Université Bordeaux 1
Master d'Informatique 1, 2010/2011

Modèles de Calcul : corrigé du CC 2

Modèles alternatifs aux machines de Turing

Exercice 2.1 [8 points]
Construisons précisément une machine à 4 compteurs telle que : partant de la configuration $(0, n, m, 0)$ la machine s'arrête dans la configuration :

$$(1, n, m, 0) \text{ si } n \leq m, \quad (2, n, m, 0) \text{ si } n > m.$$

Idée :

```
save r1 in r3, save r2 in r4;
while (r1 > 0  and r2 > 0)
    decrement r1; decrement r2;
if r1=0 then return 1 else  return 2.
```

Machine à 4 compteurs avec copie de registres et remise a zero :

```
0: r3 ← r1; goto 100
1: stop
2: stop

100: r4 ← r2; goto 110
110: if r1=0 thengoto 120 elsegoto 200
120: #restore r1,r2
     r1 ← r3;goto 130
130: r2 ← r4;goto 140
140: #reset r3, reset r4, return(1)
     r3 ← 0;goto 150
150: r4 ← 0;goto 1

200: if r2 = 0 thengoto 210 elsegoto 300
210: #restore r1, r2
     r1 ← r3;goto 220
220: r2 ← r4;goto 230
230: #reset r3, reset r4, return(2)
     r3 ← 0;goto 240
240: r4 ← 0;goto 2

300: #r1>0 , r2>0
    dec(r1); goto 310
310: dec(r2); goto 110
```

On peut traduire chaque instruction de copie de registre $r_i \leftarrow r_j$ par :

```
N+1: if ri=0 thengoto N+3 elsegoto N+2
N+2: dec(ri); goto N+1
N+3: #ri=0 at first execution; then ri + rj is invariant
     if rj=0 then goto N+6 else goto N+4
N+4  inc(ri); goto N+5
N+5: dec(rj); goto N+3
N+6:
```

On peut traduire chaque instruction de mise à zéro du registre r_i par :

```
N+1: if ri=0 thengoto N+3 elsegoto N+2
N+2: dec(ri); goto N+1
N+3:
```

Exercice 2.2 [4 points]
Le problème suivant (PB de l'arrêt des machines à 2 compteurs) est-il décidable ?
Donnée : une machine à deux compteurs M.
Question : le calcul de M partant de la configuration $(0, 0, 0)$ se termine-t-il ?
Soit M une machine de Turing déterministe (à une bande, infinie vers la droite). On peut construire une machine à 2 compteurs C qui simule la machine M (le principe a été décrit en TD). En particulier, le calcul de M partant de l'état initial q_0 de M et d'une bande vide est infini si et seulement si le calcul de C partant de $(0, 0, 0)$ est infini. Cette construction est une *réduction* du problème de l'arrêt des machines de Turing (déterministes à une bande) sur la bande vide au problème de l'arrêt des machines à deux compteurs. Comme le problème de l'arrêt des machines de Turing sur la bande vide est indécidable (vu en cours), le PB de l'arrêt des machines à 2 compteurs est indécidable.
Exercice 2.3 [22 points]
1-Idée du calcul de P_2 :
Placer n dans r1
Placer dans r2 les puissances de 2 successives jusqu'à ce que $r2 > r1$
Retourner $r2/2$ dans le registre 2.
Machine à 4 compteurs :

```
0: if r1 = 0 thengoto 100 elsegoto 1
1: inc(r2); goto 2
2: if r2 <= r1 then goto 3 elsegoto 4
3: r2 ← r2 + r2; goto 2
4: #r2 > r1 and r2/2 <= r1
   r2 ← r2 div 2; goto 101
100: r2 ← r1+r1; goto 101
101: stop
```

2- Soit $w = x_{\ell-1} \cdots x_i \cdots x_0$. Par définition $\rho(w) = \sum_{i=0}^{\ell-1} x_i * 2^i$.

$$\begin{aligned} \rho(w \cdot x) &= \sum_{i=0}^{\ell-1} x_i * 2^{i+1} + x * 2^0 \\ &= 2 * \rho(w) + x. \end{aligned}$$

3- Idée du calcul de d :
pour un entier $n \neq 0$: d(n) = if (n est pair) then 2 else 1.
Machine à 4 compteurs :

```
0: r2 ← r1 mod 2; goto 1
1: if r2=0 thengoto 2 elsegoto 4
2: #x0=2
   inc(r2); goto 3
3: inc(r2); goto 100
4: #x0=1
   inc(r2); goto 100
100: stop
```

4- Idée du calcul :
Soit r l'entier $\rho(x_n x_{n-1} \cdots x_1 x_0)$. Lorsque $r \neq 0$ on calcule :
if $d(r) = 2$ then $r1 \leftarrow (r-2)$div2 else $r1 \leftarrow (r-1)$div2.
Machine à 4 compteurs :

```
0: if r1=0 thengoto 100 elsegoto 1
1: r2 ← r1 mod 2; goto 2
2: if r2 = 0 thengoto 3 elsegoto 10

3: #x0=2
   dec(r1); goto 4
4: dec(r1); goto 5
5: r1 ← r1 div 2; goto 100

10: #x0=1
    dec(r1); goto 11
11: r1 ← r1 div 2; goto 100

100: stop
```

5- Soit $w \in \{1,2\}^*$: $w = x_{\ell-1} \cdots x_i \cdots x_0, \;\; \ell = |\rho(w)|$.
Comme chaque x_i est compris entre 1 et 2 :

$$\sum_{i=0}^{\ell-1} 2^i \leq \sum_{i=0}^{\ell-1} x_i * 2^i \leq \sum_{i=0}^{\ell-1} x_i * 2^{i+1}$$

donc

$$2^\ell - 1 \leq \rho(w) \leq 2^{\ell+1} - 1$$

ce qui est équivalent à

$$2^{|\rho(w)|} - 1 \leq \rho(w) < 2^{|\rho(w)|+1} - 1.$$

6- Idée directrice :
Au départ : $r1 = x, r2 = \rho(y_m y_{m-1} \cdots y_1 y_0) = \rho(w), r3 = 0, r4 = 0$.
A l'arrivée : $r3 = \rho(x y_m y_{m-1} \cdots y_1 y_0) = \rho(w) + x * 2^{|\rho(w)|}$.
Calcul :
On affecte à r3 l'entier $P_2(r2+1)$ [en appliquant la machine de la qu. 1 à (r2,r3)]
$r3 \leftarrow r2 + (r1 * r3)$. Machine à 4 compteurs :

```
0: inc(r2); goto 1
1: inc(r3); goto 2
2: if r3 <= r2 then goto 3 elsegoto 4
3: r3 ← r3 + r3; goto 2
4: #r3 > r2 and r3/2 <= r2
   r3 ← r3 div 2; goto 101

101: # r3= P_2(r2+1)
     dec(r1)
102: if r1=0 thengoto 103 elsegoto 104

103: #x=1
     r3 ← r2 + r3; goto 200

104: #x=2
     r3 ← r3 + r3; goto 105
105: r3 ← r2 + r3; goto 200

200: stop
```

7- Description :
Au départ : $r1 = \rho(u), r2 = \rho(v)$ où u, v sont des mots sur $\{1, 2\}$.
Au l'arrivée : $r1 = \rho(u \cdot (d(u))^{-1}), r2 = \rho(d(u) \cdot v)$.

```
0: if r1=0 thengoto 1000 else goto 1
1: r3 ← d(u)                 [machine de la q.3 appliquee a (r1,r3)]]

100: r4 ← d(u) ·v            [machine de la q.6 appliquee a (r3,r2,r4)]

200: r2 ← u · d(u)^−1          [machine de la q.4 appliquee a (r2,r3)]

300: r3 ← 0

400: r1 ← r2 + r3

500: r2 ← r4 + r3; goto 1000

1000: stop
```

8- Description :
Au départ : $r1 = \rho(u), r2 = \rho(v)$ où u, v sont des mots sur $\{1, 2\}$.
À l'arrivée : $r1 = \rho(\varepsilon), r2 = \rho(u \cdot v)$.

```
0: if r1=0 thengoto 2000 else goto 1

1:  programme de la q. 7 (sauf lignes 0 et 1000)
```

```
...:programme de la q. 7 (sauf lignes 0 et 1000)

1000: goto 0

2000: stop
```