

Theorems of Church and Trakhtenbrot

Stephen G. Simpson

November 17, 1995

1 Undecidability

Let P be a register-machine program. We shall associate to P a first-order sentence ψ^P describing the deterministic register machine computation or *run* of P starting with all registers empty.

Our sentence ψ^P will be written in a first-order language with a constant symbol 0 , a unary function symbol σ , a binary predicate symbol $<$, a unary function symbol f , and a binary function symbol g . The idea is that $0, \sigma 0, \sigma\sigma 0, \dots$ are intended to form an initial segment of the natural numbers, ordered by $<$, representing the stages of the computation. Moreover $f(x)$ is intended to be the number of the instruction executed at stage x , and $g(x, y)$ is intended to represent the contents of register y just before stage x .

Our sentence ψ^P will be a conjunction of sentences $\psi_0^P, \psi_1^P, \dots, \psi_\ell^P$ where ℓ is the number of instructions in P . Let R_1, \dots, R_k be the registers that are mentioned in P . We write $\bar{0} = 0$ and, for all $n \in \mathbb{N}$, $\overline{n+1} = \sigma\bar{n}$.

The sentence ψ_0^P is defined to be the conjunction of the following clauses:

1. $\forall x \forall y \forall z ((x < y \wedge y < z) \rightarrow x < z)$
2. $\forall x \forall y (x < y \vee x = y \vee y < x)$
3. $\forall x (\neg x < x)$
4. $\forall x (x = \bar{0} \vee \bar{0} < x)$
5. $\forall x (x \neq \sigma x \rightarrow \forall y (y < \sigma x \leftrightarrow (y = x \vee y < x)))$
6. $\forall x (x = \sigma x \rightarrow \forall y (y = x \vee y < x))$

7. $f(\bar{0}) = \bar{1}$
8. $\forall y(g(\bar{0}, y) = \bar{0})$
9. $\forall x(f(x) = \bar{0} \rightarrow \forall z(x < z \rightarrow f(z) = \bar{0}))$
10. $\forall x(f(x) = \bar{0} \rightarrow \forall z \forall y(x < z \rightarrow g(z, y) = g(x, y)))$
11. $\forall x(x = \sigma x \leftrightarrow (f(x) = \bar{0} \wedge \overline{\max(k, \ell)} < x))$.

Let I_1, \dots, I_ℓ be the sequence of numbered instructions which constitutes the program P . For each $1 \leq m \leq \ell$ we shall have a sentence ψ_m^P corresponding to the instruction I_m . There are two types of instructions: *increment* and *decrement*. If I_m is an increment instruction, then it is of the form

increment register R_i and go to instruction I_n

where $1 \leq i \leq k$ and $0 \leq n \leq \ell$; in this case ψ_m^P is defined to be the conjunction of

1. $\forall x(f(x) = \bar{m} \rightarrow g(x, \bar{i}) < \sigma g(x, \bar{i}))$
2. $\forall x(f(x) = \bar{m} \rightarrow g(\sigma x, \bar{i}) = \sigma g(x, \bar{i}))$
3. $\forall x(f(x) = \bar{m} \rightarrow \forall y(y \neq \bar{i} \rightarrow g(\sigma x, y) = g(x, y)))$
4. $\forall x(f(x) = \bar{m} \rightarrow f(\sigma x) = \bar{n})$.

If I_m is a decrement instruction, then it is of the form

if R_i is empty then go to I_{n_0} , otherwise decrement R_i and go to I_{n_1}

where $1 \leq i \leq k$ and $0 \leq n_0 \leq \ell$ and $0 \leq n_1 \leq \ell$; in this case ψ_m^P is defined to be the conjunction of

1. $\forall x((f(x) = \bar{m} \wedge g(x, \bar{i}) = \bar{0}) \rightarrow g(\sigma x, \bar{i}) = \bar{0})$
2. $\forall x((f(x) = \bar{m} \wedge g(x, \bar{i}) = \bar{0}) \rightarrow f(\sigma x) = \bar{n}_0)$
3. $\forall x((f(x) = \bar{m} \wedge g(x, \bar{i}) \neq \bar{0}) \rightarrow \sigma g(\sigma x, \bar{i}) = g(x, \bar{i}))$
4. $\forall x((f(x) = \bar{m} \wedge g(x, \bar{i}) \neq \bar{0}) \rightarrow f(\sigma x) = \bar{n}_1)$
5. $\forall x(f(x) = \bar{m} \rightarrow \forall y(y \neq \bar{i} \rightarrow g(\sigma x, y) = g(x, y)))$.

Finally, let ψ^P be the conjunction $\psi_0^P \wedge \psi_1^P \wedge \cdots \wedge \psi_\ell^P$. We also consider another sentence φ^P , defined as $\psi^P \rightarrow \exists x(f(x) = \bar{0})$.

Consider the run of the register machine program P starting with all registers empty. If P halts, then clearly ψ^P has a finite model

$$\mathcal{A}^P = (A^P, 0^P, \sigma^P, <^P, f^P, g^P)$$

where $A^P = \{0, 1, \dots, n\}$ for a certain $n \in \mathbb{N}$. Moreover \mathcal{A}^P is the unique model of ψ^P , and in \mathcal{A}^P we have $f^P(n) = 0$. Hence in this case φ^P is logically valid.

On the other hand, if P does not halt, then ψ^P has an infinite model

$$\mathcal{A}^P = (\mathbb{N}, 0, \sigma, <, f^P, g^P)$$

which is an initial segment of every model of ψ^P . Hence in this case ψ^P has no finite model. Moreover in \mathcal{A}^P we have $f^P(n) \neq 0$ for all $n \in \mathbb{N}$; hence φ^P is not logically valid.

The unsolvability of the halting problem now implies:

Theorem 1 (Church's Theorem) *The set of logically valid sentences is undecidable.*

Theorem 2 (Trakhtenbrot's Theorem) *The set of sentences which are valid in all finite models is undecidable.*

2 Inseparability

Let V be the set of Gödel numbers of logically valid sentences, and let V_{fin} be the set of Gödel numbers of sentences which are valid in all finite models. Note that $V \subseteq V_{fin}$. The theorems of Church and Trakhtenbrot can be rephrased by saying that neither V nor V_{fin} is recursive.

We shall now prove the following stronger result.

Theorem 3 *There is no recursive set X such that $V \subseteq X \subseteq V_{fin}$.*

Remark. By the Gödel completeness theorem, V is recursively enumerable, *i.e.*, Σ_1^0 . It can also be shown that V_{fin} is co-recursively enumerable, *i.e.*, Π_1^0 (this is straightforward). Thus Theorem 3 implies that V and the complement of V_{fin} form a recursively inseparable pair of recursively enumerable sets.

In general, a pair of recursively enumerable sets A and B is said to be *recursively inseparable* if $A \cap B = \emptyset$ and there is no recursive set X such that $A \subseteq X$ and $X \cap B = \emptyset$. The existence of a recursively inseparable pair of recursively enumerable sets is easily proved by a diagonal argument. For example, we may take $A = \{e \mid \varphi_e^{(1)}(e) \simeq 0\}$ and $B = \{e \mid \varphi_e^{(1)}(e) \simeq 1\}$. If X were a recursive set separating A from B , then letting e be an index of the characteristic function of X we would have $e \in X$ if and only if $e \notin X$, a contradiction.

In order to prove Theorem 3, we shall slightly modify the construction of the Section 1.

Let A and B be a recursively inseparable pair of recursively enumerable sets. Let h be the partial recursive function defined by

$$h(m) \simeq \begin{cases} 0 & \text{if } m \in A, \\ 1 & \text{if } m \in B, \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Let P be a register machine program which computes h . Let ψ^P be a sentence as defined in Section 1, except that clauses 8 and 11 are weakened to

$$8'. \quad \forall y (y \neq \bar{1} \rightarrow g(\bar{0}, y) = \bar{0})$$

and

$$11'. \quad \forall x (x = \sigma x \rightarrow (f(x) = \bar{0} \wedge \overline{\max(k, \ell)} < x))$$

respectively.

For each $m \in \mathbb{N}$, let H_m be the sentence

$$\psi^P \wedge g(\bar{0}, \bar{1}) = \bar{m} \wedge \forall x (x = \sigma x \rightarrow \bar{m} < x),$$

and let θ_m be the sentence

$$H_m \rightarrow \exists x (f(x) = \bar{0} \wedge g(x, \bar{1}) = \bar{0}).$$

Note that if $m \in A$, then $h(m) \simeq 0$, hence θ_m is logically valid, hence the Gödel number of θ_m belongs to V . On the other hand, if $m \in B$, then $h(m) \simeq 1$, hence there is a finite model of

$$H_m \wedge \exists x (f(x) = \bar{0} \wedge g(x, \bar{1}) = \bar{1}),$$

hence θ_m is false in this model, so the Gödel number of θ_m does not belong to V_{fin} .

We can now complete the proof of Theorem 3. If there were a recursive set X such that $V \subseteq X \subseteq V_{fin}$, then

$$\{m \mid \text{the Gödel number of } \theta_m \text{ belongs to } X\}$$

would be a recursive set that separates A from B . Since A and B are recursively inseparable, Theorem 3 follows.