Resumptions, weak bisimilarity and big-step semantics for interactive I/O

An exercise in mixed induction-coinduction

Tarmo Uustalu, Institute of Cybernetics, Tallinn
joint work with Keiko Nakata

TSEM, Tallinn, 13 May 2010

Motivation

- Now and then we need rigorous semantic descriptions and program logics for languages where programs may not terminate, but can perform observable actions
- Especially relevant in compiler correctness, program analysis/verification
- Big-step semantics: operational (an abstract machine), but nonetheless compositional (close to denotational)—useful in metatheory
- Typical operational accounts of computation with observable actions are small-step
- Constructive type theory: a "finer" mathematics, only computable functions exist—a useful stance for people concerned with "executable specification"

This talk

- Big-step semantics for While (imperative programs) with interactive I/O
- We only want to observe I/O actions
- We want to identify any finite sequences of delays (silent actions)
- Main tools: interactive I/O resumptions (computation trees), termination-sensitive weak bisimilarity
- This needs coinduction and mixed induction-coinduction
- Inspiration for the bigger context: Leroy's certified compiler project and coinductive big-step semantics

Outline

- Basic (delayful) resumptions, strong bisimilarity, responsive and commited resumptions
- Weak bisimilarity
- Basic (delayful) semantics
- Delay-free resumptions, delay-free semantics (removes finite sequences of delays on-the-fly)
- Classical-style resumptions, classical-style semantics (removes finite sequences of delays on-the-fly but also detects divergence)

Syntax; states

- We extend While with input and output statements:

$$\begin{array}{rcl} s & ::= & \mathsf{skip} \mid s_0; s_1 \mid x := e \mid \mathsf{if}\ e\ \mathsf{then}\ s_t\ \mathsf{else}\ s_f \mid \mathsf{while}\ e\ \mathsf{do}\ s_t \\ & & \mid \mathsf{input}\ x \mid \mathsf{output}\ e \end{array}$$

- States: assignments of integers to variables.

Basic (delayful) resumptions

- Resumptions (computation trees), δ corresponds to an internal action, observable as a delay:

$$\frac{\sigma : \mathit{state}}{\mathit{ret}\ \sigma : \mathit{res}} \quad \frac{f : \mathit{Int} \to \mathit{res}}{\mathit{in}\ f : \mathit{res}} \quad \frac{v : \mathit{Int} \quad r : \mathit{res}}{\mathit{out}\ v\ r : \mathit{res}} \quad \frac{r : \mathit{res}}{\delta\ r : \mathit{res}}$$

  (double horizontal rules—coinductive definition, single—inductive definition)

- Strong bisimilarity:

$$\frac{}{\mathit{ret}\ \sigma \approx \mathit{ret}\ \sigma} \quad \frac{\forall v.\, f\ v \approx f_*\ v}{\mathit{in}\ f \approx \mathit{in}\ f_*} \quad \frac{r \approx r_*}{\mathit{out}\ v\ r \approx \mathit{out}\ v\ r_*} \quad \frac{r \approx r_*}{\delta\ r \approx \delta\ r_*}$$

- We think of strong bisimilarity as equality of resumptions, but in intensional type theory, it is weaker
  (just as equality of two functions on all argument values does not entail their equality).

Convergence and divergence

- Convergence (a resumption eventually terminates or does a properly observable action):

$$\frac{}{\mathit{ret}\ \sigma \downarrow \mathit{ret}\ \sigma} \quad \frac{}{\mathit{in}\ f \downarrow \mathit{in}\ f} \quad \frac{}{\mathit{out}\ v\ r \downarrow \mathit{out}\ v\ r} \quad \frac{r \downarrow r'}{\delta\ r \downarrow r'}$$

- Divergence (it keeps doing internal actions forever)

$$\frac{r \uparrow}{\delta\ r \uparrow}$$

- Both are setoid predicates, i.e., stable under strong bisimilarity.
- Classically, one is the negation of the other.
- Constructively, we have $\forall r.\, \neg\, (\exists r'.\, r \downarrow r') \rightarrow r \uparrow$, but neither $\forall r.\, \neg\, r \uparrow \rightarrow \exists r'.\, r \downarrow r'$ nor $\forall r.\, (\exists r'.\, r \downarrow r') \vee r \uparrow$.

Responsiveness

- Responsiveness (a resumption keeps converging):

$$\frac{r \downarrow \mathit{ret}\ \sigma}{r \Downarrow} \qquad \frac{r \downarrow \mathit{in}\ f \quad \forall v.\,(f\ \ v)\Downarrow}{r \Downarrow} \qquad \frac{r \downarrow \mathit{out}\ v\ \ r' \quad r' \Downarrow}{r \Downarrow}$$

- Responsiveness is the analogue in interactive computation of termination from non-interactive computation.

Commitedness

- Commitedness (a resumption always converges or diverges):

$$\frac{r \downarrow \mathit{ret}\ \sigma}{r \Updownarrow} \qquad \frac{r \downarrow \mathit{in}\ f \quad \forall v.\,(f\ \ v) \Updownarrow}{r \Updownarrow} \qquad \frac{r \downarrow \mathit{out}\ \ v\ \ r' \quad r' \Updownarrow}{r \Updownarrow}$$

$$\frac{r \uparrow}{r \Updownarrow}$$

- Both responsiveness and commitedness are setoid predicates.
- Classically, every resumption is committed. Constructively, this is not the case (cannot decide halting).

Weak bisimilarity

- In process calculi, divergence is identified with termination. We reject this view as unrealistic. Our version is "termination-sensitive".
- Weak bisimilarity (two resumptions keep converging to related resumptions):

$$\frac{r \downarrow \mathit{ret}\ \sigma \qquad r_* \downarrow \mathit{ret}\ \sigma}{r \cong r_*} \qquad \frac{r \downarrow \mathit{in}\ f \qquad \forall v.\, f\ v \cong f_*\ v \qquad r_* \downarrow \mathit{in}\ f_*}{r \cong r_*}$$

$$\frac{r \downarrow \mathit{out}\ v\ r' \qquad r' \cong r'_* \qquad r_* \downarrow \mathit{out}\ v\ r'_*}{r \cong r_*}$$

$$\frac{r \cong r_*}{\delta\ r \cong \delta\ r_*}$$

Classical-style weak bisimilarity

- Classical-style version:

$$\frac{r \downarrow \mathit{ret}\ \sigma \qquad r_* \downarrow \mathit{ret}\ \sigma}{r \cong_{\mathrm{c}} r_*} \qquad \frac{r \downarrow \mathit{out}\ v\ r' \qquad r' \cong_{\mathrm{c}} r'_* \qquad r_* \downarrow \mathit{out}\ v\ r'_*}{r \cong_{\mathrm{c}} r_*}$$

$$\frac{r \downarrow \mathit{in}\ f \qquad r_* \downarrow \mathit{in}\ f_* \qquad \forall v.\, f\ v \cong_{\mathrm{c}} f_*\ v}{r \cong_{\mathrm{c}} r_*}$$

$$\frac{r \uparrow \qquad r_* \uparrow}{r \cong_{\mathrm{c}} r_*}$$

- Classically, the two notions are equivalent.
- Constructively, the classical-style version is stronger.

Basic (delayful) semantics

- Evaluation: associates a resumption to an (initial state):

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

$$\frac{}{(\mathrm{skip}, \sigma) \Rightarrow ret\ \sigma} \qquad \frac{(s_0, \sigma) \Rightarrow r \quad (s_1, r) \overset{*}{\Rightarrow} r'}{(s_0; s_1, \sigma) \Rightarrow r'}$$

$$\frac{e \models \sigma \quad (s_t, \delta\ (ret\ \sigma)) \overset{*}{\Rightarrow} r}{(\mathrm{if}\ e\ \mathrm{then}\ s_t\ \mathrm{else}\ s_f, \sigma) \Rightarrow r} \qquad \frac{e \not\models \sigma \quad (s_f, \delta\ (ret\ \sigma)) \overset{*}{\Rightarrow} r}{(\mathrm{if}\ e\ \mathrm{then}\ s_t\ \mathrm{else}\ s_f, \sigma) \Rightarrow r}$$

$$\frac{e \models \sigma \quad (s_t, \delta\ (ret\ \sigma)) \overset{*}{\Rightarrow} r \quad (\mathrm{while}\ e\ \mathrm{do}\ s_t, r) \overset{*}{\Rightarrow} r'}{(\mathrm{while}\ e\ \mathrm{do}\ s_t, \sigma) \Rightarrow r'}$$

$$\frac{e \not\models \sigma}{(\mathrm{while}\ e\ \mathrm{do}\ s_t, \sigma) \Rightarrow \delta\ (ret\ \sigma)}$$

$$\frac{}{(\mathrm{input}\ x, \sigma) \Rightarrow in\ (\lambda v.ret\ \sigma[x \mapsto v])} \qquad \frac{}{(\mathrm{output}\ e, \sigma) \Rightarrow out\ ([\![e]\!]\sigma)\ (ret\ \sigma)}$$

Basic (delayful) semantics ctd

- Extended evaluation: associates a (total) resumption to an (already accumulated) resumption:

$$\frac{(s,\sigma) \Rightarrow r}{(s, ret\ \sigma) \overset{*}{\Rightarrow} r} \qquad \frac{\forall v.\,(s, f\ v) \overset{*}{\Rightarrow} f'\ v}{(s, in\ f) \overset{*}{\Rightarrow} in\ f'} \qquad \frac{(s,r) \overset{*}{\Rightarrow} r'}{(s, out\ v\ r) \overset{*}{\Rightarrow} out\ v\ r'}$$

$$\frac{(s,r) \overset{*}{\Rightarrow} r'}{(s, \delta\ r) \overset{*}{\Rightarrow} \delta\ r'}$$

  i.e., the coinductive prefix closure of evaluation.
- Design choice: evaluation of an expression to assign/update of a variable and evaluation of a guard constitute internal actions, observable as delays.
- Consideration: every loop always progresses, for instance, while true do skip is not weakly bisimilar to skip.

Delay-free resumptions

- Delay-free resumptions (the delay constructor removed):

$$\frac{\sigma : \mathit{state}}{\mathit{ret}_{\mathrm{r}}\ \sigma : \mathit{res}_{\mathrm{r}}} \qquad \frac{f : \mathit{Int} \longrightarrow \mathit{res}_{\mathrm{r}}}{\mathit{in}_{\mathrm{r}}\ f : \mathit{res}_{\mathrm{r}}} \qquad \frac{v : \mathit{Int} \quad r : \mathit{res}_{\mathrm{r}}}{\mathit{out}_{\mathrm{r}}\ v\ r : \mathit{res}_{\mathrm{r}}}$$

- Delay-free resumptions embed in delayful resumptions.
- Any responsive resumption normalizes to a weakly bisimilar delay-free resumption.

Delay-free semantics

- Convergent evaluation: inductive, parameterized over a relation X, to be instantiated with the coinductive extended evaluation:

$$\frac{}{(x := e, \sigma) \Rightarrow\downarrow(X)\ ret_{\mathrm{r}}\ (\sigma[x \mapsto [\![e]\!]\sigma])}$$

$$\frac{}{(\mathrm{skip}, \sigma) \Rightarrow\downarrow(X)\ ret_{\mathrm{r}}\ \sigma} \qquad \frac{(s_0, \sigma) \Rightarrow\downarrow(X)\ ret_{\mathrm{r}}\ \sigma' \quad (s_1, \sigma') \Rightarrow\downarrow(X)\ r}{(s_0; s_1, \sigma) \Rightarrow\downarrow(X)\ r}$$

$$\frac{(s_0, \sigma) \Rightarrow\downarrow(X)\ in_{\mathrm{r}}\ f \quad \forall v.\, (s_1, f\ v)\ X\ f'\ v}{(s_0; s_1, \sigma) \Rightarrow\downarrow(X)\ in_{\mathrm{r}}\ f'} \qquad \frac{(s_0, \sigma) \Rightarrow\downarrow(X)\ out_{\mathrm{r}}\ v\ r \quad (s_1, r)\ X\ r'}{(s_0; s_1, \sigma) \Rightarrow\downarrow(X)\ out_{\mathrm{r}}\ v\ r'}$$

$$\frac{e \models \sigma \quad (s_t, \sigma) \Rightarrow\downarrow(X)\ r}{(\mathrm{if}\ e\ \mathrm{then}\ s_t\ \mathrm{else}\ s_f, \sigma) \Rightarrow\downarrow(X)\ r} \qquad \frac{e \not\models \sigma \quad (s_f, \sigma) \Rightarrow\downarrow(X)\ r}{(\mathrm{if}\ e\ \mathrm{then}\ s_t\ \mathrm{else}\ s_f, \sigma) \Rightarrow\downarrow(X)\ r}$$

$$\frac{e \models \sigma \quad (s_t, \sigma) \Rightarrow\downarrow(X)\ ret_{\mathrm{r}}\ \sigma' \quad (\mathrm{while}\ e\ \mathrm{do}\ s_t, \sigma') \Rightarrow\downarrow(X)\ r}{(\mathrm{while}\ e\ \mathrm{do}\ s_t, \sigma) \Rightarrow\downarrow(X)\ r}$$

$$\frac{e \models \sigma \quad (s_t, \sigma) \Rightarrow\downarrow(X)\ in_{\mathrm{r}}\ f \quad \forall v.\, (\mathrm{while}\ e\ \mathrm{do}\ s_t, f\ v)\ X\ f'\ v}{(\mathrm{while}\ e\ \mathrm{do}\ s_t, \sigma) \Rightarrow\downarrow(X)\ in_{\mathrm{r}}\ f'}$$

$$\frac{e \models \sigma \quad (s_t, \sigma) \Rightarrow\downarrow(X)\ out_{\mathrm{r}}\ v\ r \quad (\mathrm{while}\ e\ \mathrm{do}\ s_t, r)\ X\ r'}{(\mathrm{while}\ e\ \mathrm{do}\ s_t, \sigma) \Rightarrow\downarrow(X)\ out_{\mathrm{r}}\ v\ r'}$$

$$\frac{e \not\models \sigma}{(\mathrm{while}\ e\ \mathrm{do}\ s_t, \sigma) \Rightarrow\downarrow(X)\ ret_{\mathrm{r}}\ \sigma}$$

Delay-free semantics ctd

- Extended evaluation: coinductive:

$$\frac{\begin{bmatrix}\dfrac{(s,\sigma) \Rightarrow\downarrow(\overset{*}{\Rightarrow})\ r}{(s, ret_{\mathrm{r}}\ \sigma) \overset{*}{\Rightarrow} r}\end{bmatrix} \\ X \subseteq \overset{*}{\Rightarrow} \quad (s,\sigma) \Rightarrow\downarrow(X)\ r}{(s, ret_{\mathrm{r}}\ \sigma) \overset{*}{\Rightarrow} r}$$

$$\frac{\forall v.\, (s, f\ v) \overset{*}{\Rightarrow} f'\ v}{(s, in_{\mathrm{r}}\ f) \overset{*}{\Rightarrow} in_{\mathrm{r}}\ f'} \qquad \frac{(s,r) \overset{*}{\Rightarrow} r'}{(s, out_{\mathrm{r}}\ v\ r) \overset{*}{\Rightarrow} out_{\mathrm{r}}\ v\ r'}$$

- Evaluation:

$$\frac{(s,r) \Rightarrow\downarrow(\overset{*}{\Rightarrow})\ r'}{(s,r) \Rightarrow_{\mathrm{r}} r'}$$

- The delay-free semantics agrees with the delayful semantics for responsive resumptions.

Classical-style resumptions

- Classical-style resumptions (a special constructor for divergence added to delay-free resumptions):

$$\frac{\sigma : \mathit{state}}{\mathit{ret}_{\mathrm{c}}\ \sigma : \mathit{res}_{\mathrm{c}}} \qquad \frac{f : \mathit{Int} \to \mathit{res}_{\mathrm{c}}}{\mathit{in}_{\mathrm{c}}\ f : \mathit{res}_{\mathrm{c}}} \qquad \frac{r : \mathit{res}_{\mathrm{c}}}{\mathit{out}_{\mathrm{c}}\ v\ r : \mathit{res}_{\mathrm{c}}} \qquad \frac{}{\bullet : \mathit{res}_{\mathrm{c}}}$$

- Classical-style resumptions embed in delayful resumptions ($\mathit{emb}\,\bullet = \delta\ (\mathit{emb}\,\bullet)$).
- Any committed resumption (classically, any resumption) normalizes to a weakly bisimilar delay-free resumption.

Classical-style semantics ctd

- Convergent evaluation: as in the delay-free semantics (inductive)
- Divergent evaluation: a new coinductive relation, also parameterized in X:

$$\begin{array}{c} (s_0, \sigma) \Rightarrow\uparrow(X) \\ \hline\hline (s_0; s_1, \sigma) \Rightarrow\uparrow(X) \end{array} \qquad \begin{array}{c} (s_0, \sigma) \Rightarrow\downarrow(X)\ \mathit{ret}_{\mathrm{c}}\ \sigma' \quad (s_1, \sigma') \Rightarrow\uparrow(X) \\ \hline\hline (s_0; s_1, \sigma) \Rightarrow\uparrow(X) \end{array}$$

$$\begin{array}{c} e \models \sigma \quad (s_t, \sigma) \Rightarrow\uparrow(X) \\ \hline\hline (\mathsf{if}\ e\ \mathsf{then}\ s_t\ \mathsf{else}\ s_f, \sigma) \Rightarrow\uparrow(X) \end{array} \quad \begin{array}{c} e \not\models \sigma \quad (s_f, \sigma) \Rightarrow\uparrow(X) \\ \hline\hline (\mathsf{if}\ e\ \mathsf{then}\ s_t\ \mathsf{else}\ s_f, \sigma) \Rightarrow\uparrow(X) \end{array}$$

$$\begin{array}{c} e \models \sigma \quad (s_t, \sigma) \Rightarrow\uparrow(X) \\ \hline\hline (\mathsf{while}\ e\ \mathsf{do}\ s_t, \sigma) \Rightarrow\uparrow(X) \end{array}$$

$$\begin{array}{c} e \models \sigma \quad (s_t, \sigma) \Rightarrow\downarrow(X)\ \mathit{ret}_{\mathrm{c}}\ \sigma' \quad (\mathsf{while}\ e\ \mathsf{do}\ s_t, \sigma') \Rightarrow\uparrow(X) \\ \hline\hline (\mathsf{while}\ e\ \mathsf{do}\ s_t, \sigma) \Rightarrow\uparrow(X) \end{array}$$

(divergent evaluation depends on convergent evaluation, but not the other way around)

Classical-style semantics ctd

- Extended evaluation: as in the delay-free semantics, but with a choice between convergent and divergent evaluation, and an additional case for the accumulated resumption being a black hole:

$$
\begin{array}{c} X \subseteq \stackrel{*}{\Rightarrow} \quad (s, \sigma) \Rightarrow\downarrow(X)\ r \\ \hline\hline (s, \mathit{ret}_{\mathrm{c}}\ \sigma) \stackrel{*}{\Rightarrow} r \end{array}
\qquad
\begin{array}{c} X \subseteq \stackrel{*}{\Rightarrow} \quad (s, \sigma) \Rightarrow\uparrow(X) \\ \hline\hline (s, \mathit{ret}_{\mathrm{c}}\ \sigma) \stackrel{*}{\Rightarrow} \bullet \end{array}
$$

$$
\begin{array}{c} \forall v.\, (s, f\ v) \stackrel{*}{\Rightarrow} f'\ v \\ \hline\hline (s, \mathit{in}_{\mathrm{c}}\ f) \stackrel{*}{\Rightarrow} \mathit{in}_{\mathrm{c}}\ f' \end{array}
\qquad
\begin{array}{c} (s, r) \stackrel{*}{\Rightarrow} r' \\ \hline\hline (s, \mathit{out}_{\mathrm{c}}\ v\ r) \stackrel{*}{\Rightarrow} \mathit{out}_{\mathrm{c}}\ v\ r' \end{array}
\qquad
\begin{array}{c} \\ \hline\hline (s, \bullet) \stackrel{*}{\Rightarrow} \bullet \end{array}
$$

- Evaluation: again a choice between convergent and divergent evaluation:

$$
\frac{(s, \sigma) \Rightarrow\downarrow(\stackrel{*}{\Rightarrow})\ r}{(s, \sigma) \Rightarrow_{\mathrm{c}} r}
\qquad
\frac{(s, \sigma) \Rightarrow\uparrow(\stackrel{*}{\Rightarrow})}{(s, \sigma) \Rightarrow_{\mathrm{c}} \bullet}
$$

- The classical-style semantics agrees with the delayful semantics for commited resumptions (classically, all resumptions).

Conclusion

- Constructive glasses give a nuanced picture of subtle concepts like weak bisimilarity.
- Interactive I/O is not considerably more complicated than non-interactive (batch) computation—the same basic considerations apply and help.
- We hope we can also to treat concurrency.
- An interesting exercise with mixed inductive-coinductive definitions of the form $\nu X\, \mu Y\, F(X, Y)$.
- In the Coq development we had to parameterized inductive types and Mendler-style coinductive types (to facilitate guarded corecursion).
- For more details, check our TPHOLs 2009, ESOP 2010 papers and a new submission, and the accompanying Coq code.