

Definitive Semantic Descriptions

Peter D. Mosses

BRICS & Department of Computer Science

University of Aarhus, Denmark

1st APPSEM-II Workshop, Nottingham, March 2003

Conventional semantic descriptions

Abstract syntax (fragment)

Expressions $e \in Exp$

$e ::= con \mid x \mid e_0 \text{ bop } e_1 \mid \sim e \mid \dots$

Commands $c \in Com$

$c ::= x := e \mid c_0 ; c_1 \mid \mathbf{if } e \mathbf{ then } c \mid \dots$

...

Conventional semantic descriptions

Auxiliary entities (fragment)

Environments $\rho \in Env = Var \rightarrow BV$

Stores $\sigma \in S = Loc \rightarrow SV \dots$

...

Conventional semantic descriptions

Semantics (SOS fragment)

Expressions

$$\rho \vdash \langle e, \sigma \rangle \longrightarrow \langle e', \sigma' \rangle$$

$$\frac{\rho(x) = l, \quad \sigma(l) = v}{\rho \vdash \langle x, \sigma \rangle \longrightarrow \langle v, \sigma \rangle} \quad (1)$$

Conventional semantic descriptions

Semantics (SOS fragment)

Commands

$$\rho \vdash \langle c, \sigma \rangle \longrightarrow \langle c', \sigma' \rangle$$

$$\frac{\rho \vdash \langle e, \sigma \rangle \longrightarrow \langle e', \sigma' \rangle}{\rho \vdash \langle \mathbf{if } e \mathbf{ then } c, \sigma \rangle \longrightarrow \langle \mathbf{if } e' \mathbf{ then } c, \sigma' \rangle} \quad (2)$$

$$\rho \vdash \langle \mathbf{if true then } c, \sigma \rangle \longrightarrow \langle c, \sigma \rangle \quad (3)$$

$$\rho \vdash \langle \mathbf{if false then } c, \sigma \rangle \longrightarrow \langle \mathbf{nil}, \sigma \rangle \quad (4)$$

Possibility of reuse of parts of descriptions?

- usually cut-and-paste, edit, . . .
- explicit modules don't help much . . .

Best chance for reuse with descriptions of

individual constructs

(or of a few closely-related constructs)

Conventional descriptions of constructs

Commands

$$c \in Com$$
$$\rho \in Env, \sigma \in S, \dots$$
$$\rho \vdash \langle c, \sigma \rangle \longrightarrow \langle c', \sigma' \rangle$$

Conventional descriptions of constructs

Commands: Conditional

$c ::= \text{if } e \text{ then } c$

$V \supseteq \{\text{true, false}\}$

$$\frac{\rho \vdash \langle e, \sigma \rangle \longrightarrow \langle e', \sigma' \rangle}{\rho \vdash \langle \text{if } e \text{ then } c, \sigma \rangle \longrightarrow \langle \text{if } e' \text{ then } c, \sigma' \rangle} \quad (5)$$

...

Possibility of reuse of parts of descriptions!

- a **language** description is the collection of the descriptions of its **individual constructs**
- need to develop **libraries** of descriptions of **individual constructs** and auxiliary entities

Unfortunately, there's a major problem:

combining constructs sometimes requires **reformulation** of their descriptions

We need **definitive** descriptions of **constructs!**

- conventional SOS and denotational semantics don't support definitive descriptions
- **modular SOS** [see the proceedings] and **action semantics** definitely do
- does **monadic** denotational semantics?

Definitive descriptions of constructs

Commands

$$c \in Com$$

$$c \xrightarrow{X} c'$$

$$Final \supseteq \{\mathbf{nil}\}$$

Definitive descriptions of constructs

Commands: Conditional

$c ::= \text{if } e \text{ then } c$

$V \supseteq \{\text{true, false}\}$

$$\frac{e \xrightarrow{X} e'}{\text{if } e \text{ then } c \xrightarrow{X} \text{if } e' \text{ then } c}$$

...

(6)

Definitive descriptions of constructs

Expressions

$$e \in Exp$$

$$e \xrightarrow{X} e'$$

$$Final \supseteq Con$$

Definitive descriptions of constructs

Expressions: Constant Identifier

$$e ::= x$$
$$\rho : Env$$

$$\frac{U = \{\rho, \dots\}, \quad \rho(x) = con}{x \xrightarrow{U} con} \quad (7)$$

Status

- Libraries of definitive descriptions of constructs (and auxiliary entities) are being developed for **MSOS** and **action semantics**
- A language-independent **abstract syntax** is being developed
- **Bisimulation proofs** can be language-independent too, based on the definitive descriptions of the constructs involved

Conclusion

- Describe individual constructs **definitively**
- **Contribute** to libraries
- **Refer** to libraries