

Le retour des portées des variables en sémantique naturelle

Équipe de Sémantique M1

V4, 12 octobre 2005

1 Portée dynamique variables et procédures

Les règles Config = (env, S, σ) et $env : p \mapsto S$.

$$\frac{(D_V, \sigma) \rightarrow_D \sigma' \quad (\text{upd}(env, D_P), S, \sigma') \rightarrow \sigma''}{(env, \text{begin } D_V D_P; S \text{ end } , \sigma) \rightarrow \sigma'' [\text{DV}(D_V) \mapsto \sigma]}$$

avec :

- $\text{upd}(env, \varepsilon) = env$
- $\text{upd}(env, \text{proc } p \text{ is } S; D_P) = \text{upd}(env[p \mapsto S], D_P)$

$$\frac{(env, env(p), \sigma) \rightarrow \sigma'}{(env, \text{call } p, \sigma) \rightarrow \sigma'}$$

Les autres commandes ont des sémantiques "copiées" sur la sémantique sans bloc : (évidemment, l'environnement n'est jamais modifié dans les autres cas...) :

$$(env, x := a, \sigma) \rightarrow \sigma[x \mapsto \mathcal{A}[a]\sigma];$$

$$\frac{(env, S_1, \sigma) \rightarrow \sigma' \quad (env, S_2, \sigma') \rightarrow \sigma''}{(env, S_1; S_2, \sigma) \rightarrow \sigma''}$$

etc

Exercice Source : http://www.daimi.au.dk/~bra8130/Wiley_book/wiley.html

Construire un arbre de dérivation pour l'exécution du programme suivant, où l'état initial est avec $x = 3$:

```
begin
  var y:=1;                -- DVO
  proc fac is              -- début DPO
    begin                  -- début Sp
      var z:=x;            -- DV1
      if x=1 then skip     -- début S1
        else (x:=x-1;call fac;y:=z*y) -- fin S1
      end;                 -- fin Sp/DPO
    call fac;              -- S0
  end;
```

2 Portée statique procédures

Les règles Config = (env, S, σ) et $env : p \mapsto (S, env)$.

$$\frac{(D_V, \sigma) \rightarrow_D \sigma' \quad (\text{upd}(env, D_P), S, \sigma') \rightarrow \sigma''}{(env, \text{begin } D_V D_P; S \text{ end } , \sigma) \rightarrow \sigma'' [\text{DV}(D_V) \mapsto \sigma]}$$

- $\text{upd}(env, \varepsilon) = env$

- $\text{upd}(\text{env}, \text{proc } p \text{ is } S; D_P) = \text{upd}(\text{env}[p \mapsto (S, \text{env})], D_P)$.
- et l'appel, avec $\text{env}(p) = (S, \text{env}')$:

$$\frac{(\text{env}', S, \sigma) \rightarrow \sigma'}{(\text{env}, \text{call } p, \sigma) \rightarrow \sigma'}$$

ou [rec]

$$\frac{(\text{env}'[p \mapsto [(S, \text{env}')], S, \sigma) \rightarrow \sigma'}{(\text{env}, \text{call } p, \sigma) \rightarrow \sigma'}$$

3 Portée statique procédures ET variables

Règles

- Config = $(\text{env}_V, \text{env}_P, S, \text{sto})$ ou $(\text{env}_V, \text{sto})$ avec :
- $\text{env}_V : x \mapsto \text{adresse}$
 - $\text{env}_P : p \mapsto (\text{env}_V, \text{env}_P, S)$ pour garder en mémoire ce que valent les variables au moment de la déclaration.
 - $\text{sto} : \text{adresse} \mapsto \mathbb{Z}$ (ancien $\sigma(x) = \text{sto}(\text{env}_V(x))$).

Définition des variables (var) :

- $(\varepsilon, \text{env}_V, \text{sto}) \rightarrow_D (\text{env}_V, \text{sto})$
- $\frac{(D_V, \text{env}_V[x \mapsto \text{nc}], \text{sto}[\text{nc} \mapsto v]) \rightarrow_D (\text{env}'_V, \text{sto}')}{((\text{var } x := a; D_V), \text{env}_V, \text{sto}) \rightarrow_D (\text{env}'_V, \text{sto}')}$
avec $v = \mathcal{A}[a](\text{sto} \circ \text{env}_V)$, $\text{nc} = \text{new}(\text{sto})$ pour que l'on ait : $x \rightarrow \text{nc} \rightarrow v$

Statement Les changements opérés sur les règles sont plus complexes, donc on va toutes les réécrire :

- Assignation :

$$(\text{env}_V, \text{env}_P, x := a, \text{sto}) \rightarrow (\text{env}_V, \text{sto}[\text{nc} \mapsto v])$$

avec $v = \mathcal{A}[a](\text{sto} \circ \text{env}_V)$ et $\text{nc} = \text{env}_V(x)$

- Skip :

$$(\text{env}_V, \text{env}_P, \text{skip}, \text{sto}) \rightarrow (\text{env}_V, \text{sto})$$

- Séquence :

$$\frac{(\text{env}_V, \text{env}_P, S_1, \text{sto}) \rightarrow (\text{env}_V, \text{sto}') \quad (\text{env}_V, \text{env}_P, S_2, \text{sto}') \rightarrow (\text{env}_V, \text{sto}'')}{(\text{env}_V, \text{env}_P, S_1; S_2, \text{sto}) \rightarrow (\text{env}_V, \text{sto}'')}$$

- If-true et If-false et while : similaires

Statement=block

$$\frac{(D_V, \text{env}_V, \text{sto}) \rightarrow_D (\text{env}'_V, \text{sto}') \quad (\text{env}'_V, \text{env}'_P, S, \text{sto}) \rightarrow (\text{env}_V'', \text{sto}'')}{(\text{env}_V, \text{env}_P, \text{begin } D_V D_P; S \text{ end}, \text{sto}) \rightarrow (\text{env}_V, \text{sto}'')}$$

avec $\text{env}'_P = \text{upd}(\text{env}'_V, \text{env}_P, D_P)$. La fonction upd est la même que la précédente, sauf que l'environnement des procédures est un triplet qui contient l'environnement des variables à la définition.

et l'appel avec $\text{env}_P(p) = (\text{env}'_V, \text{env}'_P, S)$:

$$\frac{(\text{env}'_V, \text{env}'_P, S, \text{sto}) \rightarrow (\text{env}'_V, \text{sto}')}{(\text{env}_V, \text{env}_P, \text{call } p, \text{sto}) \rightarrow (\text{env}_V, \text{sto}'')}$$

ou [rec]

$$\frac{(\text{env}'_V, \text{env}'_P[p \mapsto (\text{env}'_V, \text{env}'_P, S)], S, \text{sto}) \rightarrow (\text{env}'_V, \text{sto}')}{(\text{env}_V, \text{env}_P, \text{call } p, \text{sto}) \rightarrow (\text{env}_V, \text{sto}'')}$$

Exercice Le même ...