A Hoare Logic for the Coinductive Trace-Based Big-Step Semantics of While

Keiko Nakata
Institute of Cybernetics, Tallinn University of Technology
Joint work with Tarmo Uustalu

November 2009

Motivation

There are important programs that are not supposed to terminate, e.g. operating systems and data base systems.

Our motivation is to set up a foundational framework that accounts for both terminating and non-terminating program runs.

Applications include

- certified compilers, program transformations
- information flow analysis

Results, so far

We study the While language.

We have devised:

- trace-based big-step relational semantics, as well as small-step relational and big-step & small-step functional. They are all defined coinductively and equivalent constructively.

  (Appeared in TPHOLs 2009)

- Hoare logic, proved sound and complete.

All results are formalized in a fully constructive setting of Coq.

My talk

Today I will present

- the big-step relational semantics
- the Hoare logic

The While language

$$
\begin{array}{rcl}
x, y, z & \in & \textit{Variables} \\
e & \in & \textit{Expressions} \\
v & \in & \textit{Integers} \\
\sigma & \in & \textit{Variables} \rightarrow \textit{Integers}
\end{array}
$$

$$
\begin{array}{lrcl}
\textit{statement} & s & ::= & \text{skip} \mid s_0; s_1 \mid x := e \\
 & & \mid & \text{if } e \text{ then } s_t \text{ else } s_f \mid \text{while } e \text{ do } s_t
\end{array}
$$

Notations

The While language

$\sigma[x \mapsto v]$ denotes the update of σ with v at x.

$[\![e]\!]\sigma$ evaluates e in the state σ.
E.g. $[\![x + y]\!]\{x \mapsto 2, x \mapsto 2\} = 4$

$\sigma \models e$ denotes that e evaluates to truth (non-zero) in σ.
E.g $\{x \mapsto 2, x \mapsto 2\} \models x + y$

$\sigma \not\models e$ denotes that e evaluates to falsity (zero) in σ.
E.g $\{x \mapsto 2, x \mapsto 2\} \not\models x - y$

Traces

Traces $\tau \in \mathit{trace}$ are possibly infinite non-empty sequences of states, defined coinductively by:

$$\frac{}{\langle \sigma \rangle \in \mathit{trace}} \qquad \frac{\tau \in \mathit{trace}}{\sigma :: \tau \in \mathit{trace}}$$

We define bisimilarity (equivalence relation) between traces, $\tau \approx \tau'$, coinductively by:

$$\frac{}{\langle \sigma \rangle \approx \langle \sigma \rangle} \qquad \frac{\tau \approx \tau'}{\sigma :: \tau \approx \sigma :: \tau'}$$

We think of bisimilar traces as equal, i.e. traces as a setoid with bisimilarity as the equivalence relation.

Finiteness and infiniteness

Traces

We define convergence of τ at σ, $\tau \downarrow \sigma$, inductively:

$$\frac{}{\langle \sigma \rangle \downarrow \sigma} \qquad \frac{\tau \downarrow \sigma}{\sigma' :: \tau \downarrow \sigma}$$

Finiteness of τ, *finite* τ, is defined as

$$\textit{finite}\ \tau \quad \text{if} \quad \exists \sigma . \tau \downarrow \sigma$$

We define infiniteness of τ, *infinite* τ, coinductively:

$$\frac{\textit{infinite}\ \tau}{\textit{infinite}\ \sigma :: \tau}$$

Finiteness and infiniteness (2)

Traces

Working in a constructive logic, our trace predicates have a rich structure.

- $\neg$ *finite* $\models$ *infinite*
- $\neg$ *infinite* $\models$ *finite* is not probable constructively.
  (But is provably classically.)

where $P \models Q$ abbreviates $\forall \tau. P\ \tau \Rightarrow Q\ \tau$.

In particular we do not have

$$\forall \tau. \mathit{finite}\ \tau \vee \mathit{infinite}\ \tau$$

(I.e. finiteness is undecidable.)

Big-step semantics

The judgment forms

The evaluation $(s, \sigma) \Rightarrow \tau$ expresses that running a statement s from a state σ produces a trace τ.

E.g.

$(x := 1 + 3; y := 2, (0, 0)) \Rightarrow (0, 0) :: (4, 0) :: \langle (4, 2) \rangle$

$(x := 1; \text{while true do } x := x + 1, (0)) \Rightarrow$
$(0) :: (1) :: (1) :: (2) :: (2) :: (3) :: (3) \ldots$

$(s, \sigma) \Rightarrow \tau$ is defined by mutual coinduction together with the extended evaluation $(s, \tau) \stackrel{*}{\Rightarrow} \tau'$.

The judgment forms

Big-step semantics

$(s, \tau) \overset{*}{\Rightarrow} \tau'$ expresses that running a statement s from the last state (if it exists) of an already accumulated trace τ results in a total trace τ'. Or:

$$\frac{(s, \sigma) \Rightarrow \tau}{(s, \langle \sigma \rangle) \overset{*}{\Rightarrow} \tau} \qquad \frac{(s, \tau) \overset{*}{\Rightarrow} \tau'}{(s, \sigma :: \tau) \overset{*}{\Rightarrow} \sigma :: \tau'}$$

E.g.

$(x := 1 + 3; y := 2, (0, 0) :: \langle (0, 1) \rangle) \overset{*}{\Rightarrow}$
$(0, 0) :: (0, 1) :: (4, 1) :: \langle (4, 2) \rangle$

Inference rules

Big-step semantics

$$\frac{}{(x := e, \sigma) \Rightarrow \sigma :: \langle \sigma[x \mapsto [\![e]\!]\sigma] \rangle}$$

$$\frac{}{(\text{skip}, \sigma) \Rightarrow \langle \sigma \rangle} \qquad \frac{(s_0, \sigma) \Rightarrow \tau \quad (s_1, \tau) \stackrel{*}{\Rightarrow} \tau'}{(s_0; s_1, \sigma) \Rightarrow \tau'}$$

$$\frac{\sigma \models e \quad (s_t, \sigma :: \langle \sigma \rangle) \stackrel{*}{\Rightarrow} \tau}{(\text{if } e \text{ then } s_t \text{ else } s_f, \sigma) \Rightarrow \tau} \qquad \frac{\sigma \not\models e \quad (s_f, \sigma :: \langle \sigma \rangle) \stackrel{*}{\Rightarrow} \tau}{(\text{if } e \text{ then } s_t \text{ else } s_f, \sigma) \Rightarrow \tau}$$

$$\frac{\sigma \models e \quad (s_t, \sigma :: \langle \sigma \rangle) \stackrel{*}{\Rightarrow} \tau \quad (\text{while } e \text{ do } s_t, \tau) \stackrel{*}{\Rightarrow} \tau'}{(\text{while } e \text{ do } s_t, \sigma) \Rightarrow \tau'}$$

$$\frac{\sigma \not\models e}{(\text{while } e \text{ do } s_t, \sigma) \Rightarrow \sigma :: \langle \sigma \rangle}$$

$$\frac{(s, \sigma) \Rightarrow \tau}{(s, \langle \sigma \rangle) \stackrel{*}{\Rightarrow} \tau} \qquad \frac{(s, \tau) \stackrel{*}{\Rightarrow} \tau'}{(s, \sigma :: \tau) \stackrel{*}{\Rightarrow} \sigma :: \tau'}$$

The effect of the extended evaluation $(s, \tau) \stackrel{*}{\Rightarrow} \tau'$

$(s, \tau) \stackrel{*}{\Rightarrow} \tau'$ is carefully crafted so that

- if τ is finite, then s is run from the last state of τ and τ' is obtained from τ by appending the trace produced by s;
- if τ is infinite, then $(s, \tau) \stackrel{*}{\Rightarrow} \tau'$ is derivable for any τ' bisimilar to τ, in particular for τ.

This design has the desirable consequence that, if $(s_0, \sigma) \Rightarrow \tau$ and τ is infinite, then $(s_1, \tau) \stackrel{*}{\Rightarrow} \tau$ is derivable and further so is $(s_0; s_1, \sigma) \Rightarrow \tau$.

So, s_1 is not run when s_0 diverges.

Moreover we need not decide if τ is finite or infinite, which we cannot decide constructively.

Design issues

Testing guards of the if- and while-statements augments the trace, but skip does not.

This is good for several reasons:

- skip is a unit of sequential composition.

  $\forall s, \sigma, \tau$, $(s; \text{skip}, \sigma) \Rightarrow \tau$ iff $(\text{skip}; s, \sigma) \Rightarrow \tau$ iff $(s, \sigma) \Rightarrow \tau$
- A notion of small steps that fully agrees with the textbook-style inductive small-step semantics.
- Any while-loop always progresses.

  $(\text{while true do skip}, \sigma) \Rightarrow \sigma :: \sigma :: \sigma :: \ldots$

What if ... (1)

Design issues

If we give up progress of loops and modify the rules for the while-loop to take the forms

$$\frac{\sigma \models e \quad (s_t, \sigma) \Rightarrow \tau \quad (\text{while } e \text{ do } s_t, \tau) \stackrel{*}{\Rightarrow} \tau'}{(\text{while } e \text{ do } s_t, \sigma) \Rightarrow \tau'} \qquad \frac{\sigma \not\models e}{(\text{while } e \text{ do } s_t, \sigma) \Rightarrow \langle \sigma \rangle}$$

then we get anomalies.

E.g.

$(\text{while true do skip}, \sigma) \Rightarrow \langle \sigma \rangle$

$(\text{while true do skip}; x := 17, \sigma) \Rightarrow \sigma :: \langle \sigma[x \mapsto 17] \rangle$

Indeed we have $(\text{while true do skip}, \sigma) \Rightarrow \tau$ for any τ!

What if ... (2)

Design issues

$$\forall \tau . \mathit{finite}\ \tau \vee \mathit{infinite}\ \tau$$

is not provable constructively.

If the semantics is given as the sum of

- an inductive trace-based semantics for terminating runs, and

- a coinductive trace-based semantics for non-terminating runs,

then we would stumble upon the halting problem.

Technical results (1)

The evaluation relation is a setoid predicate (insensitive to bisimilarity):

Lemma
For any σ, s, τ, τ'*, if* $(s, \sigma) \Rightarrow \tau$ *and* $\tau \approx \tau'$ *then* $(s, \sigma) \Rightarrow \tau'$*.*

It is deterministic (up to bisimilarity):

Lemma
For any σ, s, τ *and* τ'*, if* $(s, \sigma) \Rightarrow \tau$ *and* $(s, \sigma) \Rightarrow \tau'$ *then* $\tau \approx \tau'$*.*

Technical results (2)

It is equivalent to the textbook-style coinductive small-step counterpart:

Proposition

For any s, σ and τ, $(s, \sigma) \Rightarrow \tau$ iff $(s, \sigma) \rightsquigarrow \tau$.

It agrees with the standard inductive state-based semantics.

Proposition

For any s, σ, σ', existence of τ such that $(s, \sigma) \Rightarrow \tau$ and $\tau \downarrow \sigma'$ is equivalent to $(s, \sigma) \Rightarrow^{\mathrm{ind}} \sigma'$.

Hoare logic

Our Hoare-triple $\{U\}\ s\ \{P\}$ consists of

U : predicate on states

s : statement

P : predicate on traces

$\{U\}\ s\ \{P\}$ means that running a statement s from a initial state σ satisfying U produces a total trace τ satisfying P.

Notations

U, V : state predicates
P, Q : trace predicates

$\sigma \models U$ expresses that σ satisfies U.
$\tau \models P$ expresses that τ satisfies P.

Logical consequences and equivalence:

$$\frac{\forall \sigma\, (\sigma \models U \rightarrow \sigma \models V)}{U \models V} \qquad \frac{\forall \tau\, (\tau \models P \rightarrow \tau \models Q)}{P \models Q} \qquad \frac{P \models Q \quad Q \models P}{P \Leftrightarrow Q}$$

Assertions

$$\frac{\sigma \models U}{\langle \sigma \rangle \models \langle U \rangle} \qquad \frac{\sigma \models U}{\sigma :: \langle \sigma \rangle \models \langle U \rangle^2} \qquad \frac{\sigma \models U}{\sigma :: (\sigma[x \mapsto e]) \models U[x \mapsto e]}$$

$$\frac{\langle \sigma \rangle \models P}{\langle \sigma \rangle \models_{\langle \sigma \rangle} P} \qquad \frac{\sigma :: \tau \models P}{\sigma :: \tau \models_{\langle \sigma \rangle} P} \qquad \frac{\tau' \models_{\tau} P}{\sigma :: \tau' \models_{\sigma :: \tau} P}$$

$$\frac{\tau' \models P \quad \tau \models_{\tau'} Q}{\tau \models P * * Q} \qquad \frac{\tau \models \langle \mathsf{true} \rangle}{\tau \models P^{\dagger}} \qquad \frac{\tau' \models P \quad \tau \models_{\tau'} P^{\dagger}}{\tau \models P^{\dagger}}$$

Singleton operator $\langle U \rangle$

Assertions

$\langle U \rangle$ is a trace predicate that is true of a singleton trace given by a state satisfying U:

$$\frac{\sigma \models U}{\langle \sigma \rangle \models \langle U \rangle}$$

Doubleton operator $\langle U \rangle^2$

Assertions

$\langle U \rangle^2$ is a trace predicate that is true of a doubleton trace of an identical state satisfying U:

$$\frac{\sigma \models U}{\sigma :: \langle \sigma \rangle \models \langle U \rangle^2}$$

Update operator $U[x \mapsto e]$

Assertions

$U[x \mapsto e]$ is a trace predicate that is the strong postcondition of $x := e$ for the precondition U:

$$\frac{\sigma \models U}{\sigma :: \langle \sigma[x \mapsto e] \rangle \models U[x \mapsto e]}$$

Chop operator $P \ast\ast Q$

Assertions

Roughly, $\tau \models P \ast\ast Q$ holds when τ is split into two parts τ' and τ'' such that the last state of τ' is the first state of τ'' and the prefix τ' (resp. the postfix τ'') satisfies P (resp. Q):

$$\frac{\tau' \models P \quad \tau \models_{\tau'} Q}{\tau \models P \ast\ast Q}$$

$\tau \models_{\tau'} P$ first traverses τ', which must be a prefix of τ, then checks validity of P against the postfix:

$$\frac{\langle\sigma\rangle \models P}{\langle\sigma\rangle \models_{\langle\sigma\rangle} P} \quad \frac{\sigma :: \tau \models P}{\sigma :: \tau \models_{\langle\sigma\rangle} P} \quad \frac{\tau' \models_{\tau} P}{\sigma :: \tau' \models_{\sigma::\tau} P}$$

Chop operator $P \ast\ast Q$ (2)

Assertions

$$\frac{\langle\sigma\rangle \models P}{\langle\sigma\rangle \models_{\langle\sigma\rangle} P} \qquad \frac{\sigma :: \tau \models P}{\sigma :: \tau \models_{\langle\sigma\rangle} P} \qquad \frac{\tau' \models_{\tau} P}{\sigma :: \tau' \models_{\sigma::\tau} P}$$

Importantly $\tau \models_{\tau'} P$ necessarily holds when τ' is infinite.

But we need not decide if τ' is infinite or not.

Intuitively we delay checking of P until the last state of τ' is hit.

Consequently $\tau \models P \ast\ast Q$ has the desirable property that if *infinite* τ and $\tau \models P$ then $\tau \models P \ast\ast Q$ for any Q.

Iteration operator $P^\dagger$

Assertions

$P^\dagger$ is a trace predicate that is true of a trace that is zero or possibly infinite concatenations of traces, each of which satisfies P:

$$\frac{\tau \models \langle \mathsf{true} \rangle}{\tau \models P^\dagger} \qquad \frac{\tau' \models P \quad \tau \models_{\tau'} P^\dagger}{\tau \models P^\dagger}$$

In particular we have

$$P^\dagger \Leftrightarrow \langle \mathsf{true} \rangle \vee (P ** P^\dagger)$$

Inference rules of the Hoare logic

$$\frac{}{\{U\}\ x := e\ \{U[x \mapsto e]\}} \qquad \frac{}{\{U\}\ \mathsf{skip}\ \{\langle U\rangle\}}$$

$$\frac{\{U\}\ s_0\ \{P \ast\ast \langle V\rangle\} \quad \{V\}\ s_1\ \{Q\}}{\{U\}\ s_0; s_1\ \{P \ast\ast Q\}}$$

$$\frac{\{e \wedge U\}\ s_t\ \{P\} \quad \{\neg e \wedge U\}\ s_f\ \{P\}}{\{U\}\ \mathsf{if}\ e\ \mathsf{then}\ s_t\ \mathsf{else}\ s_f\ \{\langle U\rangle^2 \ast\ast P\}}$$

$$\frac{U \models I \quad \{e \wedge I\}\ s_t\ \{P \ast\ast \langle I\rangle\}}{\{U\}\ \mathsf{while}\ e\ \mathsf{do}\ s_t\ \{\langle U\rangle^2 \ast\ast (P \ast\ast \langle I\rangle^2)^\dagger \ast\ast \langle \neg e\rangle\}}$$

$$\frac{U \models U' \quad \{U'\}\ s\ \{P'\} \quad P' \models P}{\{U\}\ s\ \{P\}} \qquad \frac{\forall z.\ \{U\}\ s\ \{P\}}{\{\exists z.\ U\}\ s\ \{\exists z.\ P\}}$$

Soundness and Completeness

Proposition (Soundness)

For any s, U, P, σ, τ*, if* $\{U\}\ s\ \{P\}$ *and* $\sigma \models U$ *and* $(s, \sigma) \Rightarrow \tau$*, then* $\tau \models P$*.*

Proposition (Completeness)

For any s, U, P*, if for all* σ, τ*,* $\sigma \models U$ *and* $(s, \sigma) \Rightarrow \tau$ *imply* $\tau \models P$*, then* $\{U\}\ s\ \{P\}$*.*

Embedding of the standard Hoare logics

Proposition (Partial correctness)

For any U, s and V if $\{U\}\ s\ \{V\}$ is derivable in the partial correctness Hoare logic, then $\{U\}\ s\ \{\text{true} \ast\!\ast \langle V \rangle\}$.

Proof.

By induction on the derivation of $\{U\}\ s\ \{V\}$. □

Proposition (Total correctness)

For any U, s and V if $\{U\}\ s\ \{V\}$ is derivable in the total correctness Hoare logic, then $\{U\}\ s\ \{\mathit{finite} \ast\!\ast \langle V \rangle\}$.

Proof.

By induction on the derivation of $\{U\}\ s\ \{V\}$. □

Example

Our logic is expressive enough to perform the same analyses that the partial and total correctness Hoare logics can perform without additional verification overhead.

The expressivity of our logic goes beyond that of these standard logics.

E.g. we distinguish between termination and nondivergent.

Unbounded total search fails to be terminating but is still nondivergent.

Unbounded total search

Example

Variable $B : nat \to bool$
Axiom *B_noncontradictory*: $\neg(\forall n.\neg B\ n)$

$$Search \equiv x := 0; \text{while } \neg(B\ x) \text{ do } x := x + 1$$

Search fails to be terminating, but is nondivergent.

cf.
Markov's principle: $(\neg(\forall x, \neg B\ x)) \Rightarrow \exists x, B\ x$
is a classical tautology, but is not valid constructively.

Proof sketch

Unbounded total search is nondivergent

$$\frac{\sigma\ x = n \quad B\ n}{\sigma :: \langle\sigma\rangle \models \mathit{cofinally}\ n} \qquad \frac{\sigma\ x = n \quad \neg B\ n \quad \tau \models \mathit{cofinally}\ (n+1)}{\sigma :: \sigma :: \tau \models \mathit{cofinally}\ n}$$

Lemma

$\mathit{cofinally}\ 0 \models \neg \mathit{infinite}$.

Proof sketch (2)

Unbounded total search is nondivergent

$$\dfrac{\dfrac{\{\neg B\,x\}\ x := x+1\ \{(\neg B\,x)[x \mapsto x+1]\}}{\{x=0\}\ \text{while}\ \neg B\,x\ \text{do}\ x := x+1\ \{\langle x=0\rangle^2 \ast\ast ((\neg B\,x)[x \mapsto x+1] \ast\ast \langle \text{true}\rangle^2)^\dagger \ast\ast \langle B\ x\rangle\}} \qquad \dfrac{\dfrac{}{\{\text{true}\}\ x := 0\ \{\text{true}[x \mapsto 0]\}} \quad \dfrac{\dfrac{}{\{x=0\}\text{while}\ \neg B\,x\ \text{do}\ x := x+1\ \{\textit{cofinally}\ 0\}}}{\{\text{true}\}x := 0; \text{while}\ \neg B\,x\ \text{do}\ x := x+1\ \{\text{true}[x \mapsto 0] \ast\ast \textit{cofinally}\ 0\}}}{\{\text{true}\}x := 0; \text{while}\ \neg B\,x\ \text{do}\ x := x+1\ \{(\text{true} \ast\ast \langle B\ x\rangle) \wedge \neg \textit{infinite}\}}$$

Summary & Future work

I have presented

- a trace-based coinductive big-step semantics
- a Hoare logic

Extending our framework with function calls and exceptions is straightforward, as found in the literature.

We are now working on language-based information flow analysis, to exploit the extra expressivity of our framework.