Sémantique du langage Oz

Rappels et résolution de la question 3 de l'examen 2004

D. Saucez A. Tang Mac

15 août 2005

Plan

Langage noyau[1] = langage exécuté par l'interpréteur du système Mozart.
Permet la compréhension de programmes sans **ambiguité**.

[1] Concepts of computer programming p.56 à 72 et p.415 à 417

Exemple d'« ambiguité » :

```
local X in
    X=1
    local X in
        X=2
        {Browse X}
    end
    {Browse X}
end
```

Quel est le résultat affiché ?
2 puis 1

Autre :

```
local Y LB in
    Y=10
    proc {LB X ?Z}
        if X>=Y then Z=X else Z=Y end
    end
    local Y=15 Z in
        {LB 5 Z}
        {Browse Z}
    end
end
```

Quel est le résultat affiché ?
10

Single-assignment store σ

Ensemble de variables déclaratives[2].

Exemple :
$\{x = 1, y = 237\}$

[2] Convention : on les note en minuscule

Environnement

Ensemble de (identificateurs[3] $\longrightarrow$ variables).

Exemple :
$\{X \rightarrow x\}$

[3] Toujours notées en majuscule

Instruction sémantique (Semantic statement)

paire $(\langle s \rangle, E)$ où

- $\langle s \rangle$ est une instruction
- E est un environnement

Exemple :

$\left(\texttt{\{Browse X\}}, \{X \rightarrow x\}\right)$

Etat d'exécution (Execution state)

$$\left(\left[\texttt{instruction sémantique}\right], \sigma\right)$$
$$= \left(\left[(\langle s\rangle_1, E_1), (\langle s\rangle_2, E_2), \ldots\right], \sigma\right)$$

Exemple :

$$\left(\left[\left(\{\texttt{Browse X}\}, \{\mathit{Browse} \rightarrow b, X \rightarrow x\}\right)\right],$$
$$\{(b = \texttt{proc...end}, CE_1), x = 1\}\right)$$

Récapitulatif

$$\Big(\big[\underbrace{\underbrace{(\ \underbrace{\langle s \rangle_1}_{\text{instruction}}\ ,\ \underbrace{E_1}_{\text{environnement}}\ )}_{\text{instruction sémantique}}, (\langle s \rangle_2, E_2), \ldots \big]}_{\text{pile sémantique}}, \underbrace{\sigma}_{\text{store}} \Big)$$

état d'exécution

Transformation en langage noyau

```
{Browse 7}
```

```
local X in
    X=7
    {Browse X}
end
```

```
local T X in
    T=1 X=2
    if T<X then
        {Browse X}
    end
end
```

```
local T X Z in
    T=1 X=2 Z=T<X
    if Z then
        {Browse X}
    end
end
```

```
declare
fun {Test A}
   1+A
end
```

```
declare Test in
proc {Test A R}
   R=1+A
end
```

```
declare
Test=proc {$ A R}
   R=1+A
end
```

```
declare
fun {Add X Y}
   X+Y
end
{Browse {Add 2 7}}
```

```
declare
Add=proc {$ X Y ?Z}
        Z=X+Y
     end
local A B C in
   A=2 B=7
   {Add A B C}
   {Browse C}
end
```

```
declare
fun {Fact N}
   if N=<0 then 1
   else N*{Fact N-1}
   end
end
```

```
declare
Fact=proc {$ N ?Z}
   Tmp in
   Tmp=N=<0
   if Tmp then Z=1
   else
      local R in
         {Fact N-1 R}
         Z=N*R
      end
   end
end
```

```
declare
fun {App L1 L2}
   case L1
   of H|T then
      H|{App T L2}
   [] nil then L2
   end
end
```

```
declare App in
proc {App L1 L2 ?Z}
   case L1
   of H|T then
      local Tmp in
         Z=H|Tmp
         {App T L2 Tmp}
      end
   [] nil then Z=L2
   end
end
```

Début exécution

$$\left(\left[(\langle s \rangle, \varnothing)\right], \varnothing\right)$$

- $\langle s \rangle$ est un bloc d'instructions
- l'environnement est vide
- le store est vide (rajoutez les procédures existant dès le démarrage de Mozart et qui vont être utilisées)

Composition séquentielle

$$\left[(\langle s\rangle_1\ \langle s\rangle_2, E)\right] \Longrightarrow \left[(\langle s\rangle_1, E), (\langle s\rangle_2, E)\right]$$

Exemple :

$$\left[(\texttt{X=2 \{Browse X\} Y=1}, \{\mathit{Browse} \rightarrow b, X \rightarrow x, Y \rightarrow y\})\right]$$
$$\Longrightarrow$$

$$\left[(\texttt{X=2}, \{\mathit{Browse} \rightarrow b, X \rightarrow x, Y \rightarrow y\}),\right.$$
$$\left.(\texttt{\{Browse X\} Y=1}, \{\mathit{Browse} \rightarrow b, X \rightarrow x, Y \rightarrow y\})\right]$$

Déclaration de variable

$$\Big(\Big[(\texttt{local X in X=2 \{Browse X\} end}, \{\mathit{Browse} \rightarrow b\})\Big], \{(b = \texttt{proc ...end}, \mathit{CE}_1)\}\Big)$$
$$\Longrightarrow$$
$$\Big(\Big[(\texttt{X=2 \{Browse X\}}, \{\mathit{Browse} \rightarrow b, X \rightarrow x\})\Big], \{(b = \texttt{proc...end}, \mathit{CE}_1), x\}\Big)$$

Déclaration de procédure

$$\langle s \rangle \equiv \left\{ \begin{array}{l} \texttt{local X P in} \\ \quad \texttt{X=237} \\ \quad \texttt{P=proc\{\$ A Z\} Z=A+X end} \\ \texttt{end} \end{array} \right\} \equiv \langle s \rangle_2$$

1. $\left(\left[(\langle s \rangle, \varnothing) \right], \varnothing \right)$
2. $\left(\left[(\langle s \rangle_2, \{X \rightarrow x, P \rightarrow p\}) \right], \{x, p\} \right)$
3. $\Big(\big[\big], \{x = 237,$
   $(p = \texttt{proc\{\$ A Z\} Z=A+X end}, \{X \rightarrow x\})\} \Big)$

Procédure : environnement contextuel

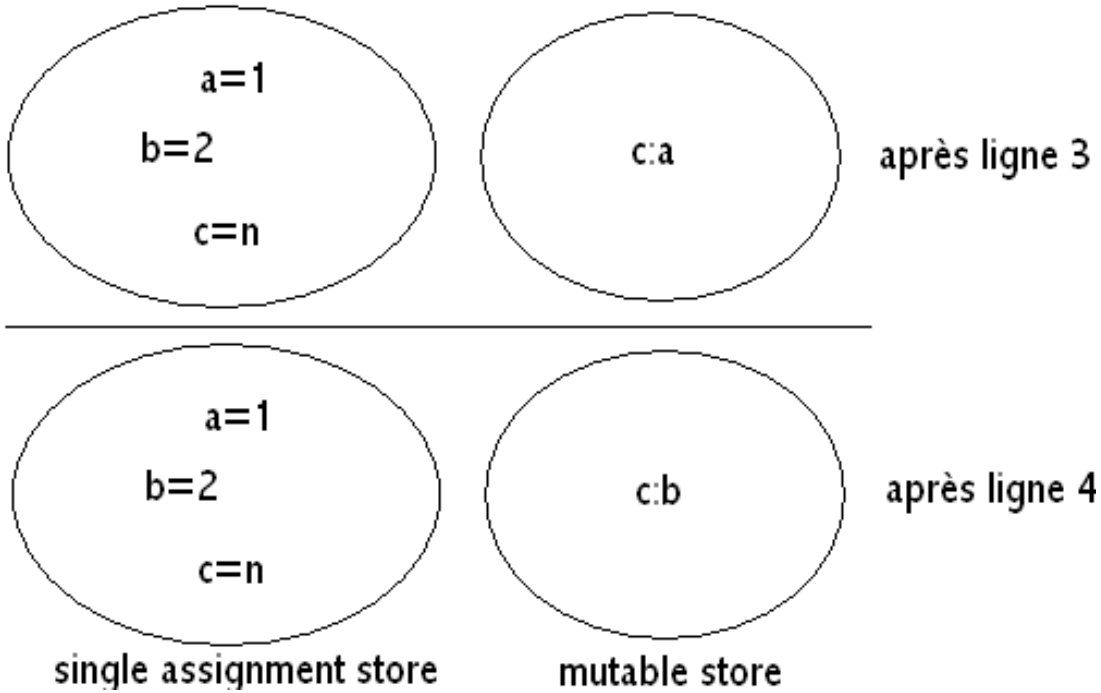
```
declare Concat Append in
proc {Append L1 L2 Z} ...end
proc {Concat Ls ?Z}
   case Ls
   of L|Lr then
      local Temp in
         {Concat Lr Temp}
         {Append L Temp Z}
      end
   [] nil then Z=nil
   end
end
```

$$\sigma = \{(a = \texttt{proc } \ldots \texttt{end}, \{\mathit{Append} \rightarrow a\}), (c = \texttt{proc } \ldots \texttt{end}, \{\mathit{Concat} \rightarrow c, \mathit{Append} \rightarrow a\})\}$$

Cellules

Cellule=(nom, référence vers une variable $\in \sigma$)

```
1 declare A B C in
2 A=1 B=2
3 {NewCell A C}
4 C:=B
```



```
1 declare A B C in
2 A=1 B=2
3 {NewCell A C}
4 C:=B
```

1. $\left(\left[(\langle s \rangle, \{\mathit{NewCell} \rightarrow w\})\right], \{(w = \texttt{proc end}, \mathit{CE}_1)\}, \varnothing\right)$
2. $\left(\left[(2\texttt{-}3\texttt{-}4, \{\mathit{NewCell} \rightarrow w, A \rightarrow a, B \rightarrow b, C \rightarrow c\})\right], \{(w = \texttt{proc end}, \mathit{CE}_1), a, b, c\}, \varnothing\right)$
3. $\left(\left[(4, \{\mathit{NewCell} \rightarrow w, A \rightarrow a, B \rightarrow b, C \rightarrow c\})\right], \underbrace{\{(w = \texttt{proc end}, \mathit{CE}_1), a = 1, b = 2, c = n\}}_{\sigma}, \underbrace{\{c : a\}}_{\mu}\right)$
4. $\left(\left[\right], \{(w = \texttt{proc end}, \mathit{CE}_1), a = 1, b = 2, c = n\}, \{c : b\}\right)$

Code source

```
1  declare
2  fun {NewCounter}
3     A1={NewCell 0}
4     proc {Inc} A1 := @A1+1 end
5     proc {Get X} X=@A1 end
6  in
7     proc {$ M}
8        case M of inc then {Inc}
9        [] get(X) then {Get X}
10       end
11    end
12 end
13 C={NewCounter}
14 {C inc}
15 local X in {C get(X)} {Browse X} end
```

```
declare NewCounter C in
proc {NewCounter R}
   local A1 Inc Get in
      local I in
         I=0
         {NewCell I A1}
      end
      proc {Inc} A1:=@A1+1 end
      proc {Get X} X=@A1 end
      proc {R M}
         case M
         of inc then {Inc}
         [] get(X) then {Get X}
         end
      end
   end
end
```

```
{NewCounter C}
local Op1 in
   Op1=inc
   {C Op1}
end
local X Op2 in
   Op2=get(X)
   {C Op2}
   {Browse X}
end
```

Soient $K = \{Browse \rightarrow b, NewCell \rightarrow w\}$ et
$k = \{(b = \texttt{proc} \ \ldots \texttt{end}, CE_1), (w = \texttt{proc} \ \ldots \texttt{end}, CE_2)\}$

1. $\left(\left[(\langle s \rangle, K)\right], k, \varnothing\right)$
2. $\left(\left[(\texttt{2-27}, K \cup \{NewCounter \rightarrow nc, C \rightarrow c\})\right], k \cup \{nc, c\}, \varnothing\right)$
3. $\left(\left[(\texttt{18-27}, K \cup \{NewCounter \rightarrow nc, C \rightarrow c\})\right], k \cup \{(nc = \texttt{proc} \ \ldots \texttt{end}, \{NewCell \rightarrow w\}), c\}, \varnothing\right)$
4. $\left(\left[(\texttt{3-16}, \{NewCell \rightarrow w, R \rightarrow c\}), (\texttt{19-27}, K \cup \{NewCounter \rightarrow nc, C \rightarrow c\})\right], k \cup \{(nc = \texttt{proc} \ \ldots \texttt{end}, \{NewCell \rightarrow w\}), c\}, \varnothing\right)$

5. $\Big(\Big[(\texttt{4-15}, K \cup \{\mathit{NewCell} \rightarrow w, R \rightarrow c, A1 \rightarrow a1, \mathit{Inc} \rightarrow \mathit{incp}, \mathit{Get} \rightarrow \mathit{getp}\}),$
$(\texttt{19-27}, K \cup \{\mathit{NewCounter} \rightarrow \mathit{nc}, C \rightarrow c\})\Big],$
$k \cup \{(\mathit{nc} = \texttt{proc} \ \ldots \texttt{end}, \{\mathit{NewCell} \rightarrow w\}),$
$c, a1, \mathit{incp}, \mathit{getp}\}, \varnothing\Big)$

6. $\Big(\Big[(\texttt{19-27}, K \cup \{\mathit{NewCounter} \rightarrow \mathit{nc}, C \rightarrow c\})\Big],$
$k \cup \{(\mathit{nc} = \texttt{proc} \ \ldots \texttt{end}, \{\mathit{NewCell} \rightarrow w\}),$
$(c = \texttt{proc} \ \ldots \texttt{end}, \{\mathit{Inc} \rightarrow \mathit{incp}, \mathit{Get} \rightarrow \mathit{getp}\}), a1 = n,$
$(\mathit{incp} = \texttt{proc} \ \ldots \texttt{end}, \{A1 \rightarrow a1\}),$
$(\mathit{getp} = \texttt{proc} \ldots \texttt{end}, \{A1 \rightarrow a1\}), i0 = 0\}, \{a1 : i0\}\Big)$

6. $\Big(\Big[(\texttt{19-27}, K \cup \{NewCounter \rightarrow nc, C \rightarrow c\})\Big], \sigma_1, \{a1 : i0\}\Big)$
7. $\Big(\Big[(\texttt{11-14}, Inc \rightarrow incp, Get \rightarrow getp, M \rightarrow op1, (\texttt{23-27}, K \cup \{NewCounter \rightarrow nc, C \rightarrow c\})\Big], \sigma_1 \cup \{op1 = inc\}, \{a1 : i0\}\Big)$
8. $\Big(\Big[(\texttt{23-27}, K \cup \{NewCounter \rightarrow nc, C \rightarrow c\})\Big], \sigma_1 \cup \{op1 = inc, i1 = 1\}, \{a1 : i1\}\Big)$

9. $\Big(\Big[(25\text{-}26, K \cup \{NewCounter \rightarrow nc, C \rightarrow c, X \rightarrow x, Op2 \rightarrow op2\})\Big], \sigma_1 \cup \{op1 = inc, i1 = 1, x, op2 = get(x)\}, \{a1 : i1\}\Big)$

10. $\Big(\Big[(11\text{-}14, Inc \rightarrow incp, Get \rightarrow getp, M \rightarrow op2\}), (26, K \cup \{NewCounter \rightarrow nc, C \rightarrow c, X \rightarrow x, Op2 \rightarrow op2\})\Big], \sigma_1 \cup \{op1 = inc, i1 = 1, x, op2 = get(x)\}, \{a1 : i1\}\Big)$

11. $\Big([], \sigma_1 \cup \{op1 = inc, i1 = 1, x = i1, op2 = get(x)\}, \{a1 : i1\}\Big)$