S(x) \leqslant x$. By $\psi(x)$ we then have that $\neg(S(x) < S(x))$. By the induction scheme this shows $\forall x\, \psi(x)$, which logically implies $\forall x\, \neg(x < x)$. □

At this point we have shown, assuming PA, thet $<$ is irreflexive, and connectedness is the second order axiom. So, it remains to show transitivity. Consider the statement $\forall x\, \chi(x)$ where

$$\chi(x) = \forall y\, \forall z\, \forall w\, ((w \leqslant x \wedge (z < w) \wedge (y < z)) \rightarrow (y < w)).$$

We show $\forall x\, \chi(x)$ by induction. $\chi(0)$ follows since if $w \leqslant 0$, then $w = 0$, in which case the assumption $z < w$ does not hold (by the first order axiom). So, assume $\chi(x)$ and we show $\chi(S(x))$. So, consider y, z, w with $w \leqslant S(x)$. If $w < S(x)$, then $w \leqslant x$, and we are done by $\chi(x)$. So, assume that $w = S(x)$. Since $z < w$, $z \leqslant x$ by the third order axiom. If $z = x$, then since $y < z$, $y < x$. By the third order-axiom, $y < S(x) = w$. If $z < x$, then $y < x$ by $\chi(x)$. By the third order axiom, $y < S(x) = w$.

By the induction axioms we now have $\forall x\, \chi(x)$, and this immediately implies $\forall x\, \forall y\, \forall z\, ((x < y) \wedge (y < z) \rightarrow (x < z))$ (using $\chi(z)$). □

Exercise 1. Show that PA gives the following "strong induction" principle. Let $\varphi(x)$ be a wff in the language of number theory, with the variable y not occurring in φ. Show that

$$\mathrm{PA} \vdash \varphi(0) \wedge \forall x\, ((\forall y \leqslant x\, \varphi(y)) \rightarrow \varphi(S(x))) \rightarrow \forall x\, \varphi(x).$$

[hint: let $\psi(x) = \forall y\, (y \leqslant x \rightarrow \varphi(y))$. Prove $\forall x\, \psi(x)$ by an induction axiom.]

Exercise 2. Show that PA proves $\forall x\, \forall y\, (x + y \approx y + x)$ and $\forall x\, \forall y\, (x \cdot y \approx y \cdot x)$. [hint: first show by an induction axiom that $\mathrm{PA} \vdash \forall y\, (\forall x\, (S(x) + y \approx S(x + y)))$.]

Exercise 3. Let F' be the version of PA in the language $\mathcal{L}' = \{+, \cdot\, \mathbf{0}\}$ consisting of the addition, multiplication, and zero axioms together with the induction axioms. Define $<$ by $x < y$ iff $\exists z\ (z \not\approx \mathbf{0} \wedge y \approx x + z)$. Show from F' that $<$ satisfies the order axioms.

Exercise 4. Show that PA proves the following stronger induction axiom: let $\varphi(x_1,\dots,x_n)$ be a formula. Then $\forall x_1 \cdots \forall x_{n-1}\,[\varphi(x_1,\dots,x_{n-1},\mathbf{0})\wedge\forall x_n\,(\varphi(x_1,\dots,x_n)\to \varphi(x_1,\dots,x_{n-1},x_n+1)] \to \forall x_n\ \varphi(x_1,\dots,x_n)]$. [hint: let

$$\chi(y)=\forall x_1\cdots\forall x_{n-1}\,(\varphi(\vec{x},y)\vee\exists z<y\,(\varphi(\vec{x},z)\wedge\neg\varphi(\vec{x},S(z)))).$$

Show $\forall y\,\chi(y)$ by induction.]

We introduce a hierarchy of formulas and sets. We say a formula $\phi(x_1,\dots,x_n)$ is Δ_0 if it is built up through the following:

(1) All atomic formulas are Δ_0.
(2) If $\psi(x_1,\dots,x_{n+1})\in\Delta_0$, then so is $\phi=\exists x_{n+1}\leqslant x_j\ \psi(x_1,\dots,x_n,x_j)$ (where $1\leqslant j\leqslant n$) and so is $\phi=\forall x_{n+1}\leqslant x_j\ \psi(x_1,\dots,x_n,x_j)$.

Thus, Δ_0 formulas are the formulas which contain only bounded number quantification. For $n\geqslant 1$ we say $\phi\in\Sigma_n$ if it is of the form $\phi=\exists x_1\ \dots\exists x_n\ \psi$, where $\psi\in\Pi_{n-1}$, and $\phi\in\Pi_n$ if it is of the form $\phi=\forall x_1\ \dots\forall x_n\ \psi$ where $\psi\in\Sigma_{n-1}$ (we interpret Σ_0, Π_0 as being Δ_0). Note that the negation of a Σ_n (or Π_n) formulas is logically equivalent to a Π_n (or Σ_n) formulas.

We say a set $A\subseteq\omega$ is Σ^0_n (or Π^0_n) if there is a Σ_n (resp. Π_n) formula ϕ which defines it, that is, for all $n\in\omega$, $n\in A\leftrightarrow\mathbb{N}\models\phi(n)$ (we interpret $\mathbb{N}$ here as the structure of integers in some metatheory, say a model of ZFC). We say A is Δ^0_n if it is both Σ^0_n and Π^0_n.

2. Recursive Functions

An important point in the proof of the incompleteness theorem is that recursive functions are "representable" within the theory F. Intuitively, the recursive function $f\colon\omega^n\to\omega$ are those which are machine computable. One approach to making this concept precise is to formalize the notion of a "machine." This can, for example, via the concept of a Turing machine. This gives a simple model of a computer/algorithm, and one can then define the collection of recursive functions to be those computable by a Turing maching. Although this approach is intuitive and rather straightforward, it is somewhat tedious to verify that all of the needed functions are computable in this sense. For this reason we take a different approach here, defining the collection of recursive functions in a direct axiomatic manner. Of course, it can be shown that the two definitions (or any of the other standard definitions) define precisely the same class of functions.

Definition 2.1. The collection of (total) recursive functions $f:\omega^n\to\omega$ (for some n) is the smallest collection of functions satisfying the following:

(1) For any $k\in\omega$, the constant function $f(\vec{x})=k$ is recursive. The projection function $f(x_1,\dots,x_n)=x_j$ is recursive, and the successor function $f(n)=n+1$ is recursive.
(2) The addition and multiplication functions $f(n,m)=n+m$, $f(n,m)=n\cdot m$ are recursive.
(3) The class is closed under compostion, that is, if $f(x_1,\dots,x_n)$ is recursive and $g_1(x_1,\dots,x_m),\dots,g_n(x_1,\dots,x_m)$ are recursive then so is $h(x_1,\dots,x_m)=f(g_1(\vec{x}),\dots g_n(\vec{x}))$.

(4) The class is closed under primitive recursion. That is, if $g(\vec{x})$ is recursive, and $h(y, z, \vec{x})$ is recursive, then so is f defined recursively by

$$f(n, \vec{x}) = \begin{cases} g(\vec{x}) & \text{if } n = 0 \\ h(f(n-1, \vec{x}), n-1, \vec{x}) & \text{if } n > 0 \end{cases}$$

(5) The class is closed under minimalization. That is, if $g(\vec{x}, n)$ is recursive and for all $\vec{x}$ there is an n such that $g(\vec{x}, n) = 0$, then the function f defined by $f(\vec{x}) = \mu n\ (g(\vec{x}, n) = 0)$ is recursive. Here "μn" denotes "the least n."

The subclass of functions defined by all but the last (minimalization) clause is called the class of *primitive recursive* functions.

There is a slight redundancy in the above list according to the next exercise.

Exercise 5. Show that multiplication can be defined by a primitive recursion from addition, and addition can be defined by a primitive recursion from the successor function. Show also that exponentiation is primitive recursive. Thus, we don't need (2) in Definition 2.1.

Definition 2.2. We say a relation $R \subseteq \omega^n$ is recursive iff its characteristic function $\chi_R \colon \omega^n \to \{0, 1\}$ is recursive. A relation is said to be primitive recursive if its characteristic function is.

Exercise 6. Show that for any $m \in \mathbb{N}$, $\chi_{\geqslant m}$ is primitive recursive, where $\chi_{\geqslant m}$ is defined by $\chi_{\geqslant m}(n) = \begin{cases} 1 & \text{if } n \geqslant m \\ 0 & \text{otherwise} \end{cases}$. Show also that $\chi_{\leqslant m}$, $\chi_{=m}$ are primitive recursive, where these are defined in the obvious manner. In particular, the function $\text{sg}(n) = \begin{cases} 1 & \text{if } n > 0 \\ 0 & \text{if } n = 0 \end{cases}$ is primitive recursive. [hint: show by induction on m that $\chi_{\geqslant m}$ is primitive recursive. Use a primitive recursion to define $\chi_{\geqslant m}$.]

Exercise 7. Show that the function $a \dot{-} b = \begin{cases} a - b & \text{if } a \geqslant b \\ 0 & \text{otherwise} \end{cases}$ is primitive recursive. Show that $\text{sg}(n) = \begin{cases} 1 & \text{if } n > 0 \\ 0 & \text{if } n = 0 \end{cases}$ is recursive. [hint: first show that $f(n) = n \dot{-} 1$ is primitive recursive.]

Exercise 8. Show that the equality relation on ω is recursive. Show also the relations $<$ and $>$ are primitive recursive.

The next lemma shows we can go back and forth between relations and functions.

Lemma 2.3. *Let $f \colon \omega^n \to \omega$ be a (total) function. Then f is recursive iff its graph $G_f = \{(\vec{a}, b) \colon f(\vec{a}) = b\}$ is.*

Proof. If f is recursive, then G_f is recursive since $\chi_{G_f}(\vec{a}, b) = \chi_{=}(f(\vec{a}), b)$ is recursive using exercise 8. Conversely, if G_f is recursive, so χ_{G_f} is a recursive function, then $f(\vec{a}) = \mu b\ (1 \dot{-} \chi_{G_f}(\vec{a}, b) = 0)$ is recursive. □

Remark 2.4. Lemma 2.3 does not hold for primitive recursive functions. If a function is primitive recursive, then its graph G_f is also primitive recursive by the same proof of Lemma 2.3. However, the converse does not hold; there are recursive functions f with primitive recursive graphs such that f is not a primitive recursive function.

Lemma 2.5. *If R is a recursive relation and f is a recursive function, then $R'(n) \leftrightarrow R(f(n))$ is also recursive. The same is true for primitive recursive.*

Proof. $\chi_{R'}(n) = \chi_R(f(n))$ is a composition of two recursive (or primitive recursive) functions. □

The next lemma shows that definitions by cases preserve recursiveness or primitive recursiveness.

Lemma 2.6. *If $R_1, \dots, R_k$ are recursive (or primitive recursive) relations, and $f_1, \dots, f_k$ are recursive (or primitive recursive) functions, the the function*

$$f(n) = \begin{cases} f_1(n) & \textit{if } R_1(n) \\ f_2(n) & \textit{if } R_2(n) \\ & \vdots \\ f_k(n) & \textit{if } R_k(n) \end{cases}$$

is also recursive (primitive recursive).

Proof. $\chi_f(n) = f_1(n) \cdot \chi_{R_1}(n) + \cdots + f_k(n) \cdot \chi_{R_k}(n)$. □

Lemma 2.7. *The class of recursive relations contains all relations defined by atomic formulas, and is closed under finite unions, intersections, complements, and bounded number quantification. The same is true for the primitive recursive relations. In particular, the primitive recursive relations contain all of the Δ^0_0 relations.*

Proof. If $t = t(x_1, \dots, x_n)$ is a term, then a straightforward induction, using the fact that $+, \cdot, E$ are primitive recursive, shows that the correspoding function $f_t(a_1, \dots, a_n) = t^{\mathbb{N}}(a_1, \dots, a_n)$ is primitive recursive (more precisely we should write $f_t(a_1, \dots, a_n) =$ the unique m such that $\mathbb{N} \models t(S^{a_1}(\mathbf{0}), \dots, S^{a_n}(\mathbf{0})) = S^m(\mathbf{0})$). If φ is atomic of the form $\varphi = (t \approx u)$, then the relation defined by φ is of the form $R_\phi(\vec{a}) \leftrightarrow (f_t(\vec{a}) = f_u(\vec{a}))$ is primitive recursive using exercise 8. The same is true if $\varphi = (t < u)$ as the relation $<$ is also primitive recursive.

If R_1, R_2 are primitive recursive, then so is $R = R_1 \cap R_2$ since $\chi_R(\vec{a}) = \chi_{R_1}(\vec{a}) \cdot \chi_{R_2}(\vec{a})$. The same is easily also true for unions and complements.

Suppose now $R(\vec{a}, n) \leftrightarrow \exists m \leqslant n \; S(\vec{a}, n, m)$ where S is recursive (or primitive recursive). Define $R'(\vec{a}, n, k) \leftrightarrow \exists m \leqslant k \; S(\vec{a}, n, m)$. Then $\chi_{R'}(\vec{a}, n, 0) = \chi_S(\vec{a}, n, 0)$, and for $k > 0$ $\chi_{R'}(\vec{a}, n, k) = \operatorname{sg}(\chi_{R'}(\vec{a}, n, k-1) + \chi_S(\vec{a}, n, k))$, and so $\chi_{R'}$ is recursive (or primitive recursive) by closure under primitive recursion. Since $\chi_R(\vec{a}, n) = \chi_{R'}(\vec{a}, n, n)$ we also have that R is (primitive) recursive. □

The proof of the previous lemma actually shows a bit more.

Lemma 2.8. *If $S(n, m)$ is recursive and $f \colon \omega \to \omega$ is recursive, then $R(n) \leftrightarrow \exists m \leqslant f(n) \; S(n, m)$ is recursive. Likewise, if S and f are primitive recursive, then R is primitive recursive.*

Proof. Define $R'(n, k) \leftrightarrow \exists m \leqslant f(k) \; S(n, m)$. Then $\chi_{R'}(n, k) = \chi_{R''}(n, f(k))$, where $R''(n, k) \leftrightarrow \exists m \leqslant k \; S(n, m)$ is recursive (or primitive recursive) from lemma 2.7. Thus, R' is (primitive) recursive and thus so is $R(n) \leftrightarrow R'(n, n)$. □

The next lemma is the version of Lemma 2.8 for functions.

Lemma 2.9. *Let $S(\vec{a},n,m)$ be a recursive (or primitive recursive relation). Let g be a (total) recursive function (or a primitive recursive function). Let f be defined by*

$$f(\vec{a},n) = \begin{cases} \mu m \leqslant g(\vec{a},n)\, S(\vec{a},n,m) & \textit{if } \exists m \leqslant g(\vec{a},n)\, S(\vec{a},n,m) \\ 0 & \textit{otherwise.} \end{cases}$$

Then f is recursive (respectively, primitive recursive).

Proof. Consider the primitive recursive case, the recursive case being essentially identical. Let $f'(\vec{a},n,k)$ be defined by

$$f'(\vec{a},n,k) = \begin{cases} \mu m \leqslant k\, S(\vec{a},n,m) & \text{if } \exists m \leqslant k\, S(\vec{a},n,m) \\ k+1 & \text{otherwise.} \end{cases}$$

Then f' is primitive recursive as it is defined by a primitive recursion on k: $f'(\vec{a},n,0) = 1 \dot{-} \chi_S(\vec{a},n,0)$, and

$$\begin{aligned} f'(\vec{a},n,k) = \chi_=(f'(\vec{a},&n,k-1),k)\cdot[k\,\chi_S(\vec{a},n,k) + (1 \dot{-} \chi_S(\vec{a},n,k))(k+1)] \\ &+ (1 \dot{-} \chi_=(f'(\vec{a},n,k-1),k))f'(\vec{a},n,k-1). \end{aligned}$$

Finally, $f(\vec{a},n) = \begin{cases} f'(\vec{a},n,g(\vec{a},n)) & \text{if } f'(\vec{a},n,g(\vec{a},n)) \leqslant g(\vec{a},n) \\ 0 & \text{if } f'(\vec{a},n,g(\vec{a},n)) > g(\vec{a},n) \end{cases}$. So f is primitive recursive.

□

We introduce some coding and decoding functions on the integers. Let $p(0) = 2, p(1) = 3, p(2) = 5, \dots$, and in general $p(n) =$ the next prime after $p(n-1)$ (we refer to $p(i)$ as the "i^{th} prime").

Definition 2.10. For $(a_0,\dots,a_k) \in \omega^{<\omega}$ let $\langle a_0,\dots,a_k\rangle = p_0^{a_0+1}p_1^{a_1+1}\cdots p_k^{a_k+1}$. Let Seq $= \{\langle\vec{a}\rangle : \vec{a} \in \omega^{<\omega}\}$ be the set of all codes of finite sequences. For $n = \langle a_0,\dots,a_k\rangle \in$ Seq, let $\mathrm{lh}(n) = k+1$ be the length of the sequence coded by n, and for $n \notin$ Seq, let $\mathrm{lh}(n) = 0$. Define the binary decoding function $(n,i) \to (n)_i$ by $(n)_i = a_i$ if $n = \langle a_0,\dots,a_k\rangle$ codes a sequence of length $> i$, and $(n)_i = 0$ otherwise.

Clearly the map $(\vec{a}) \to \langle\vec{a}\rangle$ is one-to-one on $\omega^{<\omega}$.

Lemma 2.11. *The function p and the set Seq are primitive recursive. For any fixed $k \in \omega$, the function $(a_0,\dots,a_k) \to \langle a_0,\dots,a_k\rangle$ is primitive recursive. The function lh and the decoding function $(n,i) \to (n)_i$ are primitive recursive.*

Proof. The recursive definition $p(i) = \mu n\, [n > p(i-1) \wedge n \text{ is prime }]$ shows that the prime function is recursive. More formally, p is defined by a primnitive recursion by $p(0) = 2$ and for $i > 0$, $p(i) = g(p(i-1))$, where $g(k) = \mu n\, (n > k \wedge n \text{ is prime})$.

To see p is primitive recursive, use the fact that, for example, there is a prime between any n and $2n$, so $p(n) \leqslant 2^n$. So we may write $p(i) = \mu n \leqslant 2^i\, [n > p(i-1) \wedge n \text{ is prime }]$, which shows the p function is primitive recursive using Lemma 2.9. That is, p is defined by a primitive recursion by $p(0) = 2$ and for $i > 0$, $p(i) = g(p(i-1))$, where $g(k) = \mu n \leqslant 2^k\, (n > k \wedge n \text{ is prime})$.

Note that $n \in \text{Seq} \leftrightarrow \forall p \leqslant n\ \forall q \leqslant n\ [(p,q \text{ are prime } \wedge p < q \wedge q|n) \to p|n]$. Since the dividing relation is clearly primitive recursive, this shows Seq is also. For any fixed k, the function $(a_0, \dots, a_k) \to \langle a_0, \dots, a_k \rangle$ is clearly primitive recursive. We have $\mathrm{lh}(n) = \mu l \leqslant n\ [(n \notin \mathrm{Seq} \wedge l = 0) \vee (n \in \mathrm{Seq} \wedge p(l+1) \nmid n)]$. Alternatively, we can write out the last disjunct directly as $(n \in \mathrm{Seq} \wedge s(n, l))]$ where

$$\begin{aligned} s(n,l) \leftrightarrow \exists m \leqslant n^n\ &[(\forall p \leqslant n\ (p \text{ is prime }) \to (p|n \leftrightarrow p|m)) \wedge 2|m \wedge 4 \nmid m \\ &\wedge \forall p, q \leqslant n\ \forall a \leqslant n\ ((p, q \text{ are prime }) \wedge \neg\exists r(p < r < q \wedge r \text{ is prime })) \to \\ &\quad (p^a|n \leftrightarrow q^{a+1}|n) \\ &\wedge \exists p \leqslant n\ (p^l|n \wedge p^{l+1} \nmid n \wedge \forall q \leqslant n(q \text{ is prime }) \wedge q > p \to q \nmid m)] \end{aligned}$$

Thus, $s(n, l)$ asserts that there is an m of the form $m = 2^1 3^2 5^3 \cdots p^l$ and p is the largest prime dividing n.

Finally, for the decoding function we have:

$$\begin{aligned} (n)_i = \mu k \leqslant n\ &[(n \notin \mathrm{Seq} \wedge k = 0) \vee (n \in \mathrm{Seq} \wedge i \geqslant \mathrm{lh}(n) \wedge k = 0) \\ &\vee (n \in \mathrm{Seq} \wedge i < \mathrm{lh}(n) \wedge p(i)^{k+1} \mid n \wedge p(i)^{k+2} \nmid n)]. \end{aligned}$$

□

Exercise 9. Suppose g and h are primitive recursive and f is defined from then by the following *total recursion*: $f(\vec{a}, 0) = g(\vec{a})$, $f(\vec{a}, n{+}1) = h(\langle f(\vec{a}, 0), \dots, f(\vec{a}, n)\rangle, \vec{a}, n)$. Show that f is primitive recursive. [hint: show that the function $f'(\vec{a}, n) = \langle f(\vec{a}, 0), \dots, f(\vec{a}, n)\rangle$ is primitive recursive.

We now sketch a proof of the result that the class of recursive functions coincides with the class of machine computable (total) functions. Since we have not given a precise definition of "machine computable," this sketch will necessarily be a bit vague, but the reader with a particular definition of machine computability in mind will have no trouble making the following argument precise. We will not use this equivalence in what follows.

Theorem 2.12. *the class of recursive function coincides with the class of machine computable (total) functions.*

Proof. It is straightforward to see that any recursive function is machine computable, upon consideration of the various cases. For example, if $f(\vec{a}) = \mu n\ (g(\vec{,}n) = 0)$ where g is recursive (and $\forall \vec{a}\ \exists n\ (g(\vec{a}, n) = 0)$), then from an algorithm computing g we can easily construct an algorithm computing $f(\vec{)}$: given input $\vec{a}$ we go into a loop computing $g(\vec{a}, n)$ for successive values of n until we find one where $g(\vec{a}, n) = 0$, and then we output that n. The other cases are likewise straightforward.

Suppose then that $f(n)$ is machine computable. Given input n, the algorithm will, starting in a certain initial state, move through a successive series series of states (following the algorithm), finally ending in a "halting" state where the output value is given. We can code the successive states of the machine during the computation as a sequence of integers $s = \langle s_0, s_1, \dots, s_k\rangle$ where s_0 is the initial state (which is determined in a simple way from the input value n), and each transition from s_i to s_{i+1} is a valid step according to the algorithm. Finally we require the last integer s_k to be a halting state of the machine, which encodes in some simple way the output value. Thus, we may write

$$\begin{aligned} f(n) = h(\mu s\ [&s \in \mathrm{Seq} \wedge \text{ “}s_0 \text{ codes an initial state correspoding to input } n\text{”} \\ &\wedge \forall i < \mathrm{lh}(s) - 1 \text{ “}(s_i, s_{i+1}) \text{ is a valid transition”} \\ &\wedge \text{ ”}s_{\mathrm{lh}(s)} \text{ codes a halting state }]) \end{aligned}$$

where h is a simple (primitive recursive) function which recovers the output value from the final halting state. Note the essential use of the minimalization operator here; we cannot replace this use by bounded quantification. On the other hand, everything inside the square brackets is primitive recursive. This shows that every recursive function can be defined with only one use of the minimalization operator. □

Corollary 2.13. *Every recursive relation is Δ^0_1 (we will see the converse below).*

Proof. The proof above shows that if R is recursive then it may be written in the form $R(n) \leftrightarrow \exists s\ S(n,s)$, where in fact $S \in \Delta^0_0$ (all quantifiers in the definition of S are bounded by n). □

Finally in this section we mention some of the properties of the pointclass definied earlier. First, we have the following closure properties.

Theorem 2.14. *For $n \geqslant 1$ the Σ^0_1 sets are closed under finite unions and intersections, existential number quantification, bounded universal number quantification (i.e., $\forall n \leqslant m$), and recursive substitution (i.e., if f is total recursive and $R \in \Sigma^0_n$, then so is $R'(n) \leftrightarrow R(f(n))$).*

Similarly, the Π^0_1 sets are closed under finite unions and intersections, universal number quantification, existential number quantification, and recursive substitution.

Proof. We consider the case Σ^0_n. Closure under existential number quantification is obvious. Consider a bounded universal number quantification, say $B(n) \leftrightarrow \forall m \leqslant n\ A(n,m)$ where $A \in \Sigma^0_n$. Thus, $B(n) \leftrightarrow \forall m \leqslant n\ \exists k\ C(n,m,k)$, where $C \in \Pi^0_{n-1}$. Using our coding functions we can then write $B(n) \leftrightarrow \exists l \forall m \leqslant n\ C(n,m,(l)_k)$. By induction this shows B is Σ^0_n. The finite union and intersection cases are easy. Closure under recursive substitution follows from the fact that a recursive substitution into a Δ^0_0 relation results in a recursive relation (from the closure properties of recursive relations), and the fact that a recursive relation is Δ^0_1. □

Theorem 2.15. *A relation $R \subseteq \omega$ is recursive iff $R \in \Delta^0_1$.*

Proof. We have already shown that any recursive relation is Δ^0_1. Suppose now that $R \in \Delta^0_1$. Say $R(n) \leftrightarrow \exists m\ P(n,m)$, and $\neg R(n) \leftrightarrow \exists m\ Q(n,m)$ where R, Q are recursive (since both R and its complement are Σ^0_1 by definition of R being Δ^0_1). Let $f(n) = \mu m\ [P(n,m) \vee Q(n,m)]$, so f is recursive. Then $R(n) \leftrightarrow P(n, f(n))$, so R is recursive. □

3. Representability in Arithmetic

An important point in the proof of the imcompleteness theorem is that all recursive functions and relations are “representable” in arithmetic, and even in the finite subsystem F defined in §1. We define this notion here and prove the representability of the recursive functions and relations. The proof of the incompleteness theorem itself is given in the next section.

Definition 3.1. We say a relation $R \subseteq \omega$ is representable in F if there is a formula $\phi(x)$ (in the language of number theory) such that for all $n \in \omega$, if $R(n)$ then $\mathrm{F} \vdash \phi(S^n(\mathbf{0}))$, and if $\neg R(n)$ then $\mathrm{F} \vdash \neg\phi(S^n(\mathbf{0}))$.

We say a (total) function $f \colon \omega \to \omega$ is representable iff its graph G_f is. That is, there is a formula $\phi(x, y)$ such that if $f(n) = m$ then $\mathrm{F} \vdash \phi(S^n(\mathbf{0}), S^m(\mathbf{0}))$, and if $f(n) \neq m$ then $\mathrm{F} \vdash \neg\phi(S^n(\mathbf{0}), S^m(\mathbf{0}))$.

Exercise 10. Show that every representable relation or function is recursive.

An important technical point is that representability of functions coincides with a seemingly stronger concept which we now define.

Definition 3.2. A (total) function $f \colon \omega \to \omega$ is *strongly representable* if there is a formula $\phi(x, y)$ such that for all n, $\mathrm{F} \vdash \phi(S^n(\mathbf{0}), S^{f(n)}(\mathbf{0}))$ and also $\mathrm{F} \vdash [\forall z\ (\phi(S^n(\mathbf{0}), z) \to z \approx S^{f(n)}(\mathbf{0}))]$.

Clearly strong representability implies representability. We show that the converse holds, but first a simple technical lemma.

Lemma 3.3. *For any $n \in \omega$, $F \vdash \forall z\ (z \leqslant S^n(0) \to z \approx \mathbf{0} \vee z \approx S(\mathbf{0}) \vee \cdots \vee z \approx S^n(\mathbf{0}))$.*

Proof. By induction on n. The result holds for $n = 0$ since $\mathrm{F} \vdash (z \leqslant \mathbf{0} \to z \approx \mathbf{0})$ as $\mathrm{F} \vdash \neg(z < \mathbf{0})$. Asssume the result holds for n and assume $z \leqslant S^{n+1}(\mathbf{0})$. If $z < S^{n+1}(\mathbf{0}) = S(S^n(\mathbf{0}))$, then F proves that $z \leqslant S^n(\mathbf{0})$, in which case by induction F proves $z \approx \mathbf{0} \vee \cdots \vee z \approx S^n(\mathbf{0})$. Thus, $\mathrm{F} \vdash z \approx \mathbf{0} \vee \cdots \vee z \approx S^{n+1}(\mathbf{0})$. □

Lemma 3.4. *If f is representable, then it is strongly representable.*

Proof. Suppose $\phi(x, y)$ represents $f \colon \omega \to \omega$. Define $\psi(x, y) = [\phi(x, y) \wedge \forall w < y\ \neg\phi(x, w)]$. We claim that ψ strongly represents f. Fix $n \in \omega$, and let $m = f(n)$. By asumption, $\mathrm{F} \vdash \phi(S^n(\mathbf{0}), S^m(\mathbf{0}))$. We must show that $\mathrm{F} \vdash \forall w < S^m(\mathbf{0})\ \neg\phi(S^n(\mathbf{0}), w)$. Work within F, and assume $w < S^m(\mathbf{0})$. From lemma 3.3 we can deduce $(w \approx \mathbf{0} \vee w \approx S(\mathbf{0}) \vee \cdots \vee w \approx S^{m-1}(\mathbf{0}))$. Since ϕ represents f we have that $\mathrm{F} \vdash \neg\phi(S^n(\mathbf{0}), \mathbf{0}), \ldots, \mathrm{F} \vdash \neg\phi(S^n(\mathbf{0}), S^{m-1}(\mathbf{0}))$. From these two statements it follows that $\mathrm{F} \vdash \forall w < S^m(\mathbf{0})\ \neg\phi(S^n(\mathbf{0}), w)$. Thus, $\mathrm{F} \vdash \psi(S^n(\mathbf{0}), S^m(\mathbf{0}))$.

Working within F, assume now $\psi(S^n(\mathbf{0}), z)$, so $\forall w < z\ \neg\phi(S^n(\mathbf{0}), w)$. Since $\mathrm{F} \vdash \phi(S^n(\mathbf{0}), S^m(\mathbf{0}))$, we may deduce that $z \leqslant S^m(\mathbf{0})$ (we use that fact that $\mathrm{F} \vdash (z < S^m(\mathbf{0}) \vee z \approx S^n(\mathbf{0}) \vee z > S^m(\mathbf{0})))$. So we may deduce $z \approx \mathbf{0} \vee \cdots \vee z \approx S^m(\mathbf{0})$. Since $\mathrm{F} \vdash \neg\phi(S^n(\mathbf{0}), \mathbf{0}), \ldots, \mathrm{F} \vdash \neg\phi(S^n(\mathbf{0}), S^{m-1}(\mathbf{0}))$, we may deduce $z \approx S^m(\mathbf{0})$. □

The next theorem is the result we need on representability.

Theorem 3.5. *Every recursive relation and function is representable in F.*

Lemma 3.6. *Let t be a* closed term*, that is, a term containing no free variables. Then there is an $n \in \omega$ such that $F \vdash t \approx S^n(\mathbf{0})$.*

Proof. By induction on the term t. If $t = \mathbf{0}$ this is trivial. If $t = S(u)$ this is also trivial as $\mathrm{F} \vdash u \approx S^n(\mathbf{0})$ for some n by induction, and $S(S^n(\mathbf{0})) = S^{n+1}(\mathbf{0})$. If $t = u + v$, then by induction it suffices to show that $\mathrm{F} \vdash S^n(\mathbf{0}) + S^m(\mathbf{0}) \approx S^{n+m}(\mathbf{0})$. This, in turn, is proved by induction on m with the inductive step given by $\mathrm{F} \vdash S^n(\mathbf{0}) + S(S^{m-1}(\mathbf{0})) \approx S(S^n(\mathbf{0}) + S^{m-1}(\mathbf{0})) \approx S(S^{n+m-1}(\mathbf{0})) = S^{n+m}(\mathbf{0})$. The result for terms of the form $t = u \cdot v$ follows similarly from $\mathrm{F} \vdash S^n(\mathbf{0}) \cdot S^m(\mathbf{0}) \approx$

$S^{n\cdot m}(\mathbf{0})$ which is proved by induction, using the result for addition. The result for exponentiation is similar. □

Lemma 3.7. *If ϕ is quantifier free then the relation R defined by ϕ is representable in F.*

Proof. It suffices to prove this for atomic formulas. For this it suffices to show that if t, u are closed terms then either $\mathrm{F} \vdash t < u$ or $\mathrm{F} \vdash \neg(t < u)$, and likewise $\mathrm{F} \vdash (t \approx u)$ or $\mathrm{F} \vdash \neg(t \approx u)$. We consider first the the $\approx$ case. By the lemma there are $n, m \in \omega$ such that $\mathrm{F} \vdash t \approx S^n(\mathbf{0})$ and $\mathrm{F} \vdash u \approx S^m(\mathbf{0})$. It suffices to show that if $n = m$ then $\mathrm{F} \vdash S^n(\mathbf{0}) \approx S^m(\mathbf{0})$ and if $n \neq m$ then $fa \vdash \neg(S^n(\mathbf{0}) \approx S^m(\mathbf{0}))$. The first is trivial. For the second, we prove by induction on $\min\{n, m\}$ that if $n \neq m$ then $\mathrm{F} \vdash \neg(S^n(\mathbf{0}) \approx S^m(\mathbf{0}))$. If $\min\{n, m\} = 0$, this follows from the first successor axiom. Otherwise, by induction $\mathrm{F} \vdash \neg(S^{n-1}(\mathbf{0}) \approx S^{m-1}(\mathbf{0}))$. The second successor axiom then gives that $\mathrm{F} \vdash \neg(S^n(\mathbf{0}) \approx S^m(\mathbf{0}))$.

We now consider the $<$ case. Again by the lemma there are $n, m \in \omega$ such that $\mathrm{F} \vdash t \approx S^n(\mathbf{0})$ and $\mathrm{F} \vdash S^m(\mathbf{0})$. It suffices to know that if $n < m$ then $\mathrm{F} \vdash S^n(\mathbf{0}) < S^m(\mathbf{0})$ and otherwise $\mathrm{F} \vdash \neg(S^n(\mathbf{0}) < S^m(\mathbf{0}))$. First we show by induction on $m > n$ that $\mathrm{F} \vdash S^n(\mathbf{0}) < S^m(\mathbf{0})$. For $m = n+1$, $\mathrm{F} \vdash S^n(\mathbf{0}) < S^{n+1}(\mathbf{0}) = S(S^n(\mathbf{0}))$ follows from the axiom of F $\forall x\ \forall y\ (x < S(y) \leftrightarrow x \leqslant y)$ which implies $\forall x\ (x < S(x))$. Assuming the result is true for m, $\mathrm{F} \vdash (S^n(\mathbf{0}) < S^m(\mathbf{0}))$, the same axiom then shows that $\mathrm{F} \vdash (S^n(\mathbf{0}) < S(S^m(\mathbf{0})) = S^{m+1}(\mathbf{0}))$. Assume now that $n \geqslant m$. Working in F, asssume towards a contradiction that $S^n(\mathbf{0}) < S^m(\mathbf{0})$. From lemma 3.3 we have $\mathrm{F} \vdash S^n(\mathbf{0}) \approx \mathbf{0} \vee \cdots \vee S^n(\mathbf{0}) \approx S^{m-1}(\mathbf{0})$. However, from the equality case we know that $\mathrm{F} \vdash \neg(S^n(\mathbf{0}) \approx \mathbf{0}), \ldots, \mathrm{F} \vdash \neg(S^n(\mathbf{0}) \approx S^{m-1}(\mathbf{0}))$. This is a contradiction.

Note that in all cases we have shown that the same formula ϕ represents the relation R defined by ϕ in $\mathbb{N}$. □

Lemma 3.8. *If $\phi \in \Delta_0$ then the relation R defined by ϕ is representable in F. In fact, the same formula ϕ represents R.*

Proof. It suffices to show that if $R(x, y)$ is representable by $\phi(x, y)$, then $S(n) \leftrightarrow \exists m \leqslant n\ R(n, m)$ is representable. Let $\psi(x) = \exists y\ (y \leqslant x \wedge \phi(x, y))$. Let $n \in \omega$, and first suppose $S(n)$. Thus there is an $m \leqslant n$ such that $R(n, m)$. Thus, $\mathrm{F} \vdash \phi(S^n(\mathbf{0}), S^m(\mathbf{0}))$. From lemma 3.7, $\mathrm{F} \vdash S^n(\mathbf{0}) \leqslant S^m(\mathbf{0})$. Hence, $\mathrm{F} \vdash \exists y\ (y \leqslant S^n(\mathbf{0}) \wedge \phi(S^n(\mathbf{0}), y)$, that is, $\mathrm{F} \vdash \psi(S^n(\mathbf{0}))$.

Assume now that $n \in \omega$ and $\neg S(n)$, hence for all $m \leqslant n$ we have $\neg R(n, m)$. From lemma 3.3, $\mathrm{F} \vdash \forall y\ (y \leqslant S^n(\mathbf{0}) \rightarrow y \approx \mathbf{0} \vee \cdots \vee y \approx S^n(\mathbf{0}))$. Since ϕ represents R we also have $\mathrm{F} \vdash \neg\phi(S^n(\mathbf{0}), \mathbf{0}), \ldots, \mathrm{F} \vdash \neg\phi(S^n(\mathbf{0}), S^n(\mathbf{0}))$. These two statements logically imply $\forall y\ (y \leqslant S^n(\mathbf{0}) \rightarrow \neg(\phi(S^n(\mathbf{0}), y)$. Thus, $\mathrm{F} \vdash \neg\exists y \leqslant S^n(\mathbf{0})\ (\phi(S^n(\mathbf{0}), y))$, that is, $\mathrm{F} \vdash \neg\psi(S^n(\mathbf{0}))$ and we are done. □

Corollary 3.9. *For any $k \in \omega$, the k-ary coding function $(a_0, \ldots, a_{k-1}) \to \langle a_0, \ldots, a_{k-1}\rangle$ is representable. Also for any $j < k$, there is a reprentable decoding function $n \to (n)_j$ such that for any n of the form $n = \langle a_0, \ldots, a_{k-1}\rangle$, $(n)_j = a_j$.*

Remark 3.10. The full coding and decoding functions of definition 2.10 are also representable as we will see, but this weaker result suffices for what we need below.

Proof. For fixed k, the graph of the k-ary coding function is defined by a quantifier free formula: $y = \langle x_0, \ldots, x_{k-1}\rangle \leftrightarrow y \approx (2Ex_0) \cdots (p_{k-1}Ex_{k_1})$. We define the decoding function $x \to (x)_j$ to be the function represented by the Δ_0 formula

$$\begin{aligned}\phi(x,y) =&[\exists w_0 \leqslant x \cdots \exists w_{k-1} \leqslant x\ (y \approx (2Ew_0) \cdots (p_{j-1}Ew_{j-1}) \cdots (p_jE(y+1)) \\ &\quad \cdots (p_{k-1}Ew_{k-1})) \\ &\vee \neg\exists w_0 \leqslant x \cdots \exists w_{k-1} \leqslant x\ (y \approx (2Ew_0) \cdots (p_{k-1}Ew_{k_1})) \wedge y \approx \mathbf{0}]\end{aligned}$$

□

The next theorem is the result we need on the representability of recursive functions.

Theorem 3.11. *Every total recursive function is representable in F.*

To prove this theorem, we must consider the cases in the recursive definition of recursive function. For the ground case, we need to know that the base functions $f(n) = n+1$, $f(n,m) = n+m$, $f(n,m) = n \cdot m$, $f(a_1, \dots, a_k) = a_j$, and constant functions are representable. In all cases this follows from lemma 3.7. For example, consider $f(n,m) = n+m$. The graph is represented by the formula $\phi(x,y,z) = (x+y \approx z)$. Likewise, the graph of $f(n) = n+1$ is represented by $\phi(x,y) = (y \approx S(x))$. The constant function $f(x) = a$ is represented by the formula $\phi(x,y) = (y \approx S^a(\mathbf{0}))$. The projection function $f(a_1, \dots, a_n) = a_j$ is represented by $\phi(x_1, \dots, x_n, y) = (y \approx x_j)$.

The next lemma handles the composition case.

Lemma 3.12. *If $f, g\colon \omega \to \omega$ are representable functions then so is $h = g \circ f$.*

Proof. Let $\phi(x,y)$ represent f and $\psi(x,y)$ represent g. By lemma 3.4, we may assume that ϕ strongly represents f. Let $\sigma(x,y) = [\exists z\ \phi(x,z) \wedge \psi(z,y)]$. Let $n \in \omega$ and $m = h(n)$. Let $k = f(n)$, so $m = g(k)$. By assumption F $\vdash \phi(S^n(\mathbf{0}), S^k(\mathbf{0}))$ and F $\vdash \psi(S^k(\mathbf{0}), S^m(\mathbf{0}))$. These two statements logically imply $\exists z\ [\phi(S^n(\mathbf{0}), z) \wedge \psi(z, S^m(\mathbf{0})]$, and so F $\vdash \sigma(S^n(\mathbf{0}), S^m(\mathbf{0}))$.

Suppose next that $r \neq h(n) = m$, and we must show that F $\vdash \neg\sigma(S^n(\mathbf{0}), S^r(\mathbf{0}))$, that is, F $\vdash \forall z\ (\phi(S^n(\mathbf{0}), z) \to \neg\psi(z, S^r(\mathbf{0})))$. By strong representability, F $\vdash \forall z\ (\phi(S^n(\mathbf{0}), z) \to z \approx S^k(\mathbf{0}))$. By representability of ψ we have F $\vdash \neg\psi(S^k(\mathbf{0}), S^r(\mathbf{0}))$. These statement logically imply $\forall z\ (\phi(S^n(\mathbf{0}), z) \to \neg\psi(z, S^k(\mathbf{0})))$. Thus, F $\vdash \neg\sigma(S^n(\mathbf{0}), S^r(\mathbf{0}))$. □

The next lemma handles the minimalization case.

Lemma 3.13. *Suppose $g\colon \omega^2 \to \omega$ is recursive and $\forall n\ \exists m\ g(n,m) = 0$. Then $f(n) = \mu m\ g(n,m) = 0$ is representable in F.*

Proof. Let $\phi(x,y,z)$ represent g. Let $\psi(x,y) = [\phi(x,y,\mathbf{0}) \wedge \forall w < y\ \neg\phi(x,w,\mathbf{0})]$. Let $n \in \omega$ and let $m = f(n)$, so $g(n,m) = 0$ and for all $k < m$ we have $g(n,k) \neq 0$. Since ϕ represents g we have F $\vdash \phi(S^n(\mathbf{0}), S^m(\mathbf{0}), \mathbf{0})$. From lemma 3.3, F $\vdash \forall w\ (w < S^m(\mathbf{0}) \to w \approx \mathbf{0} \vee \cdots \vee w \approx S^{m-1}(\mathbf{0}))$. Since ϕ represents g we also have F $\vdash \neg\phi(S^n(\mathbf{0}), \mathbf{0}, \mathbf{0})$, $\dots$, F $\vdash \neg\phi(S^n(\mathbf{0}), S^{m-1}(\mathbf{0}), \mathbf{0})$. These statement logically imply $\forall w\ (w < S^m(\mathbf{0}) \to \neg\phi(S^n(\mathbf{0}), w, \mathbf{0}))$. Hence, F $\vdash \psi(S^n(\mathbf{0}), S^m(\mathbf{0}))$.

Assume now that $r \neq m = f(n)$ and we must show that F $\vdash \neg\psi(S^n(\mathbf{0}), S^r(\mathbf{0}))$. If $g(n,r) \neq 0$, then F $\vdash \neg\phi(S^n(\mathbf{0}), S^r(\mathbf{0}), \mathbf{0})$ which logically implies $\neg\psi(S^n(\mathbf{0}), S^r(\mathbf{0}))$. So assume $g(n,r) = 0$, in which case we must have $r > m$. Since $m < r$, from lemma 3.7 we have F $\vdash S^m(\mathbf{0}) < S^r(\mathbf{0})$. Also, F $\vdash \phi(S^n(\mathbf{0}), S^m(\mathbf{0}), \mathbf{0})$, and these statements logically imply $\exists w < S^r(\mathbf{0})\ \phi(S^n(\mathbf{0}), w, \mathbf{0})$. This logically implies $\neg\psi(S^n(\mathbf{0}), S^r(\mathbf{0}))$, and so F $\vdash \neg\psi(S^n(\mathbf{0}), S^r(\mathbf{0}))$. □

The next lemma handles the primitive recursion case, and completes the proof of theorem 3.11.

Lemma 3.14. *Suppose $g(\vec{a})$ and $h(x, n, \vec{a})$ are representable. Then the function f defined by the primitive recursion $f(0, \vec{a}) = g(\vec{a})$, $f(n + 1, \vec{a}) = h(f(n, \vec{a}), n, \vec{a})$ is also representable.*

Proof. Let $\phi(\vec{z}, y)$ and $\psi(x, y, \vec{z}, w)$ represents g, h. We use the same trick as in the proof of lemma 2.11, this time verifying $m = f(n, \vec{a})$ by searching for an integer of the form $w = 2^{\langle 0, f(0,\vec{a})\rangle} \cdot 3^{\langle 1, f(1,\vec{a})\rangle} \cdots {p_n}^{\langle n, f(n,\vec{a})\rangle}$ as a witness. More precisely, define

$$\begin{aligned}\chi(x, \vec{z}, w) =& \forall p, q \leqslant w\ \{(p, q \text{ are prime } \wedge (p < q) \wedge \neg\exists r(p < r < q \wedge r \text{ is prime })\\ &\wedge q \mid w) \to (p \mid w) \wedge \forall t_1, u_1, v_1, t_2, u_2, v_2 \leqslant w\ ((v_1 \approx \langle t_1, u_1\rangle\\ &\wedge v_2 \approx \langle t_2, u_2\rangle) \wedge pEv_1 \mid w \wedge pE(S(v_1)) \nmid w \wedge qEv_2 \mid w \wedge\\ &qE(S(v_2)) \nmid w \to t_2 \approx S(t_1) \wedge \psi(u_1, t_1, \vec{z}, u_2))\} \wedge\\ &\exists t \leqslant w\ \exists u \leqslant w\ \{\phi(\vec{z}, t) \wedge u = \langle 0, t\rangle \wedge 2Eu \mid w \wedge 2E(S(u)) \nmid u\} \wedge\\ &\exists p, u, y \leqslant w\ (p \text{ is prime } \wedge u \approx \langle x, y\rangle \wedge pEu \mid w \wedge pE(S(u)) \nmid w)\end{aligned}$$

Here, of course, we substitute the appropriate Δ_0 formula for "p is prime" and "$v \approx \langle t, u\rangle$," etc. Note that $\chi \in \Delta_0$.

Let then

$$\begin{aligned}\sigma(x, \vec{z}, y) = \exists w\ [&\chi(x, \vec{z}, w) \wedge \forall w' < w(\neg\chi(x, \vec{z}, w'))\\ &\wedge \exists p, u \leqslant w\ (p \text{ is prime } \wedge u \approx \langle x, y\rangle \wedge pEu \mid w \wedge pE(S(u)) \nmid w)]\end{aligned}$$

We claim that σ represents f. Suppose that $f(n, \vec{a}) = m$. Let $w = 2^{\langle 0, f(0,\vec{a})\rangle} \cdot 3^{\langle 1, f(1,\vec{a})\rangle} \cdots {p_n}^{\langle n, f(n,\vec{a})\rangle}$. Clearly $\chi(n, \vec{a}, w)$ holds in $\mathbb{N}$, and w is the least integer such that $\chi(n, \vec{a}, w)$ holds. Since $\chi \in \Delta_0$, from lemma 3.8 we have that

$$\mathrm{F} \vdash \chi(S^n(\mathbf{0}), S^{\vec{a}}(\mathbf{0}), S^w(\mathbf{0})) \wedge \forall w' < S^w(\mathbf{0})\ \neg\chi(S^n(\mathbf{0}), S^{\vec{a}}(\mathbf{0}), w').$$

The last conjunct in the definition of σ holds for $x = n$, $y = m$, and since this conjunct is Δ_0, F proves the corresponding formula at $S^n(\mathbf{0})$, $S^m(\mathbf{0})$. Hence $\mathrm{F} \vdash \sigma(S^n(\mathbf{0}), S^{\vec{a}}(\mathbf{0}), S^m(\mathbf{0}))$.

Suppose next that $r \neq m = f(n, \vec{a})$. We must show that $\mathrm{F} \vdash \neg\sigma(S^n(\mathbf{0}), S^{\vec{a}}(\mathbf{0}), S^r(\mathbf{0}))$. Let $\tau(x, \vec{z}, y, w)$ be the subformula of σ in square brackets. It is enough to show that F together with $\tau(S^n(\mathbf{0}), S^{\vec{a}}(\mathbf{0}), S^r(\mathbf{0}), w)$ is inconsistent [for then we would have $\mathrm{F} \vdash \neg\tau(S^n(\mathbf{0}), S^{\vec{a}}(\mathbf{0}), S^r(\mathbf{0}), w)$ and hence $\mathrm{F} \vdash \forall w\ \neg\tau(S^n(\mathbf{0}), S^{\vec{a}}(\mathbf{0}), S^r(\mathbf{0}), w)$ which is logicaly equivalent to $\neg\sigma(S^n(\mathbf{0}), S^{\vec{a}}(\mathbf{0}), S^r(\mathbf{0}))$]. Let $w_0 = 2^{\langle 0, f(0,\vec{a})\rangle} \cdot 3^{\langle 1, f(1,\vec{a})\rangle} \cdots {p_n}^{\langle n, f(n,\vec{a})\rangle}$. Thus, $\mathrm{F} \vdash \chi(S^n(\mathbf{0}), S^{\vec{a}}(\mathbf{0}), S^{w_0}(\mathbf{0}))$. Now $\tau(S^n(\mathbf{0}), S^{\vec{a}}(\mathbf{0}), S^r(\mathbf{0}), w)$ logically implies $\chi(S^n(\mathbf{0}), S^{\vec{a}}(\mathbf{0}), w)$ as well as $\forall w' < w\ \neg\chi(S^n(\mathbf{0}), S^{\vec{a}}(\mathbf{0}), w')$. From the order axioms, $\mathrm{F} \vdash (w < S^{w_0}(\mathbf{0}) \vee w \approx S^{w_0}(\mathbf{0}) \vee w > S^{w_0}(\mathbf{0}))$. The latter case is clearly a contradiction. From lemma 3.3 and lemma 3.8 $\mathrm{F} \vdash \forall w < S^{w_0}(\mathbf{0})\ \neg\chi(S^n(\mathbf{0}), S^{\vec{a}}(\mathbf{0}), w)$ which shows that the assumption $w < S^{w_0}(\mathbf{0})$ also leads to a contradiction. Thus we may deduce (from F and $\tau(S^n(\mathbf{0}), S^{\vec{a}}(\mathbf{0}), S^r(\mathbf{0}), w)$) that $w \approx S^{w_0}(\mathbf{0})$. Thus we may deduce $\tau(S^n(\mathbf{0}), S^{\vec{a}}(\mathbf{0}), S^r(\mathbf{0}), S^{w_0}(\mathbf{0}))$, which is a contradiction since $\mathrm{F} \vdash \neg\tau(S^n(\mathbf{0}), S^{\vec{a}}(\mathbf{0}), S^r(\mathbf{0}), S^{w_0}(\mathbf{0}))$ from lemma 3.8.

□

4. Incompleteness

In this section we prove several versions of the Gödel incompleteness theorem. First we define a coding of the formulas of number theory into the integers. Fix a bijection π between the finitely many symbols of the language (including the logical symbols) excluding the (infinitely many) variable symbols and the set $\{0, 1, \dots, n_0 - 1\}$. Extend π to the variables by $\pi(x_k) = n_0 + k$. Then π is a bijection between the logical symbols and the integers, and the relation $R(a, b) \leftrightarrow \pi(x_a) = b$ is clearly recursive.

Definition 4.1. If $\phi = s_0 s_1, \dots, s_k$ is a string of symbols in the language of number theory, then the *Gödel code* of ϕ is defined by $\#(\phi) = \langle \pi(s_0), \dots, \pi(s_k) \rangle \in \omega$.

We will use in the folowing arguments the fact that certain relations and (total) functions on the integers are recursive. In fact, all of the relations and functions we need are primitive recursive. These facts can be easily checked from the closure properties of recursive functions of § 2; we leave the details to the reader.

The next result is the key technical lemma for the incompleteness results. It says, in effect, that we may construct self-referential formulas. The formulas attempt to refer to themselves by referring to the Gödel codes of themselves.

Lemma 4.2. *Let $\theta(x)$ be a formula in the language of number theory with one free variable x. Then there is a sentence σ (in the language of number theory) such that $F \vdash (\sigma \leftrightarrow \theta(S^{\#\sigma}(\mathbf{0})))$.*

Proof. Let $f \colon \omega \to \omega$ be the primitive recursive function defined as follows. If n is the code of a formula ψ with one free variable, then $f(n)$ is the code of the sentence $\psi(S^{\#\psi}(\mathbf{0}))$. Otherwise, let $f(n) = 0$. Let $\rho(x, y)$ strongly represent f in F. Let τ be the formula

$$\tau = \exists x_1 \ (\rho(x_0, x_1) \wedge \theta(x_1)).$$

Note that τ has one free variable, x_0. Let $n = \#\tau$. Let $\sigma = \tau(S^{\#\tau}(\mathbf{0}))$. Let $m = f(n)$, which is the code for $\tau(S^{\#\tau}(\mathbf{0})) = \sigma$. We show that σ works. Working within F, first assume σ. Thus, $\exists x_1 \ (\rho(S^{\#\tau}(\mathbf{0}), x_1) \wedge \theta(x_1))$. By strong representability, $\mathrm{F} \vdash \forall x_1 \ (\rho(S^{\#\tau}(\mathbf{0}), x_1) \to x_1 \approx S^m(\mathbf{0}))$. These two sentences logically imply $\theta(S^m(\mathbf{0}))$, that is, $\theta(S^{\#\sigma}(\mathbf{0}))$.

Assume next $\theta(S^{\#\sigma}(\mathbf{0}))$, that is, $\theta(S^m(\mathbf{0}))$. Since $\mathrm{F} \vdash \rho(S^n(\mathbf{0}), S^m(\mathbf{0}))$ by representability, we may deduce $\exists x_1 \ (\rho(S^n(\mathbf{0}), x_1) \wedge \theta(x_1))$. Thus, we may deduce $\tau(S^n(\mathbf{0}))$, that is, σ. □

We now state the first version of the incompleteness theorem. We call a set of sentences T recursive if $\{\#\phi \colon \phi \in T\}$ is recursive. The reader will note that the sentence σ constructed in the following proof is a formalization of the statement "this sentence is not provable."

Theorem 4.3. *Let T be a consistent, recursive set of sentences in the language of number theory which contains F. Then T is incomplete, that is, there is a sentence σ such that $T \nvdash \sigma$ and $T \nvdash \neg\sigma$.*

Proof. Towards a contradiction assume that T is complete. Let $R = \{\#\phi \colon T \vdash \phi\}$. We claim that R is recursive. This is because we may check if $n \in R$ by enumerating all possible deductions from T and checking at each step if it is a deduction from T of either ϕ (the formula with code n) or a deduction of $\neg\phi$. We output a 1 if for the least such deduction we encounter it is a deduction of ϕ. Checking if an integer codes a valid deduction from T is recursive, using the assumption that T is recursive. This algorithm will always terminate by our completeness assumption. The answer will be correct as T is consistent.

Let θ represent $\neg R$ in F. Let σ be the sentence of lemma 4.2 applied to θ. Thus, F, and hence T proves the statement

$$\sigma \leftrightarrow \theta(S^{\#\sigma}(\mathbf{0})).$$

Let $n = \#\sigma$. If $R(n)$, then $\mathrm{F} \vdash \neg\theta(S^n(\mathbf{0}))$, and so $T \vdash \neg\sigma$. Thus, $\neg R(n)$, a contradiction. If $\neg R(n)$, then $\mathrm{F} \vdash \theta(S^n(\mathbf{0}))$, and so $T \vdash \sigma$. Hence $R(n)$, a contradiction. □

Theorem 4.3 was proved by contradiction, and thus does not actually produce a concrete sentence σ which is independent of T. With a little extra argument we can do this. First we give the argument due to Gödel which shows this under a slightly stronger hypothesis.

Definition 4.4. We say T is ω-consistent if there is no formula $\phi(x)$ such that for all $n \in \omega$, $T \vdash \neg\phi(S^n(\mathbf{0}))$ but $T \vdash \exists x\ \phi(x)$.

Of course, an ω-consistent theory is consistent, but the converse is not true. An ω-inconsistent theory is one that has no standard model.

For T a recursive set of sentences in the language of number theory, let R_T be the relation defined by $R_T(a, b)$ iff b is the code of a deduction from T of the formula with code a. R_T is clearly recursive. Let $\theta(x, y)$ strongly represent R in F. Let $\tau(x) = \neg\exists y\ \theta(x, y)$, and let $\sigma_1 = \sigma_1(T)$ be the sentence such that $\mathrm{F} \vdash \sigma_1(T) \leftrightarrow \tau(S^{\#\sigma_1(T)}(\mathbf{0}))$ from lemma 4.2.

Theorem 4.5. *Let T be an ω-consistent, recursive set of sentences in the language of number theory which contains F. Then $T \nvdash \sigma_1$ and $T \nvdash \neg\sigma_1$.*

Proof. The proof is similar to theorem 4.3. Assume first that $T \vdash \sigma_1$. Let $n = \#\sigma_1$. Let m code a deduction of σ_1 from T. Thus, $T \vdash \theta(S^n(\mathbf{0}), S^m(\mathbf{0}))$ (in the notation above). This logically implies $\neg\tau(S^n(\mathbf{0}))$, and thus $T \vdash \neg\sigma_1$. This contradicts the assumption that T is consistent (note: this case only used the consistency of T).

Assume next that $T \vdash \neg\sigma_1$. Thus, $T \vdash \neg\tau(S^n(\mathbf{0}))$, and so $T \vdash \exists y\ \theta(S^n(\mathbf{0}), y)$. Since $T \nvdash \sigma_1$ from the previous paragraph, we know that for all $m \in \omega$ that $\neg R_T(n, m)$, and hence $T \vdash \neg\theta(S^n(\mathbf{0}), S^m(\mathbf{0}))$. This contradicts the ω-consistency of T. □

The extra hypothesis of ω-consistency, though minor, is slightly annoying. An improvement of theorem 4.5, due to Rosser, shows that it is actually unnecessary. Let $R_T(a, b)$ and $\theta(x, y)$ be as above. Let $g \colon \omega \to \omega$ be a recursive function such that if a is the code of ϕ, then $g(a)$ is the code of $\neg\phi$. Let $\rho(x, y)$ strongly represent g. Let $\tau(x)$ be the formula

$$\tau = \forall y\ (\theta(x, y) \to \exists z < y\ \exists w\ (\rho(x, w) \wedge \theta(w, z)))$$

Let $\sigma_2 = \sigma_2(T)$ be the sentence from lemma 4.2 for this τ.

Theorem 4.6. *Let T be a consistent, recursive set of sentences in the language of number theory which contains F. Then $T \nvdash \sigma_2$ and $T \nvdash \neg\sigma_2$.*

Proof. Let $n = \#\sigma_2$. Assume first $T \vdash \sigma_2$. Let m code a deduction of σ_2 from T. So, $T \vdash \theta(S^n(\mathbf{0}), S^m(\mathbf{0}))$. Also, $T \vdash \forall y\ (\theta(S^n(\mathbf{0}), y) \to \exists z < y\ \exists w\ (\rho(S^n(\mathbf{0}), w) \wedge$

$\theta(w, z)))$. These statements logically imply $\exists z < S^m(\mathbf{0})\ \exists w\ (\rho(S^n(\mathbf{0}), w) \wedge \theta(w, z))$. We violate the consistency of T by showing $T \vdash \forall z < S^m(\mathbf{0})\ \forall w\ (\rho(S^n(\mathbf{0}), w) \rightarrow \neg\theta(w, z))$. From lemma 3.3 it is enough to fix $k < m$ and show that $T \vdash \forall w\ (\rho(S^n(\mathbf{0}), w) \rightarrow \neg\theta(w, S^k(\mathbf{0})))$. By strong representability of ρ, it is enough to show that $T \vdash \neg\theta(S^{n'}(\mathbf{0}), S^k(\mathbf{0}))$, where n' is the code for $\neg\sigma_2$. Since T is consistent and $T \vdash \sigma_2$ by assumption, $T \nvdash \neg\sigma_2$, and so $\neg R(n', k)$. By representability, $T \vdash \neg\theta(S^{n'}(\mathbf{0}), S^k(\mathbf{0}))$ and we are done.

Assume next that $T \vdash \neg\sigma_2$. Let again $n' = \#\neg\sigma_2$, and let now m code a deduction from T of $\neg\sigma_2$. So, $T \vdash \theta(S^{n'}(\mathbf{0}), S^m(\mathbf{0}))$. Since $T \vdash \neg\sigma_2$ we also have $T \vdash \exists y\ (\theta(S^n(\mathbf{0}), y) \wedge \forall z < y\ \forall w\ (\rho(S^n(\mathbf{0}), w) \rightarrow \neg\theta(w, z)))$. To violate the consistency of T it is enough to show that $T \vdash \forall y\ \neg(\theta(S^n(\mathbf{0}), y) \wedge \forall z < y\ \forall w\ (\rho(S^n(\mathbf{0}), w) \rightarrow \neg\theta(w, z)))$, and for this it is enough to show that $T' = T \cup \{\alpha\}$ is inconsistent, where $\alpha(y) = (\theta(S^n(\mathbf{0}), y) \wedge \forall z < y\ \forall w\ (\rho(S^n(\mathbf{0}), w) \rightarrow \neg\theta(w, z)))$. Since $T \vdash \forall w\ (\rho(S^n(\mathbf{0}), w) \rightarrow \theta(w, S^m(\mathbf{0})))$, it follows that $T' \vdash y \leqslant S^m(\mathbf{0})$ (we use here the order axiom of F which gives $y \leqslant S^m(\mathbf{0})$ or $y > S^m(\mathbf{0})$). From lemma 3.3 it is enough to show that each $k \leqslant S^m(\mathbf{0})$ that $T'' = T \cup \{(\theta(S^n(\mathbf{0}), S^k(\mathbf{0})) \wedge \forall z < S^k(\mathbf{0})\ \forall w\ (\rho(S^n(\mathbf{0}), w) \rightarrow \neg\theta(w, z)))\}$ is inconsistent. This is clearly the case, however, since for all such k we have $T \vdash \neg\theta(S^n(\mathbf{0}), S^k(\mathbf{0}))$ since by consistency $R(n, k)$ holds (recall we are assuming $T \vdash \neg\sigma_2$). □

Note that the sentences σ_1, σ_2 of the Gödel theorems are Π_1 sentences in the language of number theory. Thus, incompleteness arises for sentences having only one unbounded number quantifier.

The incompleteness theorems as we stated them apply to theories in the language of number theory, however it ie not difficult to see that they consequently apply to to theories in which we can "interpret" the theory F. To make this precise, let $\mathcal{L}$ denote the language of number theory, and $\mathcal{L}'$ a first-order language (e.g., the language of set theory). Suppose we have formulas $\alpha_{\mathbb{N}}$, α_+, $\alpha_\cdot$, α_E, $\alpha_<$, α_S, $\alpha_{\mathbf{0}}$ of $\mathcal{L}'$. $\alpha_{\mathbb{N}}$ is intending to define a "copy" of $\mathbb{N}$, and the other formulas, α_+ for example, are intending to define the corresponding function, relation, or constant symbol on this copy. Let T' be a theory (set of sentences) in $\mathcal{L}'$, for example T' might be the axioms of ZFC. Suppose T' proves that $\exists x\ \alpha_{\mathbb{N}}(x)$ (i.e., the copy of $\mathbb{N}$ is non-empty) and also for each of the axioms ψ of F, $T' \vdash \psi'$ where ψ' is the interpretation of ψ into $\mathcal{L}'$ using the α formula in a natural way. For example, an atomic formula of the form $x + (y \cdot z) \approx w$ is replaced by $\exists z_1 \exists z_2\ \ (\alpha_{\mathbb{N}}(z_1) \wedge \alpha_{\mathbb{N}}(z_2) \wedge \alpha_\cdot(y, z, z_1) \wedge \alpha_+(x, z_1, z_2) \wedge z_2 \approx w)$. In this way, each formula ψ of number theory is replaced by a formula ψ' of $\mathcal{L}'$ such that if $\mathrm{F} \vdash \psi$ then $T' \vdash \psi'$. Of course, T' may prove more about its copy of $\mathbb{N}$ than does F, for example, if $\mathcal{L}'$ is the language of set theory and $\alpha_{\mathbb{N}} =$"$x \in \omega$," (and the other α are defined in the usual way these functions etc. are defined in set theory), then ZFC proves much more about $\mathbb{N}$ than F does, in particular ZFC proves that all of the Peano axioms hold in $\mathbb{N}$.

At any rate, if T' proves all of the ψ' for $\psi \in \mathrm{F}$, then all of the proofs of the incompleteness results given above for $\mathcal{L}$ may be carried over immediately for theories extending T'. Since F is finite, it follows that there is a finite T' which suffices to prove all of the ψ'.

For example, let ZFC$'$ denote the finite subset of ZFC which suffices to prove all of the ψ' for $\psi \in \mathrm{F}$. We then have:

Theorem 4.7. *Let T be a recursive, consistent theory in the language of set theory which extends the finite fragment ZFC$'$. Then T is imcomplete. Moreover, there is a Π_1 sentence $\sigma_2 = \sigma_2(T)$ such that $T \nvdash \sigma_2$ and $T \nvdash \neg\sigma_2$.*

Proof. Define $R(a,b)$ and $\theta(x,y)$ as in theorem 4.6. Let $\theta'(x,y)$ be the interpretation of θ into the language of set theory. Thus, if $R(,b)$ then F $\vdash \theta(S^a(\mathbf{0}), S^b(\mathbf{0}))$ and so $T \vdash \theta'_{a,b}$ and likewise for $\neg R(a,b)$, where $\theta'_{a,b}$ denotes the interpretation of $\theta(S^a(\mathbf{0}), S^b(\mathbf{0}))$. The proof is now essentially identical to theorem 4.6 using θ' in place of θ. □

Lastly, we discuss the second Gödel incompleteness theorem. We now consider theories T which may in the language of number theory, or set theory, etc., for which we have an interpretation of $\mathbb{N}$ as above. Let $R(b)$ iff b codes a deduction from T of a logical contradiction, say $\exists x\ (x \not\approx x)$. Let $\theta(x)$ represent R in F, and let CON$_T$ be the sentence $\neg\exists x\ \theta$. If T is in set theory, say, then we let CON$_T$ be the interpretation of this sentence into the language of set theory. The second version of the incompleteness theorem say that if T is recursive, consistent, and sufficiently strong (but we need more that T contains F now), then $T \nvdash$ CON$_T$. It is enough to have T contain PA (even a smaller fragment of it, say Π_1-induction), but we state the result now for theories extending ZFC.

Theorem 4.8. *Let T be a recursive, consistent theory in the language of set theory extending ZFC. Then $T \nvdash$ CON$_T$.*

Proof. Let σ_1 be the sentence from the first version of the Gödel incompleteness theorem, so $T \nvdash \sigma_1$ (recall this direction only used the consistency of T). The proof of this (theorem 4.5) was presented in the metatheory. That is, in the metatheory we showed that if T is consistent, then $T \nvdash \sigma_1$. Closer examination of the proof reveals that the only properties of the integers used in the proof are theorems of PA. Certainly, however, they are all theorems of ZFC. Thus, this argument in the metatheory, when formalized, becomes the statement that ZFC $\vdash$ (CON$_T \to \phi$), where ϕ is the formalization of the statement "there does not exist a proof of σ_1 from T." However, this formalization is just the statement $\tau(S^{\#\sigma_1}(\mathbf{0}))$ (using the notation of theorem 4.5), and this is T provably equivalent to σ_1 (more precisely, the interpretations of these statements into the language of set theory). Thus, ZFC $\vdash$ (CON$_T \to \sigma_1$). It follows that ZFC $\nvdash$ CON$_T$ from theorem 4.5. □