Proofs using SOS: support tooling

work by John Hughes & Cliff Jones

Picture

- VDL -> VDM -> (S)OS
  - Plotkin rules
- concepts
- tooling
  - what we can do
  - what we want to do

The production arrow is just a(n infix) relation

$$\xrightarrow{e} : \mathcal{P}\left((Expr \times \Sigma) \times Expr\right)$$
$$\xrightarrow{s} : \mathcal{P}\left((Stmt \times \Sigma) \times \Sigma\right)$$

Some (big step) SOS rules

Assign
$$\frac{(e,\sigma) \xrightarrow{e} v}{(mk\text{-}Assign(id,e),\sigma) \xrightarrow{s} \sigma \dagger \{id \mapsto v\}}$$

If-T
$$\frac{(b,\sigma) \xrightarrow{e} \textbf{true} \quad (th,\sigma) \xrightarrow{s} \sigma'}{(mk\text{-}If(b,th,el),\sigma) \xrightarrow{s} \sigma'}$$

If-F
$$\frac{(b,\sigma) \xrightarrow{e} \textbf{false} \quad (el,\sigma) \xrightarrow{s} \sigma'}{(mk\text{-}If(b,th,el),\sigma) \xrightarrow{s} \sigma'}$$

A program

```
{true}
if x < 0
then r := -x
else r := x
{r >= 0}
```

Example Proof

	from $(mk\text{-}If(x < 0, r \leftarrow \text{-}x, r \leftarrow x), \sigma_0) \xrightarrow{s} \sigma_f$	
1.	$x < 0 \in \mathbb{B}$	wf-Expr
2.	$(x < 0, \sigma_0) \xrightarrow{e}$ **true** $\vee$ $(x < 0, \sigma_0) \xrightarrow{e}$ **false**	1, $\xrightarrow{e}$
3.	**from** $(x < 0, \sigma_0) \xrightarrow{e}$ **true**	
3.1	$\text{-}\sigma_0(x) \geq 0$	h3
3.2	$(r \leftarrow \text{-}x, \sigma_0) \xrightarrow{e} \sigma_0 \dagger \{r \mapsto \text{-}\sigma_0(x)\}$	Assign
3.3	$(mk\text{-}If(x < 0, r \leftarrow \text{-}x, r \leftarrow x), \sigma_0) \xrightarrow{s} \sigma_0 \dagger \{r \mapsto \text{-}\sigma_0(x)\}$	if-T(h3, 3.2)
3.4	$\sigma_f = \sigma_0 \dagger \{r \mapsto \text{-}\sigma_0(x)\}$	h, 3.3
	infer $\sigma_f(r) \geq 0$	3.4, 3.1
4.	**from** $(x < 0, \sigma_0) \xrightarrow{e}$ **false**	
4.1	$\sigma_0(x) \geq 0$	h4
4.2	$(r \leftarrow x, \sigma_0) \xrightarrow{e} \sigma_0 \dagger \{r \mapsto \sigma_0(x)\}$	Assign
4.3	$(mk\text{-}If(x < 0, r \leftarrow \text{-}x, r \leftarrow x), \sigma_0) \xrightarrow{s} \sigma_0 \dagger \{r \mapsto \sigma_0(x)\}$	if-F(h4, 4.2)
4.4	$\sigma_f = \sigma_0 \dagger \{r \mapsto \sigma_0(x)\}$	h, 4.3
	infer $\sigma_f(r) \geq 0$	4.4, 4.1
	infer $\sigma_f(r) \geq 0$	$\vee$-E(2, 3, 4)

Non-determinism

- some non-deterministic constructs
  - e.g. guarded **if** / **while**
- nasty issues like order of expression evaluation
- essential for concurrency
- more than one rule “matches”
  - the key to “Plotkin rules”
  - shifts non-determinacy to meta level!
- -> really is a relation (not a function)
  - over configurations ($\Sigma \times$ Text)

Work with Joey Coleman

- concurrent language
  - fine-grained semantics
  - nested parallel construct
- our purpose is to justify rely/guarantee rules
  - have a *structural* proof
  - total correctness (complete induction)
  - the key lemma gives insight into expressability
- the R/G rules then viewed as “proof tactics”

Tool support (have)

- obvious how to use VDM ToolSet
  - for semantic *function*
    - McCarthy abstract interpreter
  - abstract objects etc. just perfect ☺
  - we used in teaching (see FM-Ed-2006 paper)
- there is a way to trick the ToolSet to be useful for relations
  - in-meaning: Config x Config -> Bool
  - performance is, um, not good

Tool support (would like)

- we'd like a *mural*-like interface
  - best thing since before sliced-bread
  - plus RODIN-like background search for proof
- insertion of facts about -> as extensions to logical frame
- fixes like name binding boxes