Operational semantics for procedures

Hanne Riis Nielson

Informatics and Mathematical Modelling
Technical University of Denmark

Blocks and procedures

$$
\begin{array}{rcl}
S & ::= & x := a \mid \texttt{skip} \mid S_1 ; S_2 \mid \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2 \\
 & \mid & \texttt{while}\ b\ \texttt{do}\ S \\
 & \mid & \texttt{begin}\ D_V\ D_P\ S\ \texttt{end} \mid \texttt{call}\ p \\
D_V & ::= & \texttt{var}\ x := a;\ D_V \mid \varepsilon \\
D_P & ::= & \texttt{proc}\ p\ \texttt{is}\ S;\ D_P \mid \varepsilon
\end{array}
$$

How is the semantics extended?

Semantics of blocks

- transition system for statements $\langle S, s\rangle \rightarrow s'$

$$\frac{\langle D_V, s\rangle \rightarrow_D s' \quad \langle S, s'\rangle \rightarrow s''}{\langle \texttt{begin}\ D_V\ S\ \texttt{end}, s\rangle \rightarrow s''[DV(D_V) \longmapsto s]}$$

where the defined variables of D_V are given by:

$$\begin{array}{rcl} DV(\texttt{var}\ x := a\ ;D_V) & = & \{x\} \cup DV(D_V) \\ DV(\varepsilon) & = & \emptyset \end{array}$$

- transition system for variable declarations $\langle D_V, s\rangle \rightarrow_D s'$

$$\frac{\langle D_V, s[x \mapsto \mathcal{A}[\![a]\!]s]\rangle \rightarrow_D s'}{\langle \texttt{var}\ x := a\ ;D_V, s\rangle \rightarrow_D s'} \qquad \langle \varepsilon, s\rangle \rightarrow_D s$$

The semantics of procedures

- dynamic scope for variables as well as procedures,
- dynamic scope for variables but static scope for procedures, and
- static scope for variables as well as procedures.

```
begin var x := 0;
      proc p is x := x ⋆ 2;
      proc q is call p;
      begin var x := 5;
            proc p is x := x + 1;
            call q; y := x
      end
end
```

Environments

- dynamic scope rules

$$env_P \quad \in \quad \mathbf{Env_P} \quad = \quad \mathbf{PName} \hookrightarrow \mathbf{Stm}$$

- static scope rules for variables

$$\begin{array}{rcccl} env_V & \in & \mathbf{Env_V} & = & \mathbf{Var} \hookrightarrow \mathbf{Loc} \\ sto & \in & \mathbf{Store} & = & \mathbf{Loc} \cup \{\mathsf{next}\} \to \mathbf{Z} \end{array}$$

- static scope rules for procedures

$$env_P \quad \in \quad \mathbf{Env_P} \quad = \quad \mathbf{PName} \hookrightarrow \mathbf{Stm} \times \mathbf{Env_V} \times \mathbf{Env_P}$$

Dynamic scope rules: $env_P \vdash \langle S, s\rangle \to s'$

$$env_P \vdash \langle x := a, s\rangle \to s[x \mapsto \mathcal{A}[\![a]\!]s]$$

$$env_P \vdash \langle \texttt{skip}, s\rangle \to s$$

$$\frac{env_P \vdash \langle S_1, s\rangle \to s' \quad env_P \vdash \langle S_2, s'\rangle \to s''}{env_P \vdash \langle S_1; S_2, s\rangle \to s''}$$

$$\frac{env_P \vdash \langle S_1, s\rangle \to s'}{env_P \vdash \langle \texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2, s\rangle \to s'} \qquad \text{if } \mathcal{B}[\![b]\!]s = \mathbf{tt}$$

$$\frac{env_P \vdash \langle S_2, s\rangle \to s'}{env_P \vdash \langle \texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2, s\rangle \to s'} \qquad \text{if } \mathcal{B}[\![b]\!]s = \mathbf{ff}$$

Dynamic scope rules (2)

$$\frac{env_P \vdash \langle S, s\rangle \to s' \quad env_P \vdash \langle \texttt{while}\ b\ \texttt{do}\ S, s'\rangle \to s''}{env_P \vdash \langle \texttt{while}\ b\ \texttt{do}\ S, s\rangle \to s''} \quad \text{if } \mathcal{B}[\![b]\!]s = \mathbf{tt}$$

$$env_P \vdash \langle \texttt{while}\ b\ \texttt{do}\ S, s\rangle \to s \quad \text{if } \mathcal{B}[\![b]\!]s = \mathbf{ff}$$

$$\frac{\langle D_V, s\rangle \to_D s' \quad \mathsf{upd}_P(D_P, env_P) \vdash \langle S, s'\rangle \to s''}{env_P \vdash \langle \texttt{begin}\ D_V\ D_P\ S\ \texttt{end}, s\rangle \to s''[DV(D_V) \longmapsto s]}$$

$$\frac{env_P \vdash \langle S, s\rangle \to s'}{env_P \vdash \langle \texttt{call}\ p, s\rangle \to s'} \quad \text{where } env_P(p) = S$$

where

$$\begin{array}{rcl} \mathsf{upd}_P(\texttt{proc}\ p\ \texttt{is}\ S; D_P, env_P) & = & \mathsf{upd}_P(D_P, env_P[p \mapsto S]) \\ \mathsf{upd}_P(\varepsilon, env_P) & = & env_P \end{array}$$

Static scope rules: $\langle D_V, env_V, sto\rangle \rightarrow_D (env'_V, sto')$

$$\frac{\langle D_V, env_V[x \mapsto \ell], sto[\ell \mapsto v][next \mapsto \ell']\rangle \rightarrow_D (env'_V, sto')}{\langle \texttt{var}\ x := a; D_V, env_V, sto\rangle \rightarrow_D (env'_V, sto')}$$

where
$$\begin{array}{rcl} v & = & \mathcal{A}[\![a]\!](sto \circ env_V) \\ \ell & = & sto(\mathsf{next}) \\ \ell' & = & new(\ell) \end{array}$$

$$\langle \varepsilon, env_V, sto\rangle \rightarrow_D (env_V, sto)$$

Static scope rules: $env_V, env_P \vdash \langle S, sto\rangle \to sto'$

$$env_V, env_P \vdash \langle x := a, sto\rangle \to sto[\ell \mapsto v] \text{ where } \begin{array}{l} \ell = env_V(x) \\ v = \mathcal{A}[\![a]\!](sto \circ env_V) \end{array}$$

$$env_V, env_P \vdash \langle \texttt{skip}, sto\rangle \to sto$$

$$\frac{env_V, env_P \vdash \langle S_1, sto\rangle \to sto' \quad env_V, env_P \vdash \langle S_2, sto'\rangle \to sto''}{env_V, env_P \vdash \langle S_1; S_2, sto\rangle \to sto''}$$

$$\frac{env_V, env_P \vdash \langle S_1, sto\rangle \to sto'}{env_V, env_P \vdash \langle \texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2, sto\rangle \to sto'} \quad \text{if } \mathcal{B}[\![b]\!](sto \circ env_V) = \mathbf{tt}$$

$$\frac{env_V, env_P \vdash \langle S_2, sto\rangle \to sto'}{env_V, env_P \vdash \langle \texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2, sto\rangle \to sto'} \quad \text{if } \mathcal{B}[\![b]\!](sto \circ env_V) = \mathbf{ff}$$

Static scope rules (2)

$$\frac{env_V, env_P \vdash \langle S, sto\rangle \rightarrow sto' \quad env_V, env_P \vdash \langle \texttt{while}\ b\ \texttt{do}\ S, sto'\rangle \rightarrow sto''}{env_V, env_P \vdash \langle \texttt{while}\ b\ \texttt{do}\ S, sto\rangle \rightarrow sto''}$$

if $\mathcal{B}[\![b]\!](sto \circ env_V) = \mathbf{tt}$

$$env_V, env_P \vdash \langle \texttt{while}\ b\ \texttt{do}\ S, sto\rangle \rightarrow sto \quad \text{if } \mathcal{B}[\![b]\!](sto \circ env_V) = \mathbf{ff}$$

Static scope rules (3)

$$\frac{\langle D_V, env_V, sto\rangle \to_D (env'_V, sto') \quad env'_V, env'_P \vdash \langle S, sto'\rangle \to sto''}{env_V, env_P \vdash \langle \texttt{begin}\ D_V\ D_P\ S\ \texttt{end}, sto\rangle \to sto''}$$

where $env'_P = \mathsf{upd}_P(D_P, env'_V, env_P)$

$$\frac{env'_V, env'_P \vdash \langle S, sto\rangle \to sto'}{env_V, env_P \vdash \langle \texttt{call}\ p, sto\rangle \to sto'}$$

where $env_P(p) = (S, env'_V, env'_P)$

$$\frac{env'_V, env'_P[p \mapsto (S, env'_V, env'_P)] \vdash \langle S, sto\rangle \to sto'}{env_V, env_P \vdash \langle \texttt{call}\ p, sto\rangle \to sto'}$$

where $env_P(p) = (S, env'_V, env'_P)$