Logic II

Markus Lohrey

Universität Siegen

Summer 2017

Organizational matters

Information can be found at

`http://www.eti.uni-siegen.de/ti/lehre/ss17/logikii/`

e.g.,

- current version of the slides (german and english)
- exercise sheets for the tutorials

Literature recommendations:

- Schöning: Logik für Informatiker, Spektrum Akademischer Verlag
- Ebbinghaus, Flum, Thomas: Einführung in die mathematische Logik, Spektrum Akademischer Verlag

The **tutorials** will be organized by Danny Hucke.

Recapitulation from the course GTI

Definition (semi-decidable)

A language $L \subseteq \Sigma^*$ is semi-decidable if there exists an algorithm with the following properties:

For all $x \in \Sigma^*$:

- If $x \in L$, then the algorithm terminates on input x.
- If $x \notin L$, then the algorithm does not terminate on input x.

Equivalent notion: recursively enumerable.

Definition (recursively enumerable)

A language $L \subseteq \Sigma^*$ is recursively enumerable if there exists a computable total function $f : \mathbb{N} \to \Sigma^*$ such that $L = \{f(i) \mid i \in \mathbb{N}\}$.

Recapitulation from the course GTI

Definition (decidable and undecidable)

A language $L \subseteq \Sigma^*$ is decidable if there exists an algorithm with the following properties for all $x \in \Sigma^*$:

- If $x \in L$, then the algorithm terminates on input x with output "YES".
- If $x \notin L$, then the algorithm terminates on input x with output "NO".

A language $L \subseteq \Sigma^*$ is undecidable, if it is not decidable.

Theorem

A language $L \subseteq \Sigma^*$ is decidable if and only if L and $\Sigma^* \setminus L$ are both semi-decidable.

Recapitulation from the course Logic I

A formula F of predicate logic is

- satisfiable, if there exists a suitable structure $\mathcal{A}$ for F with $\mathcal{A} \models F$ (i.e., F is true in the structure $\mathcal{A}$).
- valid, if $\mathcal{A} \models F$ for every suitable structure $\mathcal{A}$ for F.

Corollary from the theorem of Gilmore

The set of unsatisfiable formulas of predicate logic is semi-decidable.

Corollary

The set of valid formulas of predicate logic is semi-decidable.

Proof: F is valid if and only if $\neg F$ is unsatisfiable.

Undecidability of predicate logic

In the next few hours, we will prove the following important theorem:

Church's theorem

The set of valid formulas of predicate logic is undecidable.

Corollary

The set of satisfiable formulas of predicate logic is not semi-decidable.

Proof: The set of unsatisfiable formulas is semi-decidable.

If the set of satisfiable formulas would be semi-decidable too, then it would be decidable.

Hence, the set of unsatisfiable formula and therefore also the set of valid formulas would be decidable. □

Register machines

We prove Church's theorem by a reduction to the halting problem for register machine programs.

Let $R_1, R_2, \ldots$ be names for registers.

Intuition: Every register stores a natural number.

A register machine program (RMP for short) P is a sequence of instructions $A_1; A_2; \ldots; A_l$, where A_l is the STOP instruction, and for all $1 \leq i \leq l-1$ the instruction A_i has one of the following forms:

- $R_j := R_j + 1$ for a $1 \leq j \leq l$
- $R_j := R_j - 1$ for a $1 \leq j \leq l$
- IF $R_j = 0$ THEN k_1 ELSE k_2 for $1 \leq j, k_1, k_2 \leq l$,

A configuration of P is a tuple $(i, n_1, \ldots, n_l) \in \mathbb{N}^{l+1}$ with $1 \leq i \leq l$.

Intuition: i is the index of the instruction that will be executed next and n_j is the current content of register R_j.

Register machines

For configurations $(i, n_1, \dots, n_l)$ und $(i', n'_1, \dots, n'_l)$ we write

$$(i, n_1, \dots, n_l) \to_P (i', n'_1, \dots, n'_l)$$

if and only if $1 \leq i \leq l-1$ and one of the following cases holds:

- $A_i = (R_j := R_j + 1)$ for a $1 \leq j \leq l$, $i' = i+1$, $n'_j = n_j + 1$, $n'_k = n_k$ for $k \neq j$.
- $A_i = (R_j := R_j - 1)$ for a $1 \leq j \leq l$, $i' = i+1$, $n_j = n'_j = 0$ or $(n_j > 0,\ n'_j = n_j - 1)$, and $n'_k = n_k$ for $k \neq j$.
- $A_i = (\text{IF } R_j = 0 \text{ THEN } k_1 \text{ ELSE } k_2)$ for a $1 \leq j, k_1, k_2 \leq l$, $n'_k = n_k$ for all $1 \leq k \leq l$, $i' = k_1$ if $n_j = 0$, $i' = k_2$ if $n_j > 0$.

We define

$$\begin{aligned}\text{HALT} = \{P \mid P = A_1; A_2; \dots; A_l \text{ is an RMP with } l \text{ instructions,}\\ (1, 0, \dots, 0) \to_P^* (l, n_1, \dots, n_l) \text{ for } n_1, \dots, n_l \geq 0\}\end{aligned}$$

Proof of Church's theorem

Register machine programs exactly correspond to the GOTO-programs from the GTI course.

There, we proved that Turing machines can be simulated by GOTO-programs (and vice versa).

Since the halting problem is undecidable for Turing machines started on the empty tape (Does a Turing machine, when started with blanks on the input tape, finally terminate?), we get:

Undecidability of the halting problem for RMPs

The set HALT is undecidable.

Remark: HALT is semi-decidable: Simulate the given RMP on the initial configuration $(1, 0, \ldots, 0)$ and stop, if the RMP arrives at the STOP-instruction.

Proof of Church's theorem

We prove Church's theorem, by constructing from a given RMP P a sentence F_P of predicate logic (formula without free variables) such that:

$$F_P \text{ is valid } \iff P \in \text{HALT}$$

Let $P = A_1; A_2; \dots; A_l$ be an RMP.

We fix the following symbols:

- $<$: binary predicate symbol
- c: constant
- f, g: unary function symbol
- R: $(l + 2)$-ary predicate symbol

Proof of Church's theorem

We define a structure $\mathcal{A}_P$ by the following case distinction:

Case 1: $P \notin$ HALT:

- universe $U_{\mathcal{A}_P} = \mathbb{N}$
- $<^{\mathcal{A}_P} = \{(n, m) \mid n < m\}$ (the ordinary linear order on $\mathbb{N}$)
- $c^{\mathcal{A}_P} = 0$
- $f^{\mathcal{A}_P}(n) = n + 1$, $g^{\mathcal{A}_P}(n+1) = n$, $g^{\mathcal{A}_P}(0) = 0$
- $R^{\mathcal{A}_P} = \{(s, i, n_1, \ldots, n_l) \mid (1, 0, \ldots, 0) \to_P^s (i, n_1, \ldots, n_l)\}$

Case 2: $P \in$ HALT:

Let t be such that $(1, 0, \ldots, 0) \to_P^t (l, n_1, \ldots, n_l)$ and $e = \max\{t, l\}$.

- universe $U_{\mathcal{A}_P} = \{0, 1, \ldots, e\}$
- $<^{\mathcal{A}_P} = \{(n, m) \mid n < m\}$ (the ordinary linear order on $\{0, 1, \ldots, e\}$)
- $c^{\mathcal{A}_P} = 0$
- $f^{\mathcal{A}_P}(n) = n + 1$ for $0 \leq n \leq e - 1$ and $f^{\mathcal{A}_P}(e) = e$.
- $g^{\mathcal{A}_P}(n+1) = n$ for $0 \leq n \leq e - 1$ and $g^{\mathcal{A}_P}(0) = 0$.
- $R^{\mathcal{A}_P} = \{(s, i, n_1, \ldots, n_l) \mid 0 \leq s \leq t, (1, 0, \ldots, 0) \to_P^s (i, n_1, \ldots, n_l)\}$

Proof of Church's theorem

In the following, we use the abbreviation $\overline{m}$ for the term $f^m(c)$.

We define the sentence G_P (in which the symbols $<, c, f, g$ and R occur) with the following properties:

(A) $\mathcal{A}_P \models G_P$

(B) For every model $\mathcal{A}$ of G_P the following holds:
if $(1, 0, \ldots, 0) \to_P^s (i, n_1, \ldots, n_l)$, then:

$$\mathcal{A} \models R(\overline{s}, \overline{i}, \overline{n_1}, \ldots, \overline{n_l}) \wedge \bigwedge_{q=0}^{s-1} \overline{q} < \overline{q+1}.$$

We define

$$G_P = G_0 \wedge R(\overline{0}, \overline{1}, \overline{0}, \ldots, \overline{0}) \wedge G_1 \wedge \cdots \wedge G_{l-1}$$

where the sentences $G_0, G_1, \ldots, G_{l-1}$ is defined as follows (next slides):

Proof of Church's theorem

G_0 expresses

- $<$ is a linear order with smallest element c,
- $x \leq f(x)$ and $g(x) \leq x$ for all x,
- for every x, which is not the largest element with respect to $<$, $f(x)$ is the direct successor of x, and
- for every x, which is not the smallest element c, $g(x)$ is the direct predecessor of x.

$$
\begin{aligned}
\forall x, y, z\ (\neg x < x) \wedge (x = y \vee x < y \vee y < x) \wedge ((x < y \wedge y < z) \rightarrow x < z) \\
\wedge\ (x = c \vee c < x) \\
\wedge\ (x = f(x) \vee x < f(x)) \\
\wedge\ (x = g(x) \vee g(x) < x) \\
\wedge\ \big(\exists u(x < u) \rightarrow (x < f(x) \wedge \forall u(x < u \rightarrow (u = f(x) \vee f(x) < u)))\big) \\
\wedge\ \big(\exists u(u < x) \rightarrow (g(x) < x \wedge \forall u(u < x \rightarrow (u = g(x) \vee u < g(x))))\big)
\end{aligned}
$$

Proof of Church's theorem

Remark: For every model $\mathcal{A}$ of G_0 we have:

- $\mathcal{A} \models g(c) = c$
- $\mathcal{A} \models \forall x\,(\exists u(x < u) \to g(f(x)) = x)$

Proof of Church's theorem

G_i for $1 \leq i \leq l-1$ describes the effect of the instruction A_i.

Case 1: $A_i = (R_j := R_j + 1)$. Let

$$G_i = \forall x \forall x_1 \cdots \forall x_l \Big(R(x, \overline{i}, x_1, \ldots, x_l) \to$$
$$(x < f(x) \wedge R(f(x), \overline{i+1}, x_1, \ldots, x_{j-1}, f(x_j), x_{j+1}, \ldots, x_l)) \Big)$$

Case 2: $A_i = (R_j := R_j - 1)$. Let

$$G_i = \forall x \forall x_1 \cdots \forall x_l \Big(R(x, \overline{i}, x_1, \ldots, x_l) \to$$
$$(x < f(x) \wedge R(f(x), \overline{i+1}, x_1, \ldots, x_{j-1}, g(x_j), x_{j+1}, \ldots, x_l)) \Big)$$

Proof of Church's theorem

Case 3: $A_i = (\text{IF } R_j = 0 \text{ THEN } k_1 \text{ ELSE } k_2)$ for $1 \leq j, k_1, k_2 \leq l$.
Let

$$G_i = \forall x \forall x_1 \cdots \forall x_l \Big(R(x, \overline{i}, x_1, \ldots, x_l) \rightarrow (x < f(x) \wedge \\ (x_j = c \wedge R(f(x), \overline{k_1}, x_1, \ldots, x_l)) \vee \\ (x_j > c \wedge R(f(x), \overline{k_2}, x_1, \ldots, x_l))) \Big)$$

Statement (A) follows immediately from the definition of $\mathcal{A}_P$ and G_P.

Property (B) is shown by induction on s.

Base case: $s = 0$. Assume that $(1, 0, \ldots, 0) \rightarrow_P^0 (i, n_1, \ldots, n_l)$, i.e., $i = 1$ and $n_1 = n_2 = \cdots = n_l = 0$.

$\mathcal{A} \models G_P$ implies $\mathcal{A} \models R(\overline{0}, \overline{1}, \overline{0}, \ldots, \overline{0})$, i.e., $\mathcal{A} \models R(\overline{s}, \overline{i}, \overline{n_1}, \ldots, \overline{n_l})$.

Proof of Church's theorem

Induction step: Let $s > 0$ and assume that (B) holds for $s-1$.

Let $(1, 0, \dots, 0) \rightarrow_P^s (i, n_1, \dots, n_l)$.

Then, there exist $j, m_1, \dots, m_l$ with

$$(1, 0, \dots, 0) \rightarrow_P^{s-1} (j, m_1, \dots, m_l) \rightarrow_P (i, n_1, \dots, n_l).$$

The induction hypothesis implies

$$\mathcal{A} \models R(\overline{s-1}, \overline{j}, \overline{m_1}, \dots, \overline{m_l}) \wedge \bigwedge_{q=0}^{s-2} \overline{q} < \overline{q+1}.$$

We make a case distinction concerning the instruction A_j. We only consider the case that A_j has the form $R_k := R_k - 1$.

Thus, $i = j+1$, $n_1 = m_1, \dots, n_{k-1} = m_{k-1}$, $n_{k+1} = m_{k+1}, \dots, n_l = m_l$, ($n_k = m_k = 0$ or $m_k > 0$ and $n_k = m_k - 1$).

Proof of Church's theorem

$\mathcal{A} \models G_j$ implies

$$\mathcal{A} \models \forall y, y_1, \ldots, y_l \Big(R(y, \overline{j}, y_1, \ldots, y_l) \rightarrow$$
$$(y < f(y) \wedge R(f(y), \overline{j+1}, y_1, \ldots, y_{k-1}, g(y_k), y_{k+1}, \ldots, y_l)) \Big).$$

Since $\mathcal{A} \models R(\overline{s-1}, \overline{j}, \overline{m_1}, \ldots, \overline{m_l})$, we get

$$\mathcal{A} \models \overline{s-1} < f(\overline{s-1}) \wedge$$
$$R(f(\overline{s-1}), \overline{j+1}, \overline{m_1}, \ldots, \overline{m_{k-1}}, g(\overline{m_k}), \overline{m_{k+1}}, \ldots, \overline{m_l}),$$

i.e.,

$$\mathcal{A} \models \overline{s-1} < \overline{s} \wedge R(\overline{s}, \overline{i}, \overline{n_1}, \ldots, \overline{n_{k-1}}, g(\overline{m_k}), \overline{n_{k+1}}, \ldots, \overline{n_l}).$$

Proof of Church's theorem

From $\mathcal{A} \models \overline{s-1} < \overline{s}$ we get

$$\mathcal{A} \models \bigwedge_{q=0}^{s-1} \overline{q} < \overline{q+1}.$$

Moreover, $\mathcal{A} \models G_0$ implies $\mathcal{A} \ \models \ g(\overline{m_k}) = \overline{n_k}$.

Thus, we have $\mathcal{A} \ \models \ R(\overline{s}, \overline{i}, \overline{n_1}, \ldots, \overline{n_l})$.

We proved (A) and (B).

Proof of Church's theorem:

Let $F_P \ = \ (G_P \to \exists x \exists x_1 \cdots \exists x_l R(x, \overline{l}, x_1, \ldots, x_l))$

Claim: F_P is valid $\iff P \in \mathrm{HALT}$.

Proof of Church's theorem

If F_P is valid, then $\mathcal{A}_P \models F_P$.

From (A) we get $\mathcal{A}_P \models \exists x \exists x_1 \cdots \exists x_l R(x, \bar{l}, x_1, \ldots, x_l)$.

Thus, there exist $s, n_1, \ldots, n_l \geq 0$ with $(s, l, n_1, \ldots, n_l) \in R^{\mathcal{A}_P}$.

We get $P \in \mathrm{HALT}$.

Now assume that $P \in \mathrm{HALT}$ and $(1, 0, \ldots, 0) \rightarrow_P^s (l, n_1, \ldots, n_l)$.

Let $\mathcal{A}$ be a structure with $\mathcal{A} \models G_P$.

From (B) we get $\mathcal{A} \models R(\overline{s}, \bar{l}, \overline{n_1}, \ldots, \overline{n_l})$.

Thus, F_P valid. $\square$

Trachtenbrot's theorem

A formula F is finitely satisfiable if and only if F has a finite model (a model with a finite universe), otherwise, F is finitely unsatisfiable.

Lemma

The set of finitely satisfiable formulas is semi-decidable.

Proof:

Let $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \ldots$ be a systematic enumeration of all finite structures in which only the finitely many predicate symbols and function symbols that appear in F are interpreted.

The following algorithm terminates if and only if F is finitely satisfiable:

$i := 1$;
while true do
    if $\mathcal{A}_i \models F$ **then STOP else** $i := i + 1$
end

□

Trachtenbrot's theorem

A formula F is finitely valid if and only if every finite structure that is suitable for F is a model of F.

Example: The formula

$$\forall x \forall y (f(x) = f(y) \to x = y) \ \leftrightarrow \ \forall y \exists x (f(x) = y)$$

is not valid but finitely valid.

Trachtenbrot's theorem

The set of finitely satisfiable formulas is undecidable.

Corollary

The set of finitely unsatisfiable formulas and the set of finitely valid formulas are not semi-decidable.

Trachtenbrot's theorem

Proof of Trachtenbrot's theorem:

We reuse the construction from the proof of Church's theorem.

Claim: G_P is finitely satisfiable $\iff P \in \mathrm{HALT}$.

(1) Assume that $P \in \mathrm{HALT}$.

Then, $\mathcal{A}_P$ is finite and (A) implies $\mathcal{A}_P \models G_P$.

Hence, G_P is finitely satisfiable.

Trachtenbrot's theorem

(2) Assume that G_P is finitely satisfiable.

Let $\mathcal{A}$ be a finite structure with $\mathcal{A} \models G_P$.

Assume that $P \not\in$ HALT.

Then, for every $s \geq 0$ there exist $i, n_1, \ldots, n_l$ with $(1, 0, \ldots, 0) \to_P^s (i, n_1, \ldots, n_l)$.

(B) implies $\mathcal{A} \models \overline{i} < \overline{i+1}$ for all $i \geq 0$.

Since $<^{\mathcal{A}}$ is a linear order (since $\mathcal{A} \models G_0$) the set $\{\mathcal{A}(\overline{i}) \mid i \geq 0\}$ is infinite, which is a contradiction. $\square$

(Un)decidable theories

Let $\mathcal{A}$ be a structure, where the domain of the interpretation function $I_{\mathcal{A}}$ is finite and does not contain any variables.

Let $f_1, \ldots, f_n, R_1, \ldots, R_m$ be the domain of $I_{\mathcal{A}}$.

We identify $\mathcal{A}$ with the tuple $(U^{\mathcal{A}}, f_1^{\mathcal{A}}, \ldots, f_n^{\mathcal{A}}, R_1^{\mathcal{A}}, \ldots, R_m^{\mathcal{A}})$, for which we also write $(U^{\mathcal{A}}, f_1, \ldots, f_n, R_1, \ldots, R_m)$.

Definition

The theorie of $\mathcal{A}$ is the set of formulas

$$\mathrm{Th}(\mathcal{A}) = \{F \mid F \text{ is a sentence, } \mathcal{A} \text{ is suitable for } F, \, \mathcal{A} \models F\}.$$

We are interested in the question, whether a structure has a decidable theory.

(Un)decidable theories

Theorem

Let $\mathcal{A}$ be a structure. Then $\mathsf{Th}(\mathcal{A})$ *is decidable if and only if* $\mathsf{Th}(\mathcal{A})$ *is semi-decidable.*

Proof: Let $\mathsf{Th}(\mathcal{A})$ be semi-decidable and let F be a suitable sentence.

We either have $F \in \mathsf{Th}(\mathcal{A})$ or $\neg F \in \mathsf{Th}(\mathcal{A})$.

Hence, we can run in parallel a semi-decision procedure for $\mathsf{Th}(\mathcal{A})$ on input F and $\neg F$.

For either F or $\neg F$ the algorithm has to terminate. $\square$

(Un)decidable theories

For the question, whether a structure has a decidable theory, we can restrict to so called relational structures.

A structure $\mathcal{A} = (A, f_1, \ldots, f_n, R_1, \ldots, R_m)$ is relational, if $n = 0$.

For a structure $\mathcal{A} = (A, f_1, \ldots, f_n, R_1, \ldots, R_m)$ we define

$$\mathcal{A}_{\text{rel}} = (A, P_1, \ldots, P_n, R_1, \ldots, R_m)$$

where

$$P_i = \{(a_1, \ldots, a_n, a) \mid f_i(a_1, \ldots, a_n) = a\}.$$

Lemma

$\mathrm{Th}(\mathcal{A})$ *is decidable if and only if* $\mathrm{Th}(\mathcal{A}_{rel})$ *is decidable.*

Proof: Excercise.

Undecidability of arithmetic (following Ebbinhaus,Flum,Thomas)

Theorem (Gödel 1931)

$\mathrm{Th}(\mathbb{N}, +, \cdot)$ is undecidable.

Corollary

$\mathrm{Th}(\mathbb{N}, +, \cdot)$ is not semi-decidable, i.e., not recursively enumerable.

For the proof we reduce the set HALT of terminating RMPs to $\mathrm{Th}(\mathbb{N}, +, \cdot)$.

In order to simplify the technical details of the proof, we consider $\mathrm{Th}(\mathbb{N}, +, \cdot, s, 0)$ with $s(n) = n + 1$.

Excercise: $\mathrm{Th}(\mathbb{N}, +, \cdot, s, 0)$ is undecidable if and only if $\mathrm{Th}(\mathbb{N}, +, \cdot)$ is undecidable .

Undecidability of arithmetic

Let $P = A_1; A_2; \cdots ; A_l$ be a RMP, which contains the registers $R_1, \dots, R_l$.

We construct an arithmetical formula F_P with the free variables $x, x_1, \dots, x_l$, such that for all $1 \le i \le l$ and $n_1, \dots, n_l \in \mathbb{N}$ the following two statements are equivalent:

- $(\mathbb{N}, +, \cdot, s, 0)_{[x/i,\ x_1/n_1, \dots, x_l/n_l]} \models F_P$
- $(1, 0, \dots, 0) \to_P^* (i, n_1, \dots, n_l)$

It then follows: $P \in \mathrm{HALT} \iff (\mathbb{N}, +, \cdot, s, 0) \models \exists x_1 \cdots \exists x_l\ F_P[x/s^l(0)]$.

Undecidability of arithmetic

Intuitively, the formula F_P says:

There exist $s \geq 0$ and configurations $C_0, C_1, \dots, C_s$ such that:

- $C_0 = (1, 0, \dots, 0)$
- $C_s = (x, x_1, \dots, x_l)$
- $C_i \to_P C_{i+1}$ for all $0 \leq i \leq s-1$

We can encode the $(l+1)$-tuple $C_0, C_1, \dots, C_s$ by a single $(s+1)(l+1)$-tuple and have to express the following, where $k = l + 1$:

There are $s \geq 0$ and a tuple $(y_0, y_1, \dots, y_{k-1},\ y_k, y_{k+1}, \dots, y_{2k-1}, \dots, y_{sk}, y_{sk+1}, \dots, y_{(s+1)k-1})$ with:

- $y_0 = 1,\ y_1 = 0, \dots, y_{k-1} = 0$
- $y_{sk} = x,\ y_{sk+1} = x_1, \dots, y_{(s+1)k-1} = x_l$
- $(y_{ik}, \dots, y_{(i+1)k-1}) \to_P (y_{(i+1)k}, \dots, y_{(i+2)k-1})$ for all $0 \leq i \leq s-1$

Undecidability of arithmetic

If one wants to express this directly by an arithmetical formula, then one faces the problem that one cannot quantify over sequences of numbers ($\exists y \exists x_1 \cdots \exists x_y$ is not allowed).

In order to simulate quantification over sequences of numbers (of arbitrary length) by quantification over numbers, we use Gödel's β-function.

Lemma

There is a function $\beta : \mathbb{N}^3 \to \mathbb{N}$ such that:

- For every sequence $(a_0, \ldots, a_r)$ over $\mathbb{N}$ there are $t, p \in \mathbb{N}$ such that $\beta(t, p, i) = a_i$ for all $0 \leq i \leq r$.
- There is an arithmetical formula B with free variables v, x, y, z such that for all $t, p, i, a \in \mathbb{N}$ the following holds:

$$(\mathbb{N}, +, \cdot, s, 0)_{[v/t,\ x/p,\ y/i,\ z/a]} \models B \iff \beta(t, p, i) = a$$

In other words: β is arithmetically definable.

Undecidability of arithmetic

Proof of the lemma:

Let $(a_0, \ldots, a_r)$ be a sequence over $\mathbb{N}$.

Let p be a prime number such that $p > r + 1$ and $p > a_i$ for all i.

Moreover let

$$t = 1p^0 + a_0 p^1 + 2p^2 + a_1 p^3 + \cdots + (i+1)p^{2i} + a_i p^{2i+1} + \cdots + (r+1)p^{2r} + a_r p^{2r+1}.$$

Thus, $(1, a_0, 2, a_1, \ldots, (i+1), a_i, \ldots, (r+1), a_r)$ is the base-p representation of t.

Claim: For all $a \in \mathbb{N}$ and all $0 \leq i \leq r$ we have $a = a_i$ if and only if there are $b_0, b_1, b_2 \in \mathbb{N}$ with:

(a) $t = b_0 + b_1((i+1) + ap + b_2 p^2)$

(b) $a < p$

(c) $b_0 < b_1$

(d) There is an m with $b_1 = p^{2m}$.

Undecidability of arithmetic

$\Rightarrow$: If $a = a_i$, then we can choose b_0, b_1, b_2 as follows:

$$\begin{aligned} b_0 &= 1p^0 + a_0 p^1 + 2p^2 + a_1 p^3 + \cdots + ip^{2i-2} + a_{i-1} p^{2i-1} \\ b_1 &= p^{2i} \\ b_2 &= (i+2) + a_{i+1} p + \cdots + a_r p^{2(r-i)-1} \end{aligned}$$

$\Leftarrow$: Assume that (a)-(d) hold, i.e.,

$$\begin{aligned} t &= b_0 + b_1((i+1) + ap + b_2 p^2) \\ &= b_0 + (i+1)p^{2m} + ap^{2m+1} + p^{2m+2} b_2, \end{aligned}$$

where $b_0 < b_1 = p^{2m}$, $a < p$ and $(i+1) < p$.

Comparing this with

$$t = 1p^0 + a_0 p^1 + 2p^2 + a_1 p^3 + \cdots + (i+1)p^{2i} + a_i p^{2i+1} + \cdots + (r+1)p^{2r} + a_r p^{2r+1}$$

yields $m = i$ and $a = a_i$.

Undecidability of arithmetic

Since p is a prime number, (d) is equivalent to: b_1 is a square and $p|d$ for all $d \geq 2$ with $d|b_1$.

For all $t, p, i \in \mathbb{N}$ we define $\beta(t, p, i)$ as the smallest number a such that $b_0, b_1, b_2 \in \mathbb{N}$ exist with:

(a) $t = b_0 + b_1((i+1) + ap + b_2 p^2)$,

(b) $a < p$,

(c) $b_0 < b_1$,

(d) b_1 is a square and $p|d$ for all $d \geq 2$ with $d|b_1$.

If such numbers $b_0, b_1, b_2 \in \mathbb{N}$ do not exist, then we set $\beta(t, p, i) = 0$.

From the above claim we get: For every sequence $(a_0, \ldots, a_r)$ over $\mathbb{N}$ there are $t, p \in \mathbb{N}$ such that $\beta(t, p, i) = a_i$ for all $0 \leq i \leq r$.

Moreover, it is clear that β is arithmetically definable. $\square$