CS 5000: Lecture 37

Vladimir Kulyukin
Department of Computer Science
Utah State University

Outline

- A Computational Theory of Simulation

Review: Universality Theorem

Theorem 3.1 (Universality Theorem): For each $n > 0$, the function $\Phi^{(n)}(x_1, \ldots, x_n, y)$ is partially computable.

Review: Snapshots

$s = (i, \sigma), 1 \leq i \leq n+1.$

i is the number of the instruction to be executed.

σ is a state of the program.

s is terminal if $i = n+1$.

Step-Counter Predicate

$STP^{(n)}(x_1,\ldots,x_n,y,t)$

1. Program y halts on inputs $x_1,\ldots,x_n$ in t or fewer steps.

2. There is a computation of program y on inputs $x_1,\ldots,x_n$ such that the length of that computation is $\leq t+1$.

Theorem 3.2 (Ch. 4): Step-Counter Theorem

For $n > 0$, the predicate $STP^{(n)}(x_1,...,x_n, y, t)$ is primitive recursive.

Proof 3.2

$$(i, \sigma) \text{ is encoded as } \langle i, z \rangle.$$

$$Z = [0, x_1, 0, x_2, \ldots].$$

Proof 3.2: Extracting Instruction Components

Let y be a program number. Then $y+1$ is the number of its source code. So $(y+1)_i = \langle a, \langle b, c \rangle \rangle$. Then we have:

$$\mathrm{LABEL}(i, y) = l((y+1)_i).$$
$$\mathrm{VAR}(i, y) = r(r((y+1)_i)) + 1.$$
$$\mathrm{INSTR}(i, y) = l(r((y+1)_i)).$$
$$\mathrm{LABEL'}(i, y) = l(r((y+1)_i)) \dot{-} 2.$$

Proof 3.2: Skipping Instruction

Let $x = \langle i, z \rangle$ be a snapshot.

When should we skip the next instruction?

1. When it is of the type $V \leftarrow V$ and its number is $\leq$ the length of the program.
2. When it is of the type IF $V \neq 0$ GOTO L and $V = 0$.

Proof 3.2: Skipping Instruction

Let $x = \langle i, z \rangle$ be a snapshot.

When should we skip the next instruction?

1. When it is of the type $V \leftarrow V$ and its number is $\leq$ the length of the program.

$$\text{INSTR}(l(x), y) = 0 \;\&\; l(x) \leq Lt(y+1).$$

Proof 3.2: Skipping Instruction

Let $x = \langle i, z \rangle$ be a snapshot.

When should we skip the next instruction?

2. When it is of the type IF $V \neq 0$ GOTO L
   and $V = 0$ or it is of the type $V \leftarrow V - 1$
   and $V = 0$. Formally, we have:

$$\text{INSTR}(l(x), y) \geq 2 \ \& \ \neg\left(p_{VAR(l(x),y)} \mid r(x)\right)$$

Proof 3.2: Skipping Instruction

Let $x = \langle i, z \rangle$ be a snapshot.

When should we skip the next instruction?

$$\text{SKIP}(x, y) \Leftrightarrow$$
$$[\text{INSTR}(l(x), y) = 0 \,\&\, l(x) \leq Lt(y+1)] \vee$$
$$[\text{INSTR}(l(x), y) \geq 2 \,\&\, \neg(p_{VAR(l(x),y)} \mid r(x))].$$

Proof 3.2: Increment, Decrement, Branch

Let $x = \langle i, z \rangle$ be a snapshot.

$$\mathrm{INCR}(x, y) \Leftrightarrow \mathrm{INSTR}(l(x), y) = 1.$$

$$\mathrm{DECR}(x, y) \Leftrightarrow \mathrm{INSTR}(l(x), y) = 2 \,\&$$

$$p_{VAR(l(x),y)} \mid r(x).$$

$$\mathrm{BRANCH}(x, y) \Leftrightarrow$$

$$\mathrm{INSTR}(l(x), y) > 2 \,\&$$

$$p_{VAR(l(x),y)} \mid r(x) \,\&$$

$$(\exists i)_{\leq Lt(y+1)} \mathrm{LABEL}(i, y) = \mathrm{LABEL}'(l(x), y).$$

Proof 3.2: Successor State

$$\mathrm{SUCC}(x,y)=\begin{cases}\langle l(x)+1, r(x)\rangle, \text{ if SKIP}(x,y)\\ \langle l(x)+1, r(x)\cdot p_{VAR(l(x),y)}\rangle, \text{ if INCR}(x,y)\\ \langle l(x)+1, \lfloor r(x)/p_{VAR(l(x),y)}\rfloor\rangle, \text{ if DECR}(x,y)\\ \left\langle \min_{i\le Lt(y+1)}[\text{LABEL}(i,y)=\text{LABEL}'(l(x)y)], r(x)\right\rangle,\\ \text{if BRANCH}(x,y)\\ \langle Lt(y+1)+1, r(x)\rangle, \text{ otherwise.}\end{cases}$$

Proof 3.2: Initial Snapshot

$$\mathrm{INIT}^{(n)}\left(x_1,\ldots,x_n\right)=\left\langle 1,\prod_{i=1}^{n}\left(p_{2i}\right)^{X_i}\right\rangle$$

Proof 3.2: Terminal Snapshot

$$\mathrm{TERM}(x, y) = l(x) > Lt(y+1)$$

Proof 3.2: Snapshots

$$\mathrm{SNAP}^{(n)}(x_1,\ldots,x_n,y,0) = \mathrm{INIT}^{(n)}(x_1,\ldots,x_n)$$

$$\mathrm{SNAP}^{(n)}(x_1,\ldots,x_n,y,i+1) =$$

$$\mathrm{SUCC}\left(\mathrm{SNAP}^{(n)}(x_1,\ldots,x_n,y,i),y\right)$$

Proof 3.2: Primitive Recursiveness

$$\mathrm{STP}^{(n)}(x_1,\ldots,x_n,y,t) \Leftrightarrow$$
$$\mathrm{TERM}\left(\mathrm{SNAP}^{(n)}(x_1,\ldots,x_n,y,t),y\right)$$

Recommended Reading

- Section 4.3.