Logic, Computability and Incompleteness

Arithmetical Definability

Wolfgang Schwarz

14 November 2025

Recap

Recap

An algorithm is a step-by-step procedure for converting inputs into outputs.

A finite algorithm can't distinguish uncountably many different inputs or outputs.

Any countable domain on which it might operate can be effectively encoded into the natural numbers.

So we can focus on algorithms that operate on $\mathbb{N}$.

Recap

An algorithm is a step-by-step procedure for converting numbers into numbers.

An algorithm computes a function on $\mathbb{N}$.

A function is **computable** iff there is an algorithm that computes it.

Recap

A function is **computable** iff there is an algorithm that computes it.

Two formal models of computable functions:

- A (total) function is computable iff it is computed by a Turing machine.
- A (total) function is computable iff it can be built up from the base functions z, s, p_i^n using composition, regular minimization, and primitive recursion.

The two models are equivalent.

Recap

A relation is **computable** (=**decidable**) iff there is an algorithm that decides whether any given numbers stand in the relation.

A relation is computable iff its characteristic function is computable.

Recap

We can reason about $\mathbb{N}$ in first-order theories.

The **language $\mathfrak{L}_A$ of arithmetic** has non-logical symbols 0, s, $+$, and $\times$.

Recap

Robinson's Q:

Q1	$\forall x \forall y\, (s(x) = s(y) \to x = y)$
Q2	$\forall x\, 0 \neq s(x)$
Q3	$\forall x\, (x \neq 0 \to \exists y\, x = s(y))$
Q4	$\forall x\, (x + 0 = x)$
Q5	$\forall x \forall y\, (x + s(y) = s(x + y))$
Q6	$\forall x\, (x \times 0 = 0)$
Q7	$\forall x \forall y\, (x \times s(y) = (x \times y) + x)$

Recap

First-Order Peano Arithmetic (PA):

Q1	$\forall x \forall y\, (s(x) = s(y) \to x = y)$
Q2	$\forall x\, 0 \neq s(x)$
Ind	$A(0) \wedge \forall x\, (A(x) \to A(s(x))) \to \forall x\, A(x)$
Q4	$\forall x\, (x + 0 = x)$
Q5	$\forall x \forall y\, (x + s(y) = s(x + y))$
Q6	$\forall x\, (x \times 0 = 0)$
Q7	$\forall x \forall y\, (x \times s(y) = (x \times y) + x)$

Arithmetical definability

Arithmetical definability

Claim: All computable functions and relations on $\mathbb{N}$ are definable in $\mathfrak{L}_A$.

- $x < y$ iff $\exists z\,(x + s(z) = y)$.
- x is prime iff
  $s(0) < x \wedge \forall y\big(\exists z\,(z \times y = x) \to (y = s(0) \vee y = x)\big)$.
- $x^2 = y$ iff $x \times x = y$.

Arithmetical definability

Claim: All computable functions and relations on $\mathbb{N}$ are definable in $\mathfrak{L}_A$.

Any computable function or relation can be defined in terms of addition, multiplication, and logical operations.

Example:

The relation that holds between a sequence of $\mathfrak{L}_A$-sentences $A_1, \ldots, A_n$ and a sentence B iff B is derivable from $A_1, \ldots, A_n$ from the axioms of PA.

Arithmetical definability

A formula $A(x, y)$ **defines** a 2-place relation R if for all $a, b \in \mathbb{N}$,

- $R(a, b)$ iff $\mathfrak{A} \Vdash A(\overline{a}, \overline{b})$.

($\overline{a}$ is the $\mathfrak{L}_A$-term for the numeral of a.)

E.g.: $\exists z\,(x + s(z) = y)$ defines $<$.

(Definability is a semantic notion.)

Arithmetical definability

A formula $A(x, y)$ **defines** a 1-place function f if for all $a, b \in \mathbb{N}$,

- $f(a) = b$ iff $\mathfrak{A} \Vdash A(\overline{a}, \overline{b})$.

E.g.: $x \times x = y$ defines the squaring function.

Arithmetical definability

All computable functions and relations on $\mathbb{N}$ are definable in $\mathfrak{L}_A$.

Proof strategy:

- Every computable function can be built up from the base functions using composition, regular minimization, and primitive recursion.
- The base functions (zero, successor, projections) are definable in $\mathfrak{L}_A$.
- Definability in $\mathfrak{L}_A$ is preserved under composition, regular minimization, and primitive recursion.

Representability

Representability

A formula $A(x, y)$ **represents** a 2-place relation R **in** a theory T if for all $a, b \in \mathbb{N}$:

- If $R(a, b)$, then $\vdash_T A(\overline{a}, \overline{b})$.
- If $\neg R(a, b)$, then $\vdash_T \neg A(\overline{a}, \overline{b})$.

E.g.: $\exists z\,(x + s(z) = y)$ represents $<$ in Q.

(Representability is a syntactic notion.)

Representability

A formula $A(x, y)$ **represents** a 1-place function f **in** a theory T if for all $a \in \mathbb{N}$:

- $\vdash_T A(\overline{a}, \overline{f(a)})$.
- $\vdash_T \forall y\, (A(\overline{a}, y) \rightarrow y = \overline{f(a)})$.

E.g.: $x + y = z$ represents the addition function in Q.

Representability

Claim: All computable functions and relations on $\mathbb{N}$ are representable in Q.

Since all axioms of Q are true in $\mathfrak{A}$, it follows that all computable functions and relations are definable in $\mathfrak{L}_A$.

- If $R(a, b)$, then $\vdash_T A(\overline{a}, \overline{b})$, then $\mathfrak{A} \Vdash A(\overline{a}, \overline{b})$.
- If $\neg R(a, b)$, then $\vdash_T \neg A(\overline{a}, \overline{b})$, then $\mathfrak{A} \nVdash A(\overline{a}, \overline{b})$.

Representability

Claim: All computable functions and relations on $\mathbb{N}$ are representable in Q.

Proof strategy:

- Every computable function can be built up from the base functions using composition, regular minimization, and primitive recursion.
- The base functions (zero, successor, projections) are representable in Q.
- Representability in Q is preserved under composition, regular minimization, and primitive recursion.

Representability

In the lecture notes, I

- go through the base functions and the operations ($z, s, p_i^n, \mathrm{Cn}, \mathrm{Pr}, \mathrm{Mn}$) one by one,
- check for each what a theory T needs to prove so that the result can be proved.

I arrive at six conditions, R1–R6.

Then I show that Q satisfies these conditions.

Representability

To show:

- The zero function is representable.
- The successor function is representable.
- The projection functions are representable.
- If f and g are representable, then so is $\mathrm{Cn}[f, g]$.
- If f and g are representable, then so is $\mathrm{Pr}[f, g]$.
- If f is representable, then so is $\mathrm{Mn}[f]$.

Most of this is easy.

Representability

To show:

- The zero function is representable.
- ...

$A(x, y)$ represents the zero function iff for all a:

(i) $\vdash_T A(\overline{a}, \overline{0})$,

(ii) $\vdash_T \forall y\,(A(\overline{a}, y) \rightarrow y = \overline{0})$.

Let $A(x, y)$ be $x = x \wedge y = 0$, or simply $y = 0$.

Representability

To show:

- ...
- If f and g are representable, then so is $\mathrm{Cn}[f, g]$.
- ...

If $A(x, y)$ represents f and $B(x, y)$ represents g then

$$\exists v \left(B(x, v) \wedge A(v, y) \right)$$

represents $\mathrm{Cn}[f, g]$.

Representability

To show:

- ...
- If f and g are representable, then so is $\mathrm{Pr}[f, g]$.
- ...

This one is hard.

Representability

Example: the factorial function f defined by

$$f(0) = 1$$
$$f(s(y)) = s(y) \times f(y)$$

This defines a sequence $f(0), f(1), f(2), \ldots$.

We construct a formula $F(x, y)$ saying that

- there is a number c that codes the sequence $f(0), f(1), \ldots, f(x)$, and
- y is the last element in that sequence.

$\exists z(\text{SEQ}(z, x) \wedge \text{ENTRY}(z, s(x), y))$.

Representability

We've shown:

All computable functions and relations are

- definable in $\mathfrak{L}_A$.
- representable in Q.