Lesson 13: Gödel’s incompleteness theorem, part. II

Theme: Representability of recursive functions, fixed point lemma and Gödel’s first incompleteness theorem.

1 Some preliminary results on Robinson’s arithmetic \( \mathcal{Q} \)

Recall that all our formulas are over the vocabulary \( L_{ar} = \{ \tilde{0}, \text{Succ}, +, \cdot \} \), and that for every integer \( n \geq 0 \), we write \( \overline{n} \) to denote the term \( \text{Succ}^n(\tilde{0}) \), i.e., applying \( \text{Succ} \) on \( \tilde{0} \) for \( n \) number of times. For a vector \( \overline{a} = (a_1, \ldots, a_n) \) of integers, we will write \( \overline{a} \) to denote \( (a_1, \ldots, a_n) \).

By a straightforward induction on \( n \) and \( m \), it is not that difficult to show that for every integers \( n, m \geq 0 \), the following holds.

\[
\text{(C1)} \quad \mathcal{Q} \vdash (\text{Succ}(x) + \overline{n}) \approx (x + \text{Succ}(\overline{n})).
\]

\[
\text{(C2)} \quad \mathcal{Q} \vdash (m + \overline{n}) \approx m + n.
\]

\[
\text{(C3)} \quad \mathcal{Q} \vdash (m \cdot \overline{n}) \approx m \cdot n.
\]

\[
\text{(C4)} \quad \mathcal{Q} \vdash n \not\approx m, \text{ for every } n \neq m.
\]

\[
\text{(C5)} \quad \mathcal{Q} \vdash m \leq n, \text{ for every } m \leq n.
\]

Recall that our vocabulary \( L_{ar} \) does not include \( \leq \). The formula \( m \leq n \) is actually an abbreviation for \( \exists z \ m + z \approx n \).

\[
\text{(C6)} \quad \mathcal{Q} \vdash \neg(m \leq n), \text{ for every } m \not\leq n.
\]

\[
\text{(C7)} \quad \mathcal{Q}, x \leq n \vdash (x \approx \tilde{0}) \lor (x \approx \tilde{1}) \lor \cdots \lor (x \approx n).
\]

\[
\text{(C8)} \quad \mathcal{Q} \vdash (x \leq n) \lor (n \leq x).
\]

All these statements show that the natural meaning of the standard operations like addition and multiplication are provable in \( \mathcal{Q} \), and hence, in any extension \( T \supseteq \mathcal{Q} \).

Definition 13.1

- A formula \( \varphi \) is called a \( \Delta_0 \)-formula, if all its quantifiers are bounded quantifiers, i.e., of the form \( \forall x \leq t \) \( \alpha \), where \( t \) is a term over \( L_{ar} \).
  
  Intuitively \( \forall x \leq t \) \( \alpha \) states “for every \( x \leq t \), the formula \( \alpha \) holds.”

- A formula \( \varphi \) is called a \( \Sigma_1 \)-formula, if it is of the form \( \exists \bar{x} \psi \), where \( \psi \) is a \( \Delta_0 \)-formula.

- A formula \( \varphi \) is called a \( \Pi_1 \)-formula, if it is of the form \( \forall \bar{x} \psi \), where \( \psi \) is a \( \Delta_0 \)-formula.

Proposition 13.2 Let \( t \) be a term over \( L_{ar} \) with free variables \( x_1, \ldots, x_n \). For a valuation \( \text{val}: \text{VAR} \rightarrow \mathbb{N} \), consider the substitution \( \text{sub} := [x_1/\text{val}(x_1), \ldots, x_n/\text{val}(x_n)] \). Then,

\[
\begin{align*}
\text{val}^N[t] & = m \quad \text{if and only if} \quad \mathcal{Q} \vdash t[\text{sub}] \approx m \\
\text{val}^N[t] & \leq m \quad \text{if and only if} \quad \mathcal{Q} \vdash t[\text{sub}] \leq m
\end{align*}
\]

Proof. By straightforward induction on \( t \) together with (C1)–(C8) above. ■

Theorem 13.3 below will be very useful. It states that in order to check whether a \( \Delta_0 \)-sentence \( \varphi \) is provable in \( \mathcal{Q} \), it is sufficient to check whether it holds in \( \mathbb{N} \). In other words, instead of looking for a proof of \( \varphi \), we simply checks whether it holds in \( \mathbb{N} \), which is a more convenient and intuitive system to work with.
Theorem 13.3 For every $\Delta_0$-formula $\varphi(\bar{x})$, where $\bar{x} = (x_1, \ldots, x_n)$, the following holds. For every $\bar{a} = (a_1, \ldots, a_n) \in \mathbb{N}^n$:

$$\mathcal{N} \models \varphi(\bar{a}) \iff Q \vdash \varphi(\bar{a}).$$

Proof. The proof is by induction on $\varphi$. The base case, when the atomic formula of the form $s \approx t$, can be deduced directly from Proposition 13.2.

The induction step consists of three cases.

Case 1: $\varphi(\bar{x})$ is $\neg \alpha(\bar{x})$.

$\mathcal{N} \models \varphi(\bar{a})$ if and only if $\mathcal{N} \not\models \alpha(\bar{a})$, if and only if $Q \not\vdash \alpha(\bar{a})$, if and only if $Q \vdash \varphi(\bar{a})$, with the second “if and only if” coming from the induction hypothesis.

Case 2: $\varphi(\bar{x})$ is $\alpha_1(\bar{x}) \land \alpha_2(\bar{x})$.

$\mathcal{N} \models \varphi(\bar{a})$ if and only if $\mathcal{N} \models \alpha_1(\bar{a}) \land \alpha_2(\bar{a})$, if and only if $\mathcal{N} \models \alpha_1(\bar{a})$ and $\mathcal{N} \models \alpha_2(\bar{a})$, if and only if $Q \vdash \alpha_1(\bar{a})$ and $Q \vdash \alpha_2(\bar{a})$, if and only if $Q \vdash \varphi(\bar{a})$, with the third “if and only if” coming from the induction hypothesis.

Case 3: $\varphi(\bar{x})$ is $\forall z \leq t \alpha(\bar{x}, z)$.

Let $\text{val}$ denote the valuation that maps $x_i$ to $a_i$, and $\text{sub}$ denote the substitution that substitute $x_i$ with $a_i$. Let $M = t^{\mathcal{N}[\text{val}]}$. By Proposition 13.2 we have $Q \vdash t \approx M$.

$\mathcal{N} \models \varphi(\bar{a})$ if and only if for every $m \leq M$,

$$\mathcal{N}, [\text{val}, z \mapsto m] \models \alpha(\bar{a}, z),$$

which holds, if and only if for every $m \leq M$,

$$Q \vdash \alpha[\text{sub}, z/m],$$

which holds, if and only if

$$Q, z \leq M \vdash \alpha[\text{sub}, z],$$

which holds, if and only if

$$Q \vdash (\forall z \leq M) \alpha(\bar{x}, z),$$

which holds, if and only if

$$Q \vdash (\forall z \leq t) \alpha(\bar{x}, z).$$

The third “if and only if” comes from the induction hypothesis, while the fourth is from (C5) and (C7). The fifth comes from the fact that $(\forall z \leq M) \alpha(\bar{x}, z)$ is an abbreviation of $\forall z (z \leq M \rightarrow \alpha(\bar{x}, z))$. The last one comes from $Q \vdash t \approx M$. 

$\blacksquare$
2 Representable functions

In the following let \( \bar{x} \) be a vector of variables, and \( \bar{a} \) be a vector of natural numbers with the same length as \( \bar{x} \).

Representable functions in a theory \( T \supseteq \mathbb{Q} \). A function \( f : \mathbb{N}^k \to \mathbb{N} \) is called representable in a theory \( T \supseteq \mathbb{Q} \), if there is a formula \( \varphi(\bar{x},y) \) such that \( f(\bar{a}) = m \) if and only if \( T \vdash \varphi(\bar{a},m) \). Note that this is equivalent to saying that if \( T \vdash y \approx m \leftrightarrow \varphi(\bar{a},y) \).

It is \( \Sigma_1 \)-representable, if the formula \( \varphi(\bar{x}) \) is \( \Sigma_1 \)-formula, and the formula \( \varphi \) is called the representation formula for \( f \).

Likewise, a relation \( R \subseteq \mathbb{N}^k \) is called representable in a theory \( T \supseteq \mathbb{Q} \), if there is a formula \( \varphi(\bar{x}) \) such that if \( \bar{a} \in R \), then \( T \vdash \varphi(\bar{a}) \); and if \( \bar{a} \notin R \), then \( T \vdash \neg \varphi(\bar{a}) \).

Arithmetical functions (functions representable in \( \mathcal{N} \)). A function \( f : \mathbb{N}^k \to \mathbb{N} \) is called arithmetical, or representable in \( \mathcal{N} \), if there is a formula \( \varphi(\bar{x},y) \) such that \( f(\bar{a}) = m \) if and only if \( \mathcal{N} \models \varphi(\bar{a},m) \). The notions of \( \Sigma_1 \)-representable and \( \Pi_1 \)-representable are defined similarly as above.

3 Representability of recursive functions

In this section we will show the following theorem.

Theorem 13.4 Every recursive function \( f \) is representable by a \( \Sigma_1 \)-formula in \( \mathbb{Q} \).

The proof consists of two steps:

1. We show that \( f \) is representable in \( \mathcal{N} \) by a \( \Sigma_1 \)-formula, as well as by a \( \Pi_1 \)-formula.
2. We show that it can be represented by a \( \Sigma_1 \)-formula in \( \mathbb{Q} \).

Representing \( f \) in \( \mathcal{N} \). The proof is by induction on \( f \). The base case is as follows.

- \( f \) is the constant zero function, i.e., \( f(\bar{x}) = 0 \).
  Then, \( \varphi(\bar{x},y) := y \approx 0 \) is a \( \Delta_0 \)-formula representing \( f \).

- \( f \) is the successor function of one of its component, i.e., \( f(\bar{x}) = \text{Succ}(x_i) \).
  Then, \( \varphi(\bar{x},y) := y \approx \text{Succ}(x_i) \) is a \( \Delta_0 \)-formula representing \( f \).

- \( f \) is the projection function to one of its components, i.e., \( f(\bar{x}) = x_i \).
  Then, \( \varphi(\bar{x},y) := y \approx x_i \) is a \( \Delta_0 \)-formula representing \( f \).

The induction step is as follows.

- Functions obtained from applying the composition rule \( \text{Oc} \).
Let \( f = h[g_1, \ldots, g_m] \) be a function from \( \mathbb{N}^n \to \mathbb{N} \), i.e., each \( g_i : \mathbb{N}^n \to \mathbb{N} \) and \( h : \mathbb{N}^m \to \mathbb{N} \).
By the induction hypothesis, let \( \alpha \) and \( \gamma_i \) be \( \Sigma_1 \)-formulas representing \( h \) and \( \gamma_i \), respectively.
Both \( \Sigma_1 \)-formula \( \varphi_1 \) and \( \Pi_1 \)-formula \( \varphi_2 \) below represent \( f \) in \( \mathcal{N} \).

\[
\varphi_1(\bar{x},z) := \exists y_1 \cdots \exists y_m \bigwedge_{1 \leq i \leq m} \gamma_i(\bar{x},y_i) \wedge \alpha(y_1, \ldots, y_m, z)
\]
\[
\varphi_2(\bar{x},z) := \forall u \left( \varphi_1(\bar{x},u) \rightarrow u \approx z \right)
\]
• Functions obtained from applying the primitive recursive rule $\text{Op}$.

This is the most challenging part. See the appendix for the details.

• Functions obtained from applying the rule $\text{O}\mu$.

Let $\bar{x} = (x_1, \ldots, x_n)$, and let $f(\bar{x}) := \mu y[g(\bar{x}, y) = 0]$. By the induction hypothesis, there is a $\Sigma_1$-formula $\alpha_1(x, y, z)$, and a $\Pi_1$-formula $\alpha_2(x, t, z)$ representing $g$ in $\mathcal{N}$.

$$\alpha_1(x, y, z) := \exists v \psi_1(x, y, z, v)$$

where $\psi_1$ is a $\Delta_0$-formula

$$\alpha_2(x, y, z) := \forall w \psi_2(x, y, z, w)$$

where $\psi_2$ is a $\Delta_0$-formula

Consider the formula $\varphi_1$ below.

$$\varphi_1(x, y) := \alpha_1(x, y, \bar{0}) \land (\forall z < y) \neg \alpha_2(x, z, \bar{0})$$

$$:= \exists v \psi_1(x, y, z, v) \land (\forall z < y) \exists w \neg \psi_2(x, z, \bar{0}, w)$$

We have the following identity (can be easily proved) in $\mathcal{N}$:

$$\mathcal{N} \models (\forall z < y) \exists u \psi \equiv \exists u'(\forall z < y) \neg (\forall u < u') \neg \psi$$

Therefore, the following $\Sigma_1$-formula $\varphi'_1$ is equivalent to $\varphi_1$ in $\mathcal{N}$.

$$\varphi'_1(x, y) := \exists v \psi_1(x, y, z, v) \land \exists \bar{w}'(\forall z < y) \psi_2(x, z, \bar{0})$$

where $\psi_2'$ is a $\Delta_0$-formula

$$:= \exists v \exists \bar{w}' \left( \psi_1(x, y, z, v) \land (\forall z < y) \psi_2'(x, z, \bar{0}) \right)$$

Thus, $\varphi'_1$ is the desired $\Sigma_1$-formula representing $f$ in $\mathcal{N}$.

A $\Pi_1$-formula $\varphi_2$ representing $f$ can be obtained as follows.

$$\varphi_2(x, y) := \forall u \left( \varphi'_1(x, u) \rightarrow u \approx y \right)$$

Representing $f$ in $\mathcal{Q}$. Note that if $f$ is representable in $\mathcal{Q}$, then by monotonicity rule, it is representable in $T \supseteq \mathcal{Q}$.

Let $f : \mathbb{N}^n \rightarrow \mathbb{N}$ be a recursive function, and let $\varphi(\bar{x}, y) := \exists z \psi(\bar{x}, y, z)$ be its representing formula in $\mathcal{N}$, where $\psi$ is $\Delta_0$-formula. That is, for every $\bar{a} \in \mathbb{N}^n$,

$$f(\bar{a}) = b \text{ if and only if } \mathcal{N} \models \varphi(\bar{a}, b).$$

We have to show that for every $\bar{a} \in \mathbb{N}^n$,

$$f(\bar{a}) = b \text{ if and only if } \mathcal{Q} \models \varphi(\bar{a}, b).$$

We start with the “if” part. Suppose $f(\bar{a}) = b$. Since $\varphi$ represents $f$, for some $\bar{w}$,

$$\mathcal{Q} \models \psi(\bar{a}, b, \bar{w})$$

Since $\psi$ is $\Delta_0$-formula, by Theorem 13.3, we have $\mathcal{N} \models \psi(\bar{a}, b, \bar{w})$, and hence, $\mathcal{Q} \models \exists \bar{z} \psi(\bar{a}, b, \bar{z})$.

Now, we show the “only if” part. Suppose for some $\bar{w}$, $\mathcal{Q} \models \psi(\bar{a}, b, \bar{w})$. Since $\mathcal{N} \models \mathcal{Q}$, we have that $\mathcal{N} \models \psi(\bar{a}, b, \bar{w})$, and thus, $\mathcal{N} \models \exists \bar{z} \psi(\bar{a}, b, \bar{z})$. Therefore, $\mathcal{N} \models \varphi(\bar{a}, b)$. Since $\varphi$ represents $f$, we have $f(\bar{a}) = b$. This completes the proof of Theorem 13.4.
4 Fixed point lemma and Gödel’s first incompleteness theorem

Recall that in order to prove Gödel’s incompleteness theorem, we have to show that:

- Every recursive function is representable in $Q$.
- For a consistent and recursively axiomatizable theory $T \supseteq Q$, there is a sentence $\Psi$ such that $T \vdash \Psi \iff (\forall y \neg \text{ISPROOFOF}_{T}(y, \#\Psi))$.

We describe how to achieve the first part in the previous section. We will now describe how to achieve the second part.

For a variable $x$, define the function $\text{Subs}_x : \mathbb{N}^2 \to \mathbb{N}$ as follows.

$$\text{Subs}_x(N, m) := K$$

where $K$ is “the formula” obtained by substituting variable $x$ with the term $m$ in “formula” $N$. Here, “the formulas” $K$ and $N$ refer to the formulas whose Gödel’s numbers are $K$ and $N$, respectively. It is not that difficult to think of a computer program for $\text{Subs}_x$. So, it is also a recursive function, and can be represented in a theory $Q$, and hence, in any extension $T \supseteq Q$.

Let $\Lambda_{\text{Subs}_x}(v_1, v_2, v_3)$ be a $\Sigma_1$-formula representing $\text{Subs}_x$.

**Lemma 13.5 (Fixed point lemma)** Let $T \supseteq Q$. For every formula $\alpha(z)$ over vocabulary $\{0, \text{Succ}, +, \cdot\}$, there is a formula $\gamma$ such that $T \vdash \gamma \iff \alpha(\#\gamma)$.

**Proof.** Due to the definition of $\Lambda_{\text{Subs}_x}(v_1, v_2, v_3)$, for every formula $\varphi$,

$$T \vdash \Lambda_{\text{Subs}_x}(\#\varphi, n, y) \iff y \approx \#\varphi[x/n]$$

If we plug in $n$ with $\#\varphi$ itself,

$$T \vdash \Lambda_{\text{Subs}_x}(\#\varphi, \#\varphi, y) \iff y \approx \#\varphi[x/\#\varphi] \quad (1)$$

Let $\beta(x)$ be the following formula.

$$\beta(x) := \forall y \left( \Lambda_{\text{Subs}_x}(x, x, y) \rightarrow \alpha[z/y] \right)$$

Consider $\gamma := \beta[x/\#\beta]$. That is, $\gamma = \forall y \left( \Lambda_{\text{Subs}_x}(\#\beta, \#\beta, y) \rightarrow \alpha[z/y] \right)$.

By (1),

$$T \vdash \gamma \iff \forall y \left( y \approx \#\beta[x/\#\beta] \rightarrow \alpha[z/y] \right)$$

Since $\gamma = \beta[x/\#\beta]$, $T \vdash \gamma \iff \forall y \left( y \approx \#\gamma \rightarrow \alpha[z/y] \right)$.

This completes the proof of fixed point lemma. $lacksquare$

To wrap up, we state and prove formally Gödel’s incompleteness theorem.
Theorem 13.6 (Gödel’s incompleteness theorem) For every consistent and recursively axiomatizable theory \( T \supseteq Q \), there is a sentence \( \Psi \) such that neither \( T \vdash \Psi \) nor \( T \vdash \neg \Psi \).

**Proof.** Since \( T \) is recursively axiomatizable theory, we have a “computer program” on an input proof \( y \), output \( x \), which represents the conclusion of the proof \( y \). By Church-Turing thesis, every “computer program” is equivalent to a recursive function, and by Theorem 13.4, a recursive function can be represented in \( \Sigma_1 \)-formula in \( T \supseteq Q \). Thus, we have a \( \Sigma_1 \)-formula \( \text{IsProofOf}_T(y, x) \) which states that \( y \) is a proof of \( x \). In particular, we also have the following formula.

\[
\text{PROVABLE}_T(x) := \exists y \, \text{IsProofOf}_T(y, x)
\]

such that

\[
T \vdash \varphi \leftrightarrow \text{PROVABLE}_T(\sharp \varphi)
\]

Consider the negation of \( \text{PROVABLE}_T(x) \), i.e., \( \neg \text{PROVABLE}_T(x) \). By fixed-point lemma, there is \( \Psi \) such that

\[
T \vdash \Psi \leftrightarrow \neg \text{PROVABLE}_T(\sharp \Psi)
\]

which is simply

\[
T \vdash \Psi \leftrightarrow \forall y \neg \text{IsProofOf}_T(y, \sharp \Psi)
\]

Following the argument in Section 3 in Lesson 13, neither \( \Psi \) nor \( \neg \Psi \) are provable in \( T \).

**Appendix: Representing the Op rule**

The proof consists of two steps.

- First, we construct a function \( G : \mathbb{N}^2 \to \mathbb{N} \) representable with \( \Delta_0 \)-formula such that for every \( n \), for every sequence \( c_0, \ldots, c_n \), there is \( c \) such that for all \( i = 0, \ldots, n \), we have \( G(c, i) = c_i \).

- Using the function \( G \) constructed, we can represent the Op rule with a \( \Sigma_1 \)-formula.

Intuitively, the function \( G \) “encodes” every sequence element \( (c_0, \ldots, c_n) \in \mathbb{N}^n = \bigcup_{i \geq 1} \mathbb{N}^i \) as a number \( c \) such that to retrieve an element \( c_i \), we simply “access” \( G(c, i) \).

**Constructing the function \( G \) (Gödel’s way).** Consider the following bijection \( \varphi : \mathbb{N}^2 \to \mathbb{N} \).

\[
\varphi(a, b) := a + \sum_{i=1}^{a+b} i = a + \frac{1}{2}(a + b)(a + b + 1)
\]

Note that \( a, b \leq \varphi(a, b) \), for every \( a, b \). It is trivial that \( \varphi \) can be represented by a \( \Delta_0 \)-formula.

Let \( F : \mathbb{N}^3 \to \mathbb{N} \) be the following function.

\[
F(a, b, i) := \text{the remainder of } a \text{ divided by } 1 + (1 + i)b
\]

It is not that difficult to show that the function \( F \) is represented by a \( \Delta_0 \)-formula.

Let \( \text{Proj}_x \) and \( \text{Proj}_y \) be the following functions. For every \( m \in \mathbb{N} \), if \( \varphi^{-1}(m) = (a, b) \),

\[
\text{Proj}_x(m) := a \quad \text{and} \quad \text{Proj}_y(m) := b
\]
Consider the following function $G : \mathbb{N}^2 \rightarrow \mathbb{N}$.

$$G(c, i) := F(\text{Proj}_x(c), \text{Proj}_y(c), i)$$

The function $G$ can be represented with a $\Delta_0$-formula as follows.

$$G(c, i) = m \text{ if and only if } (\exists x \leq c)(\exists y \leq c) \left( \varphi(a, b) = c \land F(a, b, i) = m \right)$$

The underlined parts denote abbreviations of the formulas that represent $\varphi(a, b) = c$ and $F(a, b, i) = m$, respectively.

We will show that $G$ is our desired function. In the following we write $a \mid b$ to denote that $a$ divides $b$, i.e., when $b$ is divided by $a$, there is no remainder. For two positive integers $a, b$, we say that $a$ and $b$ are coprime, if there is no prime $p$ that divides both $a$ and $b$.

**Lemma 13.7 (Euclid)** If $a$ and $b$ are coprime, then there are $x, y \in \mathbb{N}$ such that $ax + 1 = by$.

**Theorem 13.8 (Chinese remainder theorem)** Let $c_0, \ldots, c_k, d_0, \ldots, d_k$ such that $c_i < d_i$. Let $d_1, \ldots, d_k$ be pairwise coprime. Then, there exists an integer $a \in \mathbb{N}$ such that $\text{rem}(a, d_i) = c_i$, i.e., the remainder of $a$ divided by $d_i$ is $c_i$.

**Theorem 13.9** For every $n$, for every sequence $c_0, \ldots, c_n$, there exist $a, b$ such that for all $i = 0, \ldots, n$, we have $F(a, b, i) = c_i$.

Since $G(\varphi(a, b), i) = F(a, b, i)$, we have that for every sequence $c_0, \ldots, c_n$, there is $c$, which is $\varphi(a, b)$ and greater than each $c_i$, such that for all $i = 0, \ldots, n$, we have $G(c, i) = c_i$.

**Proof.** Let $c_0, \ldots, c_n$ be a sequence of natural numbers. Consider the following two numbers $M$ and $K$.

- $M := \max(n, c_0, \ldots, c_n)$.
- $b := \text{lcm}(1, \ldots, M)$, where “lcm” is least common multiplier.

Let $d_i := 1 + (1 + i)b$, for each $i = 0, \ldots, n$. Note that $d_i > c_i$.

We claim that $d_0, \ldots, d_n$ are pairwise coprime. Suppose to the contrary that there is a prime $p$ that divides both $d_i$ and $d_j$. Thus, $p \mid d_i - d_j = (i - j)b$. So, either $p \mid (i - j)$ or $p \mid b$.

Now, $i, j \leq M$, since $b$ is the least common multiplier of all integers between 1 and $M$, we have $(i - j) \mid b$. This means that $p \mid b$. By definition of $d_i, b \mid (d_i - 1)$, which means $p \mid (d_i - 1)$. This is absurd, since $p \mid d_i$. So, there is such prime $p$ that divides $d_i$ and $d_j$. In other words, $d_0, \ldots, d_n$ are coprime.

By Theorem 13.8, there is $a$ such that $\text{rem}(a, d_i) = c_i$. By the definition of the function $F$, we have $F(a, b, i) = c_i$. By the construction, it is obvious that $\varphi(a, b) > c_i$. $\blacksquare$

**Representing functions obtained from applying Op rule.** Let $g \in F_n$ and $h \in F_{n+2}$ be recursive functions.

- Let $g$ be represented by a $\Sigma_1$-formula $\alpha_1$, as well as a $\Pi_1$-formula $\alpha_2$.
- Let $h$ be represented by a $\Sigma_1$-formula $\beta_1$, as well as a $\Pi_1$-formula $\beta_2$.

Suppose $f \in F_{n+1}$ is the function obtained via the Op rule as follows. For every $\bar{a} \in \mathbb{N}^n$,

$$f(\bar{a}, 0) := g(\bar{a}) \quad \text{and} \quad f(\bar{a}, \text{Succ}(b)) := h(\bar{a}, b, f(\bar{a}, b))$$
The following formula represents $f$.

$$
\varphi(\bar{x}, y, z) := \left(y \approx \tilde{0} \rightarrow \alpha_1(\bar{x}, z)\right) \land \exists z' (\forall y' < y) \left(G(z', \text{Succ}(y')) = h(\bar{x}, y', G(z', y'))\right)
$$

Intuitively, the variable $z'$ is such that for every $i \leq y$, $G(z', i) = f(\bar{x}, i)$.

Now, $\varphi(\bar{x}, y, z)$ can be rewritten into:

$$
\varphi(\bar{x}, y, z) := \left(y \approx \tilde{0} \rightarrow \alpha_1(\bar{x}, z)\right) \land \\
\exists z' (\forall y' < y)(\forall u < z')(\forall v < z') \\
\left(G(z', \text{Succ}(y')) = u \land G(z', y') = v \rightarrow \beta_1(\bar{x}, y', u_2, u_1)\right)
$$

By pulling all the existential quantifiers from $\beta_1$ and $\exists z'$ to the front of the formula, we obtain a $\Sigma_1$-formula. A $\Pi_1$-formula can be obtained via:

$$
\varphi'(\bar{x}, y, z) := \forall w \ \varphi(\bar{x}, y, w) \rightarrow w \approx z
$$