Semantyka i weryfikacja programów

Bartosz Klin

(slajdy Andrzeja Tarleckiego)

Instytut Informatyki

Wydział Matematyki, Informatyki i Mechaniki

Uniwersytet Warszawski

http://www.mimuw.edu.pl/~klin pok. 5680

klin@mimuw.edu.pl

Strona tego wykładu:

http://www.mimuw.edu.pl/~klin/sem19-20.html

Program Semantics & Verification

Bartosz Klin

(slides courtesy of Andrzej Tarlecki)

Institute of Informatics
Faculty of Mathematics, Informatics and Mechanics
University of Warsaw

`http://www.mimuw.edu.pl/~klin` office: 5680

`klin@mimuw.edu.pl`

This course: `http://www.mimuw.edu.pl/~klin/sem19-20.html`

Denotational semantics for TINY

Semantic clauses

$\mathcal{S} \colon \mathbf{Stmt} \to \mathbf{STMT}$

$$
\begin{aligned}
\mathcal{S}[\![x := e]\!]s &= s[x \mapsto \mathcal{E}[\![e]\!]\ s] \\
\mathcal{S}[\![\mathbf{skip}]\!]s &= s \\
\mathcal{S}[\![S_1 ; S_2]\!]s &= \mathcal{S}[\![S_2]\!](\mathcal{S}[\![S_1]\!]s) \\
\mathcal{S}[\![\mathbf{if}\ b\ \mathbf{then}\ S_1\ \mathbf{else}\ S_2]\!]s &= \mathit{ifte}_{\mathbf{State}}(\mathcal{B}[\![b]\!]s, \mathcal{S}[\![S_1]\!]s, \mathcal{S}[\![S_2]\!]s) \\
\mathcal{S}[\![\mathbf{while}\ b\ \mathbf{do}\ S]\!]s &= \mathit{ifte}_{\mathbf{State}}(\mathcal{B}[\![b]\!]s, \mathcal{S}[\![\mathbf{while}\ b\ \mathbf{do}\ S]\!]s', s) \\
&\qquad \text{where } s' = \mathcal{S}[\![S]\!]s
\end{aligned}
$$

Denotational semantics for Tiny

The same clauses with notational sugar

$$\mathcal{S} : \mathbf{Stmt} \to \mathbf{STMT}$$

$$
\begin{aligned}
\mathcal{S}[\![x := e]\!] &= \lambda s{:}\mathbf{State}.s[x \mapsto \mathcal{E}[\![e]\!]\ s] \\
\mathcal{S}[\![\mathbf{skip}]\!] &= id_{\mathbf{State}} \\
\mathcal{S}[\![S_1; S_2]\!] &= \mathcal{S}[\![S_1]\!];\mathcal{S}[\![S_2]\!] \\
\mathcal{S}[\![\mathbf{if}\ b\ \mathbf{then}\ S_1\ \mathbf{else}\ S_2]\!] &= cond(\mathcal{B}[\![b]\!], \mathcal{S}[\![S_1]\!], \mathcal{S}[\![S_2]\!]) \\
\mathcal{S}[\![\mathbf{while}\ b\ \mathbf{do}\ S]\!] &= cond(\mathcal{B}[\![b]\!], \mathcal{S}[\![S]\!];\mathcal{S}[\![\mathbf{while}\ b\ \mathbf{do}\ S]\!], id_{\mathbf{State}})
\end{aligned}
$$

Something wrong?

The clause for **while**:

$$\boxed{\mathcal{S}[\![\textbf{while } b \textbf{ do } S]\!] = cond(\mathcal{B}[\![b]\!], \mathcal{S}[\![S]\!];\mathcal{S}[\![\textbf{while } b \textbf{ do } S]\!], id_{\textbf{State}})}$$

is *not* compositional!

We "define":

$$??? \quad \mathcal{S}[\![\textbf{while } b \textbf{ do } S]\!] = \Phi(\ldots, \mathcal{S}[\![\textbf{while } b \textbf{ do } S]\!], \ldots) \quad ???$$

We need *fixed point definitions*

Potential problems with fixed point definitions

Consider fixed point definitions in $\mathbf{STMT} = \mathbf{State} \rightharpoonup \mathbf{State}$, as

$$\mathcal{S}[\![\mathbf{while}\ b\ \mathbf{do}\ S]\!] = \Phi(\ldots, \mathcal{S}[\![\mathbf{while}\ b\ \mathbf{do}\ S]\!], \ldots)$$

- Does a fixed point always exist?

$$f = \lambda s{:}\mathbf{State}.\mathit{ifte}_{\mathbf{State}}(f(s) \text{ is not defined}, s, f(s)[x \mapsto (f(s)\ x) + 1])$$

Only some functionals Φ may be allowed

- If a fixed point exists, is it unique?

$$f = \lambda s{:}\mathbf{State}.f(s)[x \mapsto 2 * (f(s)\ x)]$$

(or even: $f = \lambda s{:}\mathbf{State}.f(s)$)

Some "best" fixed point must be chosen

The guiding fixed point definition

Looking closer at the clause for **while**:

$$\boxed{\mathcal{S}[\![\mathbf{while}\ b\ \mathbf{do}\ S]\!] = \Phi(\mathcal{S}[\![\mathbf{while}\ b\ \mathbf{do}\ S]\!])}$$

where $\Phi \colon \mathbf{STMT} \to \mathbf{STMT}$ is defined as follows:

$$\Phi(F) = cond(\mathcal{B}[\![b]\!], \mathcal{S}[\![S]\!];F, id_{\mathbf{State}})$$

Whatever fixed point we choose, we want it to be adequate for our operational intuitions; we want a denotation $\mathit{fix}(\Phi) \in \mathbf{STMT}$ that is a fixed point of Φ (so that $\Phi(\mathit{fix}(\Phi)) = \mathit{fix}(\Phi)$) and is adequate for the operational semantics of **while**, i.e., such that

$$\boxed{\langle \mathbf{while}\ b\ \mathbf{do}\ S, s\rangle \Rightarrow^{*} s' \ \text{ iff } \ \mathit{fix}(\Phi)\ s = s'}$$

Right guess!

Suppose that we have such adequacy for S, i.e., $\langle S, s\rangle \Rightarrow^* s'$ iff $\mathcal{S}[\![S]\!]\, s = s'$.

Right guess:

$$\boxed{\langle \textbf{while } b \textbf{ do } S, s\rangle \Rightarrow^* s' \quad \text{iff} \quad \text{for some } n \geq 0, \Phi^n(\emptyset_{\mathbf{State} \rightharpoonup \mathbf{State}})\, s = s'}$$

where $\emptyset_{\mathbf{State} \rightharpoonup \mathbf{State}} : \mathbf{State} \rightharpoonup \mathbf{State}$ is the function undefined everywhere, $\Phi^0(\emptyset_{\mathbf{State} \rightharpoonup \mathbf{State}}) = \emptyset_{\mathbf{State} \rightharpoonup \mathbf{State}}$, and $\Phi^{n+1}(\emptyset_{\mathbf{State} \rightharpoonup \mathbf{State}}) = \Phi(\Phi^n(\emptyset_{\mathbf{State} \rightharpoonup \mathbf{State}}))$.

Proof: in a moment.

Conclusion

$$\boxed{\mathcal{S}[\![\textbf{while } b \textbf{ do } S]\!] = \mathit{fix}(\Phi) = \bigcup_{n \geq 0} \Phi^n(\emptyset_{\mathbf{State} \rightharpoonup \mathbf{State}})}$$

This is well-defined, and yields the *least* fix-point of Φ, see below.

$$\textbf{while } sqr \leq n \textbf{ do } rt := rt + 1;\ sqr := sqr + 2 * rt + 1$$

$$\Phi(F) = cond(\mathcal{B}[\![sqr \leq n]\!], \mathcal{S}[\![rt := rt + 1;\ sqr := sqr + 2 * rt + 1]\!]; F, id_{\mathbf{State}})$$

$s(n, rt, sqr)$	$\Phi^0(\emptyset)(s)$	$\Phi^1(\emptyset)(s)$	$\Phi^2(\emptyset)(s)$	$\Phi^3(\emptyset)(s)$	$\Phi^4(\emptyset)(s)$	$\cdots$	$\bigcup \Phi^n(\emptyset)(s)$
$0, 0, 1$	?	$0, 0, 1$	$0, 0, 1$	$0, 0, 1$	$0, 0, 1$	$\cdots$	$0, 0, 1$
$1, 0, 1$	?	?	$1, 1, 4$	$1, 1, 4$	$1, 1, 4$	$\cdots$	$1, 1, 4$
$2, 0, 1$	?	?	$2, 1, 4$	$2, 1, 4$	$2, 1, 4$	$\cdots$	$2, 1, 4$
$3, 0, 1$	?	?	$3, 1, 4$	$3, 1, 4$	$3, 1, 4$	$\cdots$	$3, 1, 4$
$4, 0, 1$	?	?	?	$4, 2, 9$	$4, 2, 9$	$\cdots$	$4, 2, 9$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\cdots$	$\ldots$
$8, 0, 1$	?	?	?	$8, 2, 9$	$8, 2, 9$	$\cdots$	$8, 2, 9$
$9, 0, 1$	?	?	?	?	$9, 3, 16$	$\cdots$	$9, 3, 16$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\cdots$	$\ldots$

$$\Phi(F) = cond(\mathcal{B}[\![sqr \leq n]\!], \mathcal{S}[\![rt := rt + 1; sqr := sqr + 2 * rt + 1]\!]; F, id_{\mathbf{State}})$$

$s(n, rt, sqr)$	$\Phi^0(\emptyset)(s)$	$\Phi^1(\emptyset)(s)$	$\Phi^2(\emptyset)(s)$	$\Phi^3(\emptyset)(s)$	$\Phi^4(\emptyset)(s)$	$\cdots$	$\bigcup \Phi^n(\emptyset)(s)$
$0, 0, 1$	?	$0, 0, 1$	$0, 0, 1$	$0, 0, 1$	$0, 0, 1$	$\cdots$	$0, 0, 1$
$1, 0, 1$	?	?	$1, 1, 4$	$1, 1, 4$	$1, 1, 4$	$\cdots$	$1, 1, 4$
$1, 1, 4$	?	$1, 1, 4$	$1, 1, 4$	$1, 1, 4$	$1, 1, 4$	$\cdots$	$1, 1, 4$
$2, 0, 1$	?	?	$2, 1, 4$	$2, 1, 4$	$2, 1, 4$	$\cdots$	$2, 1, 4$
$2, 1, 4$	?	$2, 1, 4$	$2, 1, 4$	$2, 1, 4$	$2, 1, 4$	$\cdots$	$2, 1, 4$
$3, 0, 1$	?	?	$3, 1, 4$	$3, 1, 4$	$3, 1, 4$	$\cdots$	$3, 1, 4$
$3, 1, 4$	?	$3, 1, 4$	$3, 1, 4$	$3, 1, 4$	$3, 1, 4$	$\cdots$	$3, 1, 4$
$4, 0, 1$	?	?	?	$4, 2, 9$	$4, 2, 9$	$\cdots$	$4, 2, 9$
$4, 1, 4$	?	?	$4, 2, 9$	$4, 2, 9$	$4, 2, 9$	$\cdots$	$4, 2, 9$
$4, 2, 9$	?	$4, 2, 9$	$4, 2, 9$	$4, 2, 9$	$4, 2, 9$	$\cdots$	$4, 2, 9$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$9, 0, 1$	?	?	?	?	$9, 3, 16$	$\cdots$	$9, 3, 16$
$9, 1, 4$	?	?	?	$9, 3, 16$	$9, 3, 16$	$\cdots$	$9, 3, 16$
$9, 2, 9$	?	?	$9, 3, 16$	$9, 3, 16$	$9, 3, 16$	$\cdots$	$9, 3, 16$
$9, 3, 16$	?	$9, 3, 16$	$9, 3, 16$	$9, 3, 16$	$9, 3, 16$	$\cdots$	$9, 3, 16$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

Proof

"$\Longrightarrow$": By induction on the length of the computation $\langle \mathbf{while}\ b\ \mathbf{do}\ S, s\rangle \Rightarrow^k s'$.

$k > 0$: Then $\langle \mathbf{while}\ b\ \mathbf{do}\ S, s\rangle \Rightarrow \gamma \Rightarrow^{k-1} s'$. By cases on this first step:

- $\mathcal{B}[\![b]\!]\ s = \mathbf{ff}$ and $\gamma = s$. Then $s' = s$, and $\Phi(\emptyset_{\mathbf{State} \rightharpoonup \mathbf{State}})\ s = s$. OK
- $\mathcal{B}[\![b]\!]\ s = \mathbf{tt}$ and $\gamma = \langle S; \mathbf{while}\ b\ \mathbf{do}\ S, s\rangle \Rightarrow^{k-1} s'$. Then $\langle S, s\rangle \Rightarrow^{k_1} \hat{s}$ and $\langle \mathbf{while}\ b\ \mathbf{do}\ S, \hat{s}\rangle \Rightarrow^{k_2} s'$, for some $\hat{s} \in \mathbf{State}$ and $k_1, k_2 > 0$ with $k_1 + k_2 = k - 1$. Hence, $\mathcal{S}[\![S]\!]\ s = \hat{s}$ and $\Phi^n(\emptyset_{\mathbf{State} \rightharpoonup \mathbf{State}})\ \hat{s} = s'$ for some $n \geq 0$. Thus, $\Phi^{n+1}(\emptyset_{\mathbf{State} \rightharpoonup \mathbf{State}})\ s = s'$. OK

BTW: This relies only on $\boxed{\langle S, s\rangle \Rightarrow^* s' \Longrightarrow \mathcal{S}[\![S]\!]\ s = s'}$

Proof

"$\Longleftarrow$": By induction on $n \geq 0$, assuming $\Phi^n(\emptyset_{\mathbf{State} \rightharpoonup \mathbf{State}})\ s = s'$.

$n > 0$: Then

$\Phi^n(\emptyset_{\mathbf{State} \rightharpoonup \mathbf{State}})\ s = cond(\mathcal{B}[\![b]\!], \mathcal{S}[\![S]\!];\Phi^{n-1}(\emptyset_{\mathbf{State} \rightharpoonup \mathbf{State}}), id_{\mathbf{State}})\ s$.

- $\mathcal{B}[\![b]\!]\ s = \mathbf{ff}$: then $\Phi^n(\emptyset_{\mathbf{State} \rightharpoonup \mathbf{State}})\ s = s$, so $s' = s$, and also $\langle \mathbf{while}\ b\ \mathbf{do}\ S, s\rangle \Rightarrow s$. OK
- $\mathcal{B}[\![b]\!]\ s = \mathbf{tt}$: then $\Phi^n(\emptyset_{\mathbf{State} \rightharpoonup \mathbf{State}})\ s = \Phi^{n-1}(\emptyset_{\mathbf{State} \rightharpoonup \mathbf{State}})\ (\mathcal{S}[\![S]\!]\ s)$. Hence, $\langle \mathbf{while}\ b\ \mathbf{do}\ S, \mathcal{S}[\![S]\!]\ s\rangle \Rightarrow^* s'$, and since $\langle S, s\rangle \Rightarrow^* (\mathcal{S}[\![S]\!]\ s)$, we get $\langle \mathbf{while}\ b\ \mathbf{do}\ S, s\rangle \Rightarrow \langle S; \mathbf{while}\ b\ \mathbf{do}\ S, s\rangle \Rightarrow^* \langle \mathbf{while}\ b\ \mathbf{do}\ S, \mathcal{S}[\![S]\!]\ s\rangle \Rightarrow^* s'$. OK

BTW: This relies only on $\boxed{\langle S, s\rangle \Rightarrow^* s' \Longleftarrow \mathcal{S}[\![S]\!]\ s = s'}$

Adequacy of denotational semantics

Fact: *For each statement* $S \in \mathbf{Stmt}$ *and states* $s, s' \in \mathbf{State}$,

$$\boxed{\mathcal{S}[\![S]\!]\, s = s' \quad \text{iff} \quad \langle S, s\rangle \Rightarrow^* s'}$$

Proof:

"$\Longrightarrow$": By structural induction on S.

"$\Longleftarrow$": By induction on the length of the computation $\langle S, s\rangle \Rightarrow^* s'$.

How general is this fixpoint construction?

For any sets X and Y, define the *information ordering* on partial functions from X to Y:

$$f \sqsubseteq g \ \text{ iff } \ [f(x) \text{ is defined implies } g(x) \text{ is defined and } f(x) = g(x)]$$

This is a *complete partial order* (c.p.o.) on $X \rightharpoonup Y$, i.e., it has the least element $\emptyset_{X \rightharpoonup Y}$ and limits (least upper bounds) of increasing chains:

$$f_0 \sqsubseteq f_1 \sqsubseteq f_2 \sqsubseteq f_3 \sqsubseteq \cdots \sqsubseteq \bigcup_{n \in \mathbb{N}} f_n$$

Putting $X = Y = \mathbf{State}$, we get a c.p.o. on the set $\mathbf{STMT}$.

Continuous functions

Fact: *For any* $S \in \mathbf{Stmt}$ *and* $b \in \mathbf{BExp}$, *the function*

$$\Phi(F) = cond(\mathcal{B}[\![b]\!], \mathcal{S}[\![S]\!]; F, id_{\mathbf{State}})$$

is monotone and even continuous:

- $\Phi(f) \sqsubseteq \Phi(g)$ *if* $f \sqsubseteq g$,
- $\Phi\left(\bigcup_{n\in\mathbb{N}} f_n\right) = \bigcup_{n\in\mathbb{N}} \Phi(f_n)$ *if* $f_1 \sqsubseteq f_2 \sqsubseteq f_3 \sqsubseteq \cdots$.

Fact: **(Kleene fixpoint theorem)** *Every continuous function* Φ *on a c.p.o.* X *has a least fixpoint, defined by:*

$$\bigcup_{n\in\mathbb{N}} \Phi^n(\emptyset).$$

Continuous functions

Some functions are not continuous, or even not monotone. Recall our flawed "fixpoint definition":

$$f = \lambda s{:}\mathbf{State}.\mathit{ifte}_{\mathbf{State}}(f(s) \text{ is not defined}, s, f(s)[x \mapsto (f(s)\,x) + 1])$$

This is supposed to be a fixpoint of the function $\Phi : \mathbf{STMT} \to \mathbf{STMT}$:

$$\Phi(F) = \lambda s{:}\mathbf{State}.\mathit{ifte}_{\mathbf{State}}(F(s) \text{ is not defined}, s, F(s)[x \mapsto (F(s)\,x) + 1])$$

This Φ is well-defined, but it is not monotone: $\emptyset_{\mathbf{State} \rightharpoonup \mathbf{State}} \sqsubseteq id_{\mathbf{State}}$, but

$$\Phi(\emptyset_{\mathbf{State} \rightharpoonup \mathbf{State}}) = id_{\mathbf{State}} \not\sqsubseteq \Phi(id_{\mathbf{State}}).$$

In practice, all "reasonable" functions that we are likely to write are continuous.