Theorems of Church and Trakhtenbrot

Stephen G. Simpson

First draft: November 5, 2001
This draft: April 8, 2011

1 Describing a Run of a Program

Let $\mathcal{P}$ be a register machine program. Let $R_1, \ldots, R_s$ be the registers that are used in $\mathcal{P}$. We imagine the registers as boxes which hold a finite number of marbles. Let $I_1, \ldots, I_l$ be the instructions of $\mathcal{P}$. Each of $I_1, \ldots, I_l$ is either an increment instruction (add one marble) or a decrement instruction (branch on empty, remove one marble). We assume that $\mathcal{P}$ starts by executing instruction I_1 at time zero. We assume that time is discrete and linearly ordered. Let I_0 be a stop instruction. We assume that time stops if and when I_0 is reached.

We are going to write a predicate calculus sentence $A_{\mathcal{P}}$ which describes a run of $\mathcal{P}$. To write our sentence, we use a language with the following $s + l + 7$ predicates:

> Ixy: x is identical to y
>
> Tx: x is a time instant
>
> Zx: x is time instant zero
>
> Lxy: x and y are time instants and y is later than x
>
> Nxy: x and y are time instants and y is immediately after x
>
> Mx: x is a marble
>
> R_ixy: marble x is in box R_i at time y, for $1 \leq i \leq s$
>
> I_mx: instruction I_m is executed at time x, for $0 \leq m \leq l$

Let $A_{\mathcal{P}}$ be the conjunction of the following axioms:

1. identity axioms:

   (a) $\forall x \, Ixx$

   (b) $\forall x \, \forall y \, (Ixy \Rightarrow \forall z \, (Ixz \Leftrightarrow Iyz))$

   (c) $\forall x \, \forall y \, (Ixy \Rightarrow \forall z \, (Izx \Leftrightarrow Izy))$

   (d) $\forall x \, \forall y \, (Ixy \Rightarrow (Tx \Leftrightarrow Ty))$

   (e) $\forall x \, \forall y \, (Ixy \Rightarrow (Zx \Leftrightarrow Zy))$

   (f) $\forall x \, \forall y \, (Ixy \Rightarrow \forall z \, (Nxz \Leftrightarrow Nyz))$

   (g) $\forall x \, \forall y \, (Ixy \Rightarrow \forall z \, (Nzx \Leftrightarrow Nzy))$

   (h) $\forall x \, \forall y \, (Ixy \Rightarrow \forall z \, (Lxz \Leftrightarrow Lyz))$

   (i) $\forall x \, \forall y \, (Ixy \Rightarrow \forall z \, (Lzx \Leftrightarrow Lzy))$

   (j) $\forall x \, \forall y \, (Ixy \Rightarrow (Mx \Leftrightarrow My))$

   (k) $\forall x \, \forall y \, (Ixy \Rightarrow \forall z \, (R_i xz \Leftrightarrow R_i yz))$, for $1 \leq i \leq s$

   (l) $\forall x \, \forall y \, (Ixy \Rightarrow \forall z \, (R_i zx \Leftrightarrow R_i zy))$, for $1 \leq i \leq s$

   (m) $\forall x \, \forall y \, (Ixy \Rightarrow (I_m x \Leftrightarrow I_m y))$, for $0 \leq m \leq l$

2. structural axioms:

   (a) $\forall x \, (Mx \vee Tx)$

   (b) $\neg \, \exists x \, (Mx \wedge Tx)$

   (c) $\forall x \, \forall u \, (R_i ux \Rightarrow (Mu \wedge Tx))$, for $1 \leq i \leq s$

   (d) $\neg \, \exists x \, \exists u \, (R_i ux \wedge R_j ux)$, for $1 \leq i < j \leq s$

   (e) $\forall x \, \forall y \, ((Tx \wedge Ty \wedge \neg \, Ixy) \Leftrightarrow (Lxy \vee Lyx))$

   (f) $\forall x \, \forall y \, \forall z \, ((Lxy \wedge Lyz) \Rightarrow Lxz)$

   (g) $\forall x \, \forall y \, (Nxy \Leftrightarrow (Lxy \wedge \neg \, \exists z \, (Lxz \wedge Lzy)))$

   (h) $\forall x \, (Tx \Leftrightarrow (I_0 x \vee I_1 x \vee \cdots \vee I_l x))$

   (i) $\neg \, \exists x \, (I_m x \wedge I_n x)$, for $0 \leq m < n \leq l$

   (j) $\exists x \, Zx$

   (k) $\forall x \, (Zx \Rightarrow (I_1 x \wedge \neg \, \exists y \, Lyx))$

   (l) $\forall x \, (I_m x \Rightarrow \exists y \, Nxy)$, for $1 \leq m \leq l$

   (m) $\forall x \, (I_0 x \Rightarrow \neg \, \exists y \, Lxy)$

3. axioms describing increment instructions:

   For each m in the range $1 \leq m \leq l$, if I_m is an increment instruction then it is of the form increment register R_i and go to instruction I_n

where $1 \leq i \leq s$ and $0 \leq n \leq l$, and we have:

(a) $\forall x \, \forall y \, ((I_m x \wedge Nxy) \Rightarrow I_n y)$

(b) $\forall x \, \forall y \, ((I_m x \wedge Nxy) \Rightarrow \forall u \, (R_i ux \Rightarrow R_i uy))$

(c) $\forall x \, \forall y \, ((I_m x \wedge Nxy) \Rightarrow \exists u \, (R_i uy \wedge \neg R_i ux))$

(d) $\forall x \, \forall y \, ((I_m x \wedge Nxy) \Rightarrow \forall u \, \forall v \, ((R_i uy \wedge \neg R_i ux \wedge R_i vy \wedge \neg R_i vx) \Rightarrow Iuv))$

(e) $\forall x \, \forall y \, ((I_m x \wedge Nxy) \Rightarrow \forall u \, (R_j ux \Leftrightarrow R_j uy))$, for $1 \leq j \leq s$, $j \neq i$

4. axioms describing decrement instructions:

   For each m in the range $1 \leq m \leq l$, if I_m is a decrement instruction then it is of the form

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

   where $1 \leq i \leq s$ and $0 \leq n_0 \leq l$ and $0 \leq n_1 \leq l$, and we have:

   (a) $\forall x \, \forall y \, ((I_m x \wedge Nxy \wedge \neg \exists u \, R_i ux) \Rightarrow I_{n_0} y)$

   (b) $\forall x \, \forall y \, ((I_m x \wedge Nxy \wedge \exists u \, R_i ux) \Rightarrow I_{n_1} y)$

   (c) $\forall x \, \forall y \, ((I_m x \wedge Nxy) \Rightarrow \forall u \, (R_i uy \Rightarrow R_i ux))$

   (d) $\forall x \, \forall y \, ((I_m x \wedge Nxy \wedge \exists u \, R_i ux) \Rightarrow \exists u \, (R_i ux \wedge \neg R_i uy))$

   (e) $\forall x \, \forall y \, ((I_m x \wedge Nxy) \Rightarrow \forall u \, \forall v \, ((R_i ux \wedge \neg R_i uy \wedge R_i vx \wedge \neg R_i vy) \Rightarrow Iuv))$

   (f) $\forall x \, \forall y \, ((I_m x \wedge Nxy) \Rightarrow \forall u \, (R_j ux \Leftrightarrow R_j uy))$, for $1 \leq j \leq s$, $j \neq i$

Note that $A_{\mathcal{P}}$ describes a run of the program $\mathcal{P}$, but we have not specified the initial contents of the registers.

2 Unsolvability

The *Halting Problem* is the problem of deciding whether a given register machine program started with all registers empty eventually stops. It is well known that the Halting Problem is unsolvable.

The *Validity Problem* is the problem of deciding whether a given predicate calculus sentence is valid. The *Satisfiability Problem* is the problem of deciding whether a given predicate calculus sentence is satisfiable. We are going to show that the Validity Problem and the Satisfiability Problem are unsolvable. This will be accomplished by reducing the Halting Problem to them. In other words, we shall show that if either of them were solvable, then the Halting Problem would be solvable.

Given a register machine program $\mathcal{P}$, let $\mathcal{P}(0)$ be the unique run of $\mathcal{P}$ starting with all registers empty. Let $A_{\mathcal{P}}$ be as in Section 1, and let $A_{\mathcal{P}}(0)$ be $A_{\mathcal{P}}$ conjuncted with

$$\forall x\,(Zx \Rightarrow \neg\,\exists u\,(R_1ux \vee \cdots \vee R_sux))\ .$$

Thus $A_{\mathcal{P}}(0)$ describes $\mathcal{P}(0)$.

Let B be the sentence $\exists x\, I_0x$. If $\mathcal{P}(0)$ eventually stops, then clearly $A_{\mathcal{P}}(0)$ is satisfiable in a finite domain, and any domain satisfying $A_{\mathcal{P}}(0)$ is necessarily finite and satisfies B. Hence in this case $A_{\mathcal{P}}(0) \Rightarrow B$ is valid.

On the other hand, if $\mathcal{P}(0)$ does not eventually stop, then $A_{\mathcal{P}}(0)$ is not satisfiable in any finite domain, but $A_{\mathcal{P}}(0) \wedge \neg\, B$ is satisfiable in an infinite domain. Thus in this case $A_{\mathcal{P}}(0) \Rightarrow B$ is not valid.

We have proved:

Theorem 1 (Church's Theorem) *The Satisfiability Problem and the Validity Problem are unsolvable.*

Theorem 2 (Trakhtenbrot's Theorem) *The problem of satisfiability in a finite domain is unsolvable. The problem of validity in finite domains is unsolvable.*

3 Recursive Inseparability

Let V be the set of Gödel numbers of valid sentences, and let V_{fin} be the set of Gödel numbers of sentences which are valid in all finite domains. Note that $V \subseteq V_{\text{fin}}$. Theorems 1 and 2 can be rephrased by saying that neither V nor V_{fin} is recursive.

We shall now prove the following stronger result, also due to Trakhtenbrot.

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.*, Σ^0_1. It can also be shown that V_{fin} is co-recursively enumerable, *i.e.*, Π^0_1 (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 I and J is said to be *recursively inseparable* if $I \cap J = \emptyset$ and there is no recursive set X such that $I \subseteq X$ and $X \cap J = \emptyset$. The existence of a recursively inseparable pair of recursively enumerable sets is easily proved by a diagonal argument. For example, we may take $I = \{e \mid \varphi_e^{(1)}(e) \simeq 0\}$ and $J = \{e \mid \varphi_e^{(1)}(e) \simeq 1\}$. If X were a recursive set separating I from J, 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 Section 2.

Let I and J be a recursively inseparable pair of recursively enumerable sets. Let ψ be the partial recursive function defined by

$$\psi(n) \simeq \begin{cases} 0 & \text{if } n \in I\ , \\ 1 & \text{if } n \in J\ , \\ \text{undefined otherwise}\ . \end{cases}$$

Let $\mathcal{P}$ be a register machine program which computes ψ. Let $\mathcal{P}(n)$ be the unique run of $\mathcal{P}$ starting with n in R_1 and all other registers empty. Let $A_\mathcal{P}$ be as before, and let $A_\mathcal{P}(n)$ be $A_\mathcal{P}$ conjuncted with

$$\forall x\,(Zx \Rightarrow \exists \text{ exactly } n\ u \text{ such that } R_1ux)$$

conjuncted with

$$\forall x\,(Zx \Rightarrow \neg\,\exists u\,(R_2ux \vee \cdots \vee R_sux)).$$

Thus $A_\mathcal{P}(n)$ describes $\mathcal{P}(n)$.

Let B_0 be the sentence $\exists x\,(I_0x \wedge \neg\,\exists u\,R_2ux)$. If $n \in I$ then $\mathcal{P}(n)$ eventually stops with 0 in R_2, hence $A_\mathcal{P}(n) \Rightarrow B_0$ is valid, hence the Gödel number of $A_\mathcal{P}(n) \Rightarrow B_0$ belongs to V. If $n \in J$, then $\mathcal{P}(n)$ eventually stops with 1 in R_2, hence $A_\mathcal{P}(n) \wedge \neg\,B_0$ is satisfiable in a finite domain, hence the Gödel number of $A_\mathcal{P}(n) \Rightarrow B_0$ does not belong to V_{fin}.

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

$$\{n \mid \text{the Gödel number of } A_\mathcal{P}(n) \Rightarrow B_0 \text{ belongs to } X\}$$

would be a recursive set which separates I from J. Since I and J are recursively inseparable, Theorem 3 follows.