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 $\psi^P$ describing the deterministic register machine computation or run of $P$ starting with all registers empty.

Our sentence $\psi^P$ will be written in a first-order language with a constant symbol 0, a unary function symbol $\sigma$, a binary predicate symbol $<$, a unary function symbol $f$, and a binary function symbol $g$. The idea is that 0, $\sigma 0$, $\sigma \sigma 0$, $\ldots$ 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 $\psi^P$ will be a conjunction of sentences $\psi^P_0$, $\psi^P_1$, $\ldots$, $\psi^P_\ell$ where $\ell$ is the number of instructions in $P$. Let $R_1$, $\ldots$, $R_k$ be the registers that are mentioned in $P$. We write $0 = 0$ and, for all $n \in \mathbb{N}$, $\overline{n+1} = \sigma \overline{n}$.

The sentence $\psi^P_0$ is defined to be the conjunction of the following clauses:

1. $\forall x \forall y \forall z((x < y \land y < z) \rightarrow x < z)$
2. $\forall x \forall y(x < y \lor x = y \lor y < x)$
3. $\forall x(\neg x < x)$
4. $\forall x(x = \overline{0} \lor \overline{0} < x)$
5. $\forall x(x \neq \sigma x \rightarrow \forall y(y < \sigma x \leftrightarrow (y = x \lor y < x)))$
6. $\forall x(x = \sigma x \rightarrow \forall y(y = x \lor y < x))$
7. $f(\emptyset) = \top$

8. $\forall y(g(\emptyset, y) = \emptyset)$

9. $\forall x(f(x) = \emptyset \rightarrow \forall z(x < z \rightarrow f(z) = \emptyset))$

10. $\forall x(f(x) = \emptyset \rightarrow \forall z \forall y(x < z \rightarrow g(z, y) = g(x, y)))$

11. $\forall x(x = \sigma x \leftrightarrow (f(x) = \emptyset \land \max(k, \ell) < x))$.

Let $I_1, \ldots, I_\ell$ be the sequence of numbered instructions which constitutes the program $P$. For each $1 \leq m \leq \ell$ we shall have a sentence $\psi_m^P$ 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 $\psi_m^P$ is defined to be the conjunction of

1. $\forall x(f(x) = m \rightarrow g(x, i) < \sigma g(x, i))$

2. $\forall x(f(x) = m \rightarrow g(\sigma x, i) = \sigma g(x, i))$

3. $\forall x(f(x) = m \rightarrow \forall y(y \neq i \rightarrow g(\sigma x, y) = g(x, y)))$

4. $\forall x(f(x) = m \rightarrow f(\sigma x) = \pi)$.

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 $\psi_m^P$ is defined to be the conjunction of

1. $\forall x((f(x) = m \land g(x, i) = \emptyset) \rightarrow g(\sigma x, i) = \emptyset)$

2. $\forall x((f(x) = m \land g(x, i) = \emptyset) \rightarrow f(\sigma x) = \overline{m})$

3. $\forall x((f(x) = m \land g(x, i) \neq \emptyset) \rightarrow \sigma g(\sigma x, i) = g(x, i))$

4. $\forall x((f(x) = m \land g(x, i) \neq \emptyset) \rightarrow f(\sigma x) = \overline{m_1})$

5. $\forall x(f(x) = m \rightarrow \forall y(y \neq i \rightarrow g(\sigma x, y) = g(x, y)))$. 

2
Finally, let $\psi^P$ be the conjunction $\psi_0^P \land \psi_1^P \land \cdots \land \psi_\ell^P$. We also consider another sentence $\varphi^P$, defined as $\psi^P \rightarrow \exists x(f(x) = \overline{0})$.

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

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

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

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

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

which is an initial segment of every model of $\psi^P$. Hence in this case $\psi^P$ has no finite model. Moreover in $A^P$ we have $f^P(n) \neq 0$ for all $n \in \mathbb{N}$; hence $\varphi^P$ 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 $V_{\text{fin}}$ be the set of Gödel numbers of sentences which are valid in all finite models. Note that $V \subseteq V_{\text{fin}}$. The theorems of Church and Trakhtenbrot can be rephrased by saying that neither $V$ nor $V_{\text{fin}}$ is recursive.

We shall now prove the following stronger result.

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

**Remark.** By the Gödel completeness theorem, $V$ is recursively enumerable, i.e., $\Sigma_1^0$. It can also be shown that $V_{\text{fin}}$ is co-recursively enumerable, i.e., $\Pi_1^0$ (this is straightforward). Thus Theorem 3 implies that $V$ and the complement of $V_{\text{fin}}$ 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 $A = \{ e \mid \varphi^{(1)}(e) \simeq 0 \}$ and $B = \{ e \mid \varphi^{(1)}(e) \simeq 1 \}$. 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

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

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

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

and

11'. $\forall x(x = \sigma x \rightarrow (f(x) = \overline{1} \land \max(k, \ell) < x))$

respectively.

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

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

and let $\theta_m$ be the sentence

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

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

$$H_m \land \exists x(f(x) = \overline{0} \land g(x, \overline{1}) = \overline{1}),$$

4
hence $\theta_m$ is false in this model, so the Gödel number of $\theta_m$ 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.