Coinductive big-step semantics for concurrency

Tarmo Uustalu, Institute of Cybernetics, Tallinn

Theory Days at Otepää, 1–3 February 2013

Two popular misconceptions

- Big-step operational semantics cannot account for diverging program behaviors.
- One cannot give big-step semantics for languages for concurrency: concurrency is inherently small-step.
- This talk: We refute both.
- To capture divergence, use the idea of possibly infinitely delayed states (taking it seriously that internal actions of a program take time).
- To deal with concurrency, use a suitable notion of resumptions/traces.
- We develop the metatheory of our semantics in a constructive setting.

Syntax

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

Big-step semantics: Resumptions

$$\frac{\sigma : state}{ret\ \sigma : res} \quad \frac{r : res}{\delta\ r : res} \quad \frac{r_0 : res \quad r_1 : res}{r_0 + r_1 : res} \quad \frac{s : stmt \quad \sigma : state}{yield\ s\ \sigma : res}$$

(double rule lines for coinductive definitions)

$ret\ \sigma$ - terminated computation
$\delta\ r$ - computation that first makes an internal action
$r_0 + r_1$ - computation that first make a choice
$yield\ s\ \sigma$ - suspended computation

eg
$\delta^3\,(\delta^2\,(ret\ [x \mapsto 5])$
$\quad +\delta^4\,(\delta^\infty + \delta\,(yield\ x := x + 7\ [x \mapsto 3])))$

Evaluation

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

$$\frac{}{\mathsf{skip}, \sigma \Rightarrow \mathit{ret}\ \sigma} \qquad \frac{s_0, \sigma \Rightarrow r \quad s_1, r \Rightarrow^{\mathrm{seq}} r'}{s_0; s_1, \sigma \Rightarrow r'}$$

$$\frac{\sigma \models e}{\mathsf{if}\ e\ \mathsf{then}\ s_t\ \mathsf{else}\ s_f, \sigma \Rightarrow \delta\ (\mathit{yield}\ s_t\ \sigma)}$$

$$\frac{\sigma \not\models e}{\mathsf{if}\ e\ \mathsf{then}\ s_t\ \mathsf{else}\ s_f, \sigma \Rightarrow \delta\ (\mathit{yield}\ s_f\ \sigma)}$$

$$\frac{\sigma \models e}{\mathsf{while}\ e\ \mathsf{do}\ s_t, \sigma \Rightarrow \delta\ (\mathit{yield}\ (s_t; \mathsf{while}\ e\ \mathsf{do}\ s_t)\ \sigma)}$$

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

$$\frac{s_0, \sigma \Rightarrow r_0 \quad s_1, r_0 \Rightarrow^{\text{par}} r_0' \quad s_1, \sigma \Rightarrow r_1 \quad s_0, r_1 \Rightarrow^{\text{par}} r_1'}{s_0 \parallel s_1, \sigma \Rightarrow r_0' + r_1'}$$

$$\frac{s, \sigma \Rightarrow r \quad r \rightsquigarrow r'}{\text{atomic } s, \sigma \Rightarrow r'}$$

$$\frac{\sigma \models e \quad s, \sigma \Rightarrow r \quad r \rightsquigarrow r'}{\text{await } e \text{ do } s, \sigma \Rightarrow \delta\ r'}$$

$$\frac{\sigma \not\models e}{\text{await } e \text{ do } s, \sigma \Rightarrow \delta\ (\mathit{yield}\ (\text{await } e \text{ do } s)\ \sigma)}$$

Sequential | parallel extension of evaluation

$$\frac{}{s, \mathit{ret}\ \sigma \Rightarrow^{\mathrm{seq}} \mathit{yield}\ s\ \sigma} \qquad \frac{s, r \Rightarrow^{\mathrm{seq}} r'}{s, \delta\ r \Rightarrow^{\mathrm{seq}} \delta\ r'} \qquad \frac{s, r_0 \Rightarrow^{\mathrm{seq}} r_0' \quad s, r_1 \Rightarrow^{\mathrm{seq}} r_1'}{s, r_0 + r_1 \Rightarrow^{\mathrm{seq}} r_0' + r_1'}$$

$$\frac{}{s, \mathit{yield}\ s_0\ \sigma \Rightarrow^{\mathrm{par}} \mathit{yield}\ (s_0; s)\ \sigma}$$

$$\frac{}{s, \mathit{ret}\ \sigma \Rightarrow^{\mathrm{par}} \mathit{yield}\ s\ \sigma} \qquad \frac{s, r \Rightarrow^{\mathrm{par}} r'}{s, \delta\ r \Rightarrow^{\mathrm{par}} \delta\ r'} \qquad \frac{s, r_0 \Rightarrow^{\mathrm{par}} r_0' \quad s, r_1 \Rightarrow^{\mathrm{par}} r_1'}{s, r_0 + r_1 \Rightarrow^{\mathrm{par}} r_0' + r_1'}$$

$$\frac{}{s, \mathit{yield}\ s_0\ \sigma \Rightarrow^{\mathrm{par}} \mathit{yield}\ (s_0 \parallel s)\ \sigma}$$

Closing a resumption

$$\frac{}{\mathit{ret}\ \sigma \rightsquigarrow \mathit{ret}\ \sigma} \qquad \frac{r \rightsquigarrow r'}{\delta\ r \rightsquigarrow \delta\ r'} \qquad \frac{r_0 \rightsquigarrow r_0' \quad r_1 \rightsquigarrow r_1'}{r_0 + r_1 \rightsquigarrow r_0' + r_1'} \qquad \frac{s, \sigma \Rightarrow r \quad r \rightsquigarrow r'}{\mathit{yield}\ s\ \sigma \rightsquigarrow \delta\ r'}$$

Examples

$s = x := 1 \parallel (x := x + 2; x := x + 2)$,
$\sigma = [x \mapsto 0]$,

$s, \sigma \Rightarrow \delta(\mathit{yield}\ (x := x + 2; x := x + 2)\ [x \mapsto 1])$
$\quad + \delta(\mathit{yield}\ (x := 1 \parallel x := x + 2)\ [x \mapsto 2])$,
atomic $s, \sigma \Rightarrow$
$\delta^5(\mathit{ret}\ [x \mapsto 5]) + \delta^2(\delta^3(\mathit{ret}\ [x \mapsto 3]) + \delta^3(\mathit{ret}\ [x \mapsto 1]))$.

$s = (\text{await } x = 0 \text{ do } x := 1) \parallel x := 2$,
$\sigma = [x \mapsto 0]$,

$s, \sigma \Rightarrow \delta^2(\mathit{yield}\ x := 2\ [x \mapsto 1])$
$\quad + \delta^1(\mathit{yield}\ (\text{await } x = 0 \text{ do } x := 1)\ [x \mapsto 2])$,
atomic $s, \sigma \Rightarrow \delta^4(\mathit{ret}\ [x \mapsto 2]) + \delta^\infty$.

Strong bisimilarity

$$\frac{}{ret\ \sigma \sim ret\ \sigma} \qquad \frac{r \sim r_*}{\delta\ r \sim \delta\ r_*} \qquad \frac{r_0 \sim r_{0*} \quad r_1 \sim r_{1*}}{r_0 + r_1 \sim r_{0*} + r_{1*}} \qquad \frac{}{yield\ s\ \sigma \sim yield\ s\ \sigma}$$

Small-step semantics: Extended configurations

$$\frac{\sigma : \mathit{state}}{\mathit{ret}\ \sigma : \mathit{xcfg}} \qquad \frac{s : \mathit{stmt} \quad \sigma : \mathit{state}}{\delta\ (s, \sigma) : \mathit{xcfg}}$$

$$\frac{s_0 : \mathit{stmt} \quad \sigma_0 : \mathit{state} \quad s_1 : \mathit{stmt} \quad \sigma_1 : \mathit{state}}{(s_0, \sigma_0) + (s_1, \sigma_1) : \mathit{xcfg}}$$

$$\frac{s : \mathit{stmt} \quad \sigma : \mathit{state}}{\mathit{yield}\ s\ \sigma : \mathit{xcfg}}$$

Single-step reduction

$$\frac{}{x := e, \sigma \to \delta\ (\mathsf{skip}, \sigma[x \mapsto [\![e]\!]\ \sigma])} \qquad \frac{}{\mathsf{skip}, \sigma \to \mathit{ret}\ \sigma}$$

$$\frac{s_0, \sigma \to \mathit{ret}\ \sigma'}{s_0; s_1, \sigma \to \mathit{yield}\ s_1 \sigma'} \qquad \frac{s_0, \sigma \to \delta\ (s_0', \sigma')}{s_0; s_1, \sigma \to \delta\ (s_0'; s_1, \sigma')}$$

$$\frac{s_0, \sigma \to (s_{00}, \sigma_0) + (s_{01}, \sigma_1)}{s_0; s_1, \sigma \to (s_{00}; s_1, \sigma_0) + (s_{01}; s_1, \sigma_1)} \qquad \frac{s_0, \sigma \to \mathit{yield}\ s_0'\ \sigma'}{s_0; s_1, \sigma \to \mathit{yield}\ (s_0'; s_1)\ \sigma'}$$

$$\frac{\sigma \models e}{\text{if } e \text{ then } s_t \text{ else } s_f, \sigma \to \delta\ (\mathsf{skip}; s_t, \sigma)}$$

$$\frac{\sigma \not\models e}{\text{if } e \text{ then } s_t \text{ else } s_f, \sigma \to \delta\ (\mathsf{skip}; s_f, \sigma)}$$

$$\frac{\sigma \models e}{\text{while } e \text{ do } s_t, \sigma \to \delta\ (\mathsf{skip}; (s_t; \text{while } e \text{ do } s_t), \sigma)}$$

$$\frac{\sigma \not\models e}{\text{while } e \text{ do } s_t, \sigma \to \delta\ (\mathsf{skip}, \sigma)}$$

$$\frac{}{s_0 \parallel s_1, \sigma \to (s_0 \between s_1, \sigma) + (s_1 \between s_0, \sigma)}$$

$$\frac{s_0, \sigma \to \mathit{ret}\ \sigma'}{s_0 \between s_1, \sigma \to \mathit{yield}\ s_1 \sigma'} \qquad \frac{s_0, \sigma \to \delta\ (s_0', \sigma')}{s_0 \between s_1, \sigma \to \delta\ (s_0' \between s_1, \sigma')}$$

$$\frac{s_0, \sigma \to (s_{00}, \sigma_0) + (s_{01}, \sigma_1)}{s_0 \between s_1, \sigma \to (s_{00} \between s_1, \sigma_0) + (s_{01} \between s_1, \sigma_1)} \qquad \frac{s_0, \sigma \to \mathit{yield}\ s_0'\ \sigma'}{s_0 \between s_1, \sigma \to \mathit{yield}\ (s_0' \parallel s_1)\ \sigma}$$

$$\frac{s, \sigma \to \mathit{ret}\ \sigma'}{\text{atomic } s, \sigma \to \mathit{ret}\ \sigma'} \qquad \frac{s, \sigma \to \delta\ (s', \sigma')}{\text{atomic } s, \sigma \to \delta\ (\text{atomic } s', \sigma')}$$

$$\frac{s, \sigma \to (s_0, \sigma_0) + (s_1, \sigma_1)}{\text{atomic } s, \sigma \to (\text{atomic } s_0, \sigma_0) + (\text{atomic } s_1, \sigma_1)} \qquad \frac{s, \sigma \to \mathit{yield}\ s'\ \sigma'}{\text{atomic } s, \sigma \to \delta\ (\text{atomic } s', \sigma'}$$

$$\frac{\sigma \models e}{\text{await } e \text{ do } s, \sigma \to \delta\ (\text{atomic } s, \sigma)}$$

$$\frac{\sigma \not\models e}{\text{await } e \text{ do } s, \sigma \to \delta\ (\text{skip}; \text{await } e \text{ do } s, \sigma)}$$

Maximal multi-step reduction

$$\frac{s, \sigma \to \mathit{ret}\ s'}{s, \sigma \to^{\mathrm{m}} \mathit{ret}\ s'}$$

$$\frac{s, \sigma \to \delta\ (s', \sigma') \quad s', \sigma' \to^{\mathrm{m}} r}{s, \sigma \to^{\mathrm{m}} \delta\ r}$$

$$\frac{s, \sigma \to (s_0, \sigma_0) + (s_1, \sigma_1) \quad s_0, \sigma_0 \to^{\mathrm{m}} r_0 \quad s_1, \sigma_1 \to^{\mathrm{m}} r_1}{s, \sigma \to^{\mathrm{m}} r_0 + r_1}$$

$$\frac{s, \sigma \to \mathit{yield}\ s'\ \sigma'}{s, \sigma \to^{\mathrm{m}} \mathit{yield}\ s'\ \sigma'}$$

Maximal multi-step reduction agrees with evaluation: $s, \sigma \Rightarrow r$ iff $s, \sigma \to^{\mathrm{m}} r$.

Traces instead of resumptions

Remove the $+$ constructor from the definitions of resumptions and extended configurations.

Instead of a single rule with $+$ in the resumption resp. extended configuration in the conclusion make two rules for evaluation and single-step reduction for $\|$. turning evaluation and single-step reduction non-deterministic.

Giant-step semantics

Replace the rule

$$\frac{s : stmt \quad \sigma : state}{yield\ s\ \sigma : res}$$

in the definition of resumptions by

$$\frac{f : state \to res \quad \sigma : state}{yield\ s\ f : res}$$

and adjust the evaluation rules.

Conclusions

- With a careful design, giving big-step semantics for languages for concurrency is entirely possible, also taking into account infinite behaviors, incl. divergence.
- Depending on the purpose at hand, there are some choices to be made: resumptions vs traces, big-step vs giant-step, what the appropriate notion of observational equivalence (weak bisimilarity is).
- Concurrency is no more complicated than, say, interactive I/O.