

Representable Functions

We will show that Peano arithmetic can express all primitive recursive functions.

Note: This is a technical chapter. Readers in a hurry may simply focus on the result stated at the beginning. However, it is also a pivotal chapter, since we will connect semantics (the world of natural numbers) with syntax (the language of Peano arithmetic). It also contains one of Gödel's technical achievements: the function β . This function encodes any list of numbers using only two natural numbers.

1. A primitive recursive predicate is representable

1.1. Representable predicate

For a fixed $p \geq 1$, write $\mathbf{x} = (x_1, \dots, x_p)$. If $\mathbf{n} = (n_1, \dots, n_p) \in \mathbb{N}^p$, write $\bar{\mathbf{n}} = (\bar{n}_1, \dots, \bar{n}_p)$ for the corresponding sequence of numerals.

Definition.

A p -place predicate $R(\mathbf{x})$ is **representable** if there exists a formula $\mathcal{P}(\mathbf{x})$ in the language $\mathcal{L}_0$ such that, for every $\mathbf{n} \in \mathbb{N}^p$:

- if $R(\mathbf{n})$ is true, then: $\mathcal{PA} \vdash \mathcal{P}(\bar{\mathbf{n}})$
- if $R(\mathbf{n})$ is false, then: $\mathcal{PA} \vdash \neg \mathcal{P}(\bar{\mathbf{n}})$

Remark: the full term is “predicate strongly representable in the theory $\mathcal{PA}$ of Peano arithmetic”.

Notice carefully the passage from a natural number to its numeral. For example, if $p = 1$, the predicate is evaluated as $R(n)$ (with $n \in \mathbb{N}$), while the formula $\mathcal{P}$ is evaluated as $\mathcal{P}(\bar{n})$, where the numeral $\bar{n} = SSS \dots S0$ represents n in $\mathcal{L}_0$.

Example.

The predicate $R(x)$: “ x is an even natural number” is representable by the following formula:

$$\mathcal{P}(x) : \exists y \ x = \bar{2} \times y$$

Checking this property for a specific natural number is a finite process that can be translated step by step into a proof within $\mathcal{PA}$.

1.2. Theorem

Here is the theorem that will be useful in the following chapters. It will be applied to the predicate “is_proof(N, n)”, which determines whether the formulas encoded by the super-Gödel number N form a proof of the formula with Gödel number n .

Theorem 1.

If R is a primitive recursive predicate, then R is representable.

Recall that a predicate is primitive recursive if its characteristic function is primitive recursive. Similarly, a predicate will be representable if its characteristic function is representable. We will therefore define and study the more general notion of a representable function, which is essential for proving the theorem above.

2. Representable functions

2.1. Definition

Definition.

A function $f : \mathbb{N}^p \rightarrow \mathbb{N}$ is **representable** if there exists a formula $\mathcal{P}(\mathbf{x}, y)$ such that, for every $\mathbf{n} = (n_1, \dots, n_p) \in \mathbb{N}^p$:

$$\mathcal{P}\mathcal{A} \vdash \forall y \left[\mathcal{P}(\bar{\mathbf{n}}, y) \leftrightarrow y = \overline{f(\mathbf{n})} \right]$$

Remarks:

- The full term is “function strongly representable in the theory $\mathcal{P}\mathcal{A}$ ”.
- Once again, this definition connects a natural number $k \in \mathbb{N}$ with its numeral $\bar{k}$ in $\mathcal{L}_0$.
- The formula $\mathcal{P}$ should be understood as the graph of f . For the graph of a function from $\mathbb{R}$ to $\mathbb{R}$, we have $y = f(x)$ if and only if (x, y) belongs to the graph of f . Here, for a function from $\mathbb{N}^p$ to $\mathbb{N}$, we have $y = f(\mathbf{x})$ if and only if $\mathcal{P}(\mathbf{x}, y)$. This is an equivalent way to define the function.

Example.

The function $f : \mathbb{N} \rightarrow \mathbb{N}$ defined by $f(x) = 2x + 1$ is representable. Indeed, consider the formula:

$$\mathcal{P}(x, y) : y = 2x + 1$$

It is therefore true (and provable in Peano arithmetic) that $\mathcal{P}(x, y)$ holds if and only if $y = f(x)$. In other words, for every $n \in \mathbb{N}$, $\mathcal{P}(n, y)$ is true if and only if $y = f(n)$.

But be careful: we have not quite constructed the expression in the language $\mathcal{L}_0$. Indeed, the natural number 2 does not exist there as a symbol; it must be written $\bar{2} = SS0$. The correct formula in $\mathcal{L}_0$ is therefore:

$$\mathcal{P}(x, y) : y = S(SS0 \times x)$$

And for every $n \in \mathbb{N}$, the equivalence is:

$$\forall y \left[\mathcal{P}(\bar{n}, y) \leftrightarrow y = \overline{f(n)} \right]$$

(where $\bar{n} = SS \dots S0$ is the numeral for the natural number n and $\overline{f(n)} = SS \dots S0$ is the numeral for the natural number $f(n)$).

This is indeed a passage from the world of natural numbers to the language $\mathcal{L}_0$.

2.2. Equivalent definition

To better understand the definition of a representable function, here is an equivalent definition.

Definition.

A function $f : \mathbb{N}^p \rightarrow \mathbb{N}$ is **representable** if there exists a formula $\mathcal{P}(\mathbf{x}, y)$ such that, for every $\mathbf{n} \in \mathbb{N}^p$, setting $m = f(\mathbf{n})$, we have:

1. $\mathcal{P}\mathcal{A} \vdash \mathcal{P}(\bar{\mathbf{n}}, \bar{m})$ **correctness**
2. $\mathcal{P}\mathcal{A} \vdash \forall y [\mathcal{P}(\bar{\mathbf{n}}, y) \rightarrow y = \bar{m}]$ **uniqueness**

Correctness means that the formula $\mathcal{P}$ does recover the value m of $f(\mathbf{n})$. Uniqueness is essential, since a function can of course take only one value.

Let us show that this second definition is equivalent to the first. The original definition contains an equivalence. Uniqueness is exactly the implication $\rightarrow$. Starting from the original definition and specializing the $\forall$ to $y = \bar{m}$ gives exactly $\mathcal{P}(\bar{\mathbf{n}}, \bar{m})$. Conversely, starting from the second definition, we have $\mathcal{P}(\bar{\mathbf{n}}, \bar{m})$, which implies that $y = \bar{m} \rightarrow \mathcal{P}(\bar{\mathbf{n}}, y)$. We then introduce $\forall y$ to obtain the reverse implication ($\leftarrow$).

In the exercises, we will see counterexamples of formulas that fail correctness or uniqueness.

2.3. Representable predicate

Let us return to the notion of a representable predicate.

Proposition 1.

A predicate $R(\mathbf{x})$ is representable if and only if its characteristic function $\chi_R(\mathbf{x})$ is representable.

Recall that, by definition, $\chi_R(\mathbf{x}) = 1$ if $R(\mathbf{x})$ is true, and $\chi_R(\mathbf{x}) = 0$ otherwise.

To simplify the proofs, here and often below, we will write them using as few variables as possible, for example with $p = 1$. To obtain the general proof, it would of course be enough to replace x by $\mathbf{x}$ and n by $\mathbf{n}$. We will also write n for both the natural number and its numeral $\bar{n}$, since there is no ambiguity. The following proof is fairly technical and may be skipped on a first reading.

Proof.

• $\Rightarrow$

Assumption: χ_R is a representable function.

Goal: R is a representable predicate.

Let $\mathcal{P}(x, y)$ be the formula provided by the assumption, and define $\mathcal{P}_1(x)$ as $\mathcal{P}(x, 1)$. In the formula obtained from the assumption “ χ_R is a representable function”, set $y = 1$ to obtain the equivalence:

$$\mathcal{P}\mathcal{A} \vdash [\mathcal{P}(n, 1) \leftrightarrow 1 = \chi_R(n)]$$

- If $R(n)$ is true, then $\chi_R(n) = 1$. Thus $1 = \chi_R(n)$ is true. By the equivalence, $\mathcal{P}_1(n)$ is also true. More precisely, $\mathcal{P}\mathcal{A} \vdash (\mathcal{P}_1(n) \leftrightarrow (1 = 1))$. Therefore, by the definition of implication, we obtain: $\mathcal{P}\mathcal{A} \vdash \mathcal{P}_1(n)$.
- If $R(n)$ is false, then $\chi_R(n) = 0$, so the equivalence shows that $\neg\mathcal{P}_1(n)$ is true.

Thus, the predicate R is indeed represented by the formula $\mathcal{P}_1$.

• $\Leftarrow$

Assumption: R is a representable predicate.

Goal: χ_R is a representable function.

The assumption gives us a formula $\mathcal{P}_0(x)$ associated with the predicate R . We then construct the formula $\mathcal{P}(x, y)$ for the function χ_R as follows:

$$\mathcal{P}(x, y) : (\mathcal{P}_0(x) \wedge (y = 1)) \vee (\neg \mathcal{P}_0(x) \wedge (y = 0))$$

We must verify that this formula does make χ_R a representable function.

Correctness. Fix a natural number n and set $m = \chi_R(n)$, which is therefore either 0 or 1.

- If $m = 1$, then $R(n)$ is true, so by assumption $\mathcal{P}\mathcal{A} \vdash \mathcal{P}_0(n)$. Therefore $\mathcal{P}\mathcal{A} \vdash \mathcal{P}_0(n) \wedge (m = 1)$, that is, $\mathcal{P}\mathcal{A} \vdash \mathcal{P}(n, m)$.
- If $m = 0$, then $R(n)$ is false, and similarly we obtain $\mathcal{P}\mathcal{A} \vdash \mathcal{P}(n, m)$.

In both cases, the correctness condition is satisfied.

Uniqueness. For uniqueness, we must show that, for every y : $\mathcal{P}\mathcal{A} \vdash \mathcal{P}(n, y) \rightarrow y = m$.

- Case $y \neq 0$ and $y \neq 1$. Then $\mathcal{P}(n, y)$ is false, so the implication holds.
- Case $y = 1$.
 - * If $R(n)$ is true, then $\mathcal{P}\mathcal{A} \vdash \mathcal{P}_0(n)$. Therefore $\mathcal{P}\mathcal{A} \vdash \mathcal{P}_0(n) \wedge (y = 1)$. Thus $\mathcal{P}\mathcal{A} \vdash \mathcal{P}_0(n) \wedge (y = 1) \wedge (y = \chi_R(n))$, so $\mathcal{P}\mathcal{A} \vdash (\mathcal{P}(n, 1) \rightarrow y = m)$ (because the statements on both sides of the implication are true).
 - * If $R(n)$ is false, then $\mathcal{P}\mathcal{A} \vdash \neg \mathcal{P}_0(n)$, so $\mathcal{P}(x, y)$ is false and the implication is true.
- Case $y = 0$. This is similar to the preceding case.

□

2.4. Important examples

The goal of the chapter is to prove that all primitive recursive functions are representable. We have already encountered the strategy: it is enough to prove this for the three basic functions and then show that the property is preserved under composition and recursion.

Proposition 2.

The basic functions Z , S , and the projections P_i^p are representable functions.

Proof. There is not much to prove. We simply need to provide a suitable formula $\mathcal{P}(\mathbf{x}, y)$.

- Consider the successor function S , defined by $S(n) = n + 1$. Set $\mathcal{P}(x, y) : y = S(x)$ (which is a formula in $\mathcal{L}_0$).

For a fixed $n \in \mathbb{N}$, writing $m = S(n) = n + 1$, we have:

$$\mathcal{P}(\bar{n}, y) \leftrightarrow y = S(\bar{n}) \leftrightarrow y = \overline{S(n)} \leftrightarrow y = \bar{m}$$

These equivalences are true in any logical system; the Peano axioms are not needed. Thus S is a representable function.

- For the zero function Z , the formula is $\mathcal{P}(x, y) : y = 0$.
- For the projection P_i^p , the formula is $\mathcal{P}(\mathbf{x}, y) : y = x_i$.

□

Proposition 3.

Let $f_1, \dots, f_q : \mathbb{N}^p \rightarrow \mathbb{N}$ and $g : \mathbb{N}^q \rightarrow \mathbb{N}$ be representable functions. Then $h = g(f_1, \dots, f_q)$ is also representable.

Thus representability is preserved under composition. If we prove the same for recursion, we will have the desired result. However, this latter preservation property will be much more difficult to establish and will occupy us for the rest of the chapter.

For now, we focus on the proof of the proposition above. As already mentioned, we will give the proof in a minimal setting, but it is worth discussing the general formula that makes h a representable function:

$$\mathcal{P}_h(\mathbf{x}, y) : \exists w_1 \dots \exists w_q \quad \mathcal{P}_0(w_1, \dots, w_q, y) \wedge \mathcal{P}_1(\mathbf{x}, w_1) \wedge \dots \wedge \mathcal{P}_q(\mathbf{x}, w_q)$$

where $\mathcal{P}_0$ is the formula associated with the representable function g , and $\mathcal{P}_i$ is the formula for f_i ($i = 1, \dots, q$). The role of the variables w_i is particularly interesting: they store the intermediate result $w_i = f_i(\mathbf{n})$ so that it can be passed to g to compute $h(\mathbf{n}) = g(w_1, \dots, w_q) = g(f_1(\mathbf{n}), \dots, f_q(\mathbf{n}))$.

Proof. We prove only the case $p = 1$ and $q = 1$. Let $f_1 : \mathbb{N} \rightarrow \mathbb{N}$ be a representable function, represented by the formula $\mathcal{P}_1(x, w)$. Let $g : \mathbb{N} \rightarrow \mathbb{N}$ be a representable function, represented by the formula $\mathcal{P}_0(w, y)$. We want to show that the composition $h = g \circ f_1 : \mathbb{N} \rightarrow \mathbb{N}$ is representable.

Define the formula:

$$\mathcal{P}_h(x, y) : \exists w \quad \mathcal{P}_0(w, y) \wedge \mathcal{P}_1(x, w)$$

which will represent the function h . For $n \in \mathbb{N}$, set $k = f_1(n)$ and $m = g(k) = g(f_1(n)) = h(n)$.

- **Correctness.** By assumption, $\mathcal{P}\mathcal{A} \vdash \mathcal{P}_0(k, m)$ and $\mathcal{P}\mathcal{A} \vdash \mathcal{P}_1(n, k)$. Therefore $\mathcal{P}\mathcal{A} \vdash \exists w (\mathcal{P}_0(w, m) \wedge \mathcal{P}_1(n, w))$ (it is enough to take $w = k$). Hence $\mathcal{P}\mathcal{A} \vdash \mathcal{P}_h(n, m)$.
- **Uniqueness.** By assumption, $\mathcal{P}\mathcal{A}$ proves the following two implications:

$$\forall w \quad (\mathcal{P}_1(n, w) \rightarrow w = k)$$

$$\text{and } \forall y \quad (\mathcal{P}_0(k, y) \rightarrow y = m)$$

Now suppose that $\mathcal{P}_h(n, y)$ holds. Then there exists w such that $\mathcal{P}_0(w, y)$ and $\mathcal{P}_1(n, w)$. Since $\mathcal{P}_1(n, w)$ holds, we have $w = k$. Therefore $\mathcal{P}_0(k, y)$ holds, which implies $y = m$. Thus the uniqueness condition is satisfied: $\mathcal{P}_h(n, y) \rightarrow y = m$.

□

Proposition 4.

The remainder function, which maps a and b to the remainder $\text{remainder}(b, a)$ of the Euclidean division of b by a , is representable.

Proof. The formula is:

$$\mathcal{P}(a, b, r) : [a = 0 \wedge b = r] \vee [\exists q \leq b \quad (b = qa + r) \wedge (r < a)]$$

We prove correctness and uniqueness. Indeed, Peano arithmetic proves the existence and uniqueness of Euclidean division. In fact, this would also hold in Robinson arithmetic $\mathcal{Q}$, without using induction.

Remark: “ $\exists q \leq b \quad \mathcal{P}$ ” means “ $\exists q \quad (q \leq b) \wedge \mathcal{P}$ ”.

□

3. Gödel's β function

The goal of this section is to study Gödel's β function. This ingenious construction stores any list of natural numbers $(n_1, \dots, n_l)$ using only two natural numbers a and b .

This construction acts as memory. Indeed, to compute a sequence recursively, we must compute $u_1, u_2, \dots, u_{l-1}$ before we can compute u_l .

3.1. Theorem

Theorem 2.

There exists a function $\beta : \mathbb{N}^3 \rightarrow \mathbb{N}$ that is primitive recursive and representable, and such that, for every natural number l and every sequence $(n_1, \dots, n_l)$ of natural numbers, there exist $a, b \in \mathbb{N}$ such that:

$$\beta(a, b, i) = n_i \quad \text{for } i = 1, \dots, l.$$

The function β is ultimately quite simple, since $\beta(a, b, i)$ will be the remainder of the Euclidean division of b by $a \times i + 1$. The difficult part is explaining the existence of these a and b . For this, we will need a well-known theorem from arithmetic.

3.2. Chinese Remainder Theorem

Theorem 3 (Chinese Remainder Theorem).

Let $a_1, \dots, a_l$ be pairwise coprime positive integers. Let $n_1, \dots, n_l$ be arbitrary natural numbers. There exists $x \in \mathbb{N}$ such that, for every $i = 1, \dots, l$:

$$x \equiv n_i \pmod{a_i}$$

For two natural numbers, the statement is as follows:

Theorem 4.

Let a and b be coprime positive integers. For every $\alpha, \beta \in \mathbb{N}$, there exists $x \in \mathbb{N}$ such that:

$$\begin{cases} x \equiv \alpha \pmod{a} \\ x \equiv \beta \pmod{b} \end{cases}$$

From an algebraic viewpoint, we even have a group isomorphism:

$$\mathbb{Z}/ab\mathbb{Z} \simeq \mathbb{Z}/a\mathbb{Z} \times \mathbb{Z}/b\mathbb{Z}$$

Proof. Define:

$$\phi : \mathbb{Z}/ab\mathbb{Z} \longrightarrow \mathbb{Z}/a\mathbb{Z} \times \mathbb{Z}/b\mathbb{Z}$$

by $\phi(\bar{x}^{ab}) = (\bar{x}^a, \bar{x}^b)$, where $\bar{x}^m$ denotes the residue class of the natural number x modulo m .

- ϕ is well-defined. Indeed, if $\bar{x}^{ab} = \bar{y}^{ab}$, then $x = y + kab$, so $\bar{x}^a = \bar{y}^a$ and $\bar{x}^b = \bar{y}^b$. Therefore $\phi(\bar{x}^{ab}) = \phi(\bar{y}^{ab})$.
- ϕ is a group homomorphism: $\phi(\bar{x}^{ab} + \bar{y}^{ab}) = \phi(\bar{x}^{ab}) + \phi(\bar{y}^{ab})$.
- ϕ is injective. If $\phi(\bar{x}^{ab}) = (\bar{0}^a, \bar{0}^b)$, then $\bar{x}^a = \bar{0}^a$, so $x = k_1a$, and $\bar{x}^b = \bar{0}^b$, so $x = k_2b$. Thus $k_1a = k_2b$. In particular, b divides k_1a . By assumption, a and b are coprime, so Euclid's lemma implies that b divides k_1 , and hence $k_1 = k_3b$. Thus $x = k_1a = k_3ba$. Therefore x is a multiple of ab , and the class $\bar{x}^{ab}$ is $\bar{0}^{ab}$. Hence the map ϕ is injective.
- Since the domain and codomain have the same cardinality (namely ab), the map is bijective.

- Thus, for a given (α, β) , there exists a preimage x under ϕ , which says exactly that:

$$\begin{cases} x \equiv \alpha \pmod{a} \\ x \equiv \beta \pmod{b} \end{cases}$$

□

3.3. Proof

Construction of a and b . Let $n_1, \dots, n_l \in \mathbb{N}$ be given. Choose $m \in \mathbb{N}$ such that $m \geq l$ and $m! \geq n_i$ ($i = 1, \dots, l$). Set $a = m!$. The other natural number b will be constructed using the Chinese Remainder Theorem.

- The natural numbers $ai + 1$ are pairwise coprime. Indeed, if a prime number p divides both $ai + 1$ and $aj + 1$ (with $i < j$), then, by subtraction, $p|a(j - i)$. Thus $p|a$ or $p|(j - i)$. Now $j - i < l \leq m$ and $a = m!$, so in either case $p \leq m$. It follows that $p|m!$ (that is, $p|a$), but since $p|ai + 1$, we then have $p|1$. This is a contradiction.
- We have $ai + 1 > n_i$ because $n_i \leq m! = a$ ($i = 1, \dots, l$).
- By the Chinese Remainder Theorem, there exists $b \in \mathbb{N}$ such that:

$$b \equiv n_i \pmod{ai + 1} \quad \text{for every } i = 1, \dots, l$$

Construction of β . This justifies the construction of β : $\beta(a, b, i)$ is the remainder of the Euclidean division of b by $ai + 1$. Indeed, this remainder is characterized by $b \equiv n_i \pmod{ai + 1}$ and $n_i < ai + 1$, as above.

β is primitive recursive. The first part ensures that this coding exists, even though it is somewhat complicated. Decoding, however, remains simple:

$$\beta(a, b, i) = \text{remainder}(b, ai + 1)$$

This remainder is indeed given by a primitive recursive function. In fact:

$$\text{remainder}(b, a) = \min\{r \leq b \mid \exists q \leq b \quad b = qa + r\}$$

The condition “ $\exists q \leq b \quad b = qa + r$ ” is a primitive recursive predicate because the quantification is bounded. The remainder is therefore primitive recursive as well, since it is obtained by bounded minimization. The bound on the minimum is not the natural one, since a Euclidean division generally has $r < a$, but the bound b also covers the exceptional case $a = 0$.

β is representable. The formula to consider is:

$$\mathcal{B}(a, b, i, r) : (\exists q \leq b \quad b = q(ai + 1) + r) \wedge (r < ai + 1)$$

In other words, r is defined as the remainder of the Euclidean division of b by $ai + 1$. We have seen that the remainder function is representable.

Abuse of notation: a, b, i, r are not variables of the language $\mathcal{L}_0$; they should be replaced by $x_1, x_2, \dots$, but the formulas would become less readable.

Note. The advantage of coding with the function β is that it uses only addition and multiplication, which are the only operations allowed by the language $\mathcal{L}_0$.

Otherwise, to encode a sequence of natural numbers, we could have used the prime-number coding $2^{n_1} 3^{n_2} 5^{n_3} \dots$. But in $\mathcal{L}_0$, we are not yet allowed to compute powers: for example, to compute x^n , we successively compute $x, x^2, x^3 \dots$ up to x^n . We therefore need recursion; in other words, we must be able to store intermediate results. But we do not yet know that recursion preserves representability, since that is exactly what we want to prove.

Once we have established the representability of primitive recursive functions using the function β , we will be able to use (finite) lists of numbers for Gödel coding in the language $\mathcal{L}_0$.

4. Primitive recursive functions are representable

Let us now tackle the final goal of the chapter.

Theorem 5.

Every primitive recursive function $f : \mathbb{N}^p \rightarrow \mathbb{N}$ is representable.

4.1. Recursion

To prove that primitive recursive functions are representable, it remains to show that representability is preserved under recursion.

Proposition 5.

Let $g : \mathbb{N}^p \rightarrow \mathbb{N}$ and $h : \mathbb{N}^{p+2} \rightarrow \mathbb{N}$ be representable functions. The function $f : \mathbb{N}^{p+1} \rightarrow \mathbb{N}$ defined by the initial function g and the recursion relation h is representable.

This proposition implies the theorem above, since we have already shown that the three basic functions Z , S , and P_i^p are representable and that composition preserves representability.

4.2. The formula

We will prove the case $p = 1$ to keep the notation as simple as possible.

- $g : \mathbb{N} \rightarrow \mathbb{N}$ is represented by $\mathcal{P}_g(x, u)$.
- $h : \mathbb{N}^3 \rightarrow \mathbb{N}$ is represented by $\mathcal{P}_h(x, y, u, v)$.
- $f : \mathbb{N}^2 \rightarrow \mathbb{N}$ is defined by $f(x, 0) = g(x)$ and $f(x, y + 1) = h(x, y, f(x, y))$.

The goal is to find a formula that represents f . The formula $\mathcal{P}_f(x, y, z)$ says: “there exists a finite sequence (encoded by natural numbers a and b) whose first term is $z_0 = g(x)$, each transition satisfies the recursion relation $z_{i+1} = h(x, i, z_i)$, and whose last term is $z_y = z$ ”.

We must translate this sentence into $\mathcal{L}_0$. To do so, we use Gödel’s β function. We make a minor modification to obtain convenient indexing: we now assume that $\beta(a, b, i)$ encodes a sequence $n_0, n_1, n_2, \dots$ (the indices start at 0). We have seen that the function β is representable; denote the corresponding formula by $\mathcal{B}(a, b, i, r)$. Recall that the role of the function β is to store a list of numbers, which here will be the terms z_i of the sequence.

Here is the formula $\mathcal{P}_f$ in the language $\mathcal{L}_0$:

$$\mathcal{P}_f(x, y, z) : \quad \exists a \exists b \quad \text{Init}(a, b, x) \wedge \text{Trans}(a, b, x, y) \wedge \mathcal{B}(a, b, y, z)$$

Fix x (in fact, this variable x serves no purpose), set $z_0 = g(x)$, $z_1 = h(x, 0, z_0)$, $z_2 = h(x, 1, z_1), \dots$, and aim to compute z_y in order to verify that it is indeed the value $z = f(x, y)$.

With $\mathcal{B}$ denoting the formula associated with Gödel’s β function, the three parts of the formula $\mathcal{P}_f$ are defined as follows:

- $\text{Init}(a, b, x) : \exists u \ B(a, b, 0, u) \wedge \mathcal{P}_g(x, u)$

Thus $u = \beta(a, b, 0)$ (the first term in memory) and at the same time $u = g(x)$, so $z_0 = g(x) = \beta(a, b, 0)$.

- $\text{Trans}(a, b, x, y) : \forall i < y \ \exists u \exists v \ B(a, b, i, u) \wedge B(a, b, i + 1, v) \wedge \mathcal{P}_h(x, i, u, v)$

The conjunctions give $u = \beta(a, b, i)$, $v = \beta(a, b, i + 1)$, and $v = h(x, i, u)$. Thus, if $z_i = \beta(a, b, i) = u$, then $z_{i+1} = h(x, i, z_i) = h(x, i, u) = v = \beta(a, b, i + 1)$. By induction, we therefore have $z_i = \beta(a, b, i)$ for every $i \leq y$.

- $B(a, b, y, z)$ means that $z = \beta(a, b, y)$, that is, $z = z_y$.

4.3. Correctness

Consider n (for the variable x), l (for the variable y), and $m = f(n, l)$ (for the variable z). Correctness means that $\mathcal{PA} \vdash \mathcal{P}_f(n, l, m)$.

This time, set $z_0 = g(n)$ and $z_{i+1} = h(n, i, z_i)$ for $i = 0, \dots, l - 1$. Final goal: show that $m = z_l$.

We know that there exist a, b encoding the sequence and satisfying $\beta(a, b, i) = z_i$ for $i = 0, \dots, l$.

- We have $\beta(a, b, 0) = z_0$, and therefore $\mathcal{PA} \vdash B(a, b, 0, z_0) \wedge \mathcal{P}_g(n, z_0)$. Thus, setting $u = z_0$, we obtain $\mathcal{PA} \vdash \text{Init}(a, b, n)$.
- Next, for z_{i+1} , set $u = z_i$ and $v = z_{i+1}$. We have:

$$B(a, b, i, z_i) \wedge B(a, b, i + 1, z_{i+1}) \wedge \mathcal{P}_h(n, i, z_i, z_{i+1})$$

This holds for $i = 0, \dots, l - 1$, so $\mathcal{PA} \vdash \text{Trans}(a, b, n, l)$.

- Finally, $z_l = \beta(a, b, l) = f(n, l) = m$, and hence $B(a, b, l, m)$ holds.

4.4. Uniqueness

We must prove that:

$$\mathcal{PA} \vdash \forall z \ \mathcal{P}_f(n, l, z) \rightarrow z = m$$

that is, $z = z_l$.

By induction on l :

- Base case 0: The formula $\mathcal{P}_f(n, 0, z)$ (with $m = f(n, 0) = g(n) = z_0$) reduces to:

$$\exists a \exists b \ \text{Init}(a, b, n) \wedge B(a, b, 0, z)$$

(because the condition “ $\forall i < 0$ ” is empty, so the predicate Trans is true). Therefore $z_0 = \beta(a, b, 0) = z$, and hence $z = m$, as desired.

- **Inductive step.** Fix i such that $0 \leq i < l$ and assume uniqueness holds at rank i :

$$\forall u \ \mathcal{P}_f(n, i, u) \rightarrow u = z_i$$

Let z be such that $\mathcal{P}_f(n, i + 1, z)$ holds. There exist witnesses a, b satisfying $\text{Init}(a, b, n)$, $\text{Trans}(a, b, n, i + 1)$, and $B(a, b, i + 1, z)$. Restricting $\text{Trans}(a, b, n, i + 1)$ to indices $k < i$ shows that these same witnesses a, b along with the value $u = \beta(a, b, i)$ satisfy $\mathcal{P}_f(n, i, u)$. The induction hypothesis then yields $u = z_i$.

The transition step at rank i yields $v = \beta(a, b, i + 1)$ such that $\mathcal{P}_h(n, i, u, v)$. By representability of h , we have $v = h(n, i, u) = h(n, i, z_i) = z_{i+1}$. Finally, since $B(a, b, i + 1, z)$ requires $z = \beta(a, b, i + 1) = v$, we conclude that $z = z_{i+1}$.

- By the induction principle (in $\mathcal{PA}$), the statement is true for every $i \leq l$, and uniqueness is proved.

Remark. It is possible to do everything in Robinson arithmetic $\mathcal{Q}$, because our induction is in fact limited to a finite number of steps (since l is fixed).

This completes the proof that, for primitive recursive functions, recursion preserves representability.

4.5. Application to predicates

We knew that the three basic primitive recursive functions were representable and that composition preserved representability. We have just completed the difficult proof that representability is also preserved under recursion. By the definition of a primitive recursive function, we deduce Theorem 5: “All primitive recursive functions are representable.”

In particular, we know that a predicate is representable if and only if its characteristic function is representable. We therefore obtain Theorem 1, announced at the very beginning: “Every primitive recursive predicate is representable.”