

FONDEMENTS DE LA PROGRAMMATION : TD 1

MACHINES DE TURING AVEC INSTRUCTIONS (CORRECTION)

Paulin de Naurois et Virgile Mogbil et Lê Thành Dũng Nguyễn
Institut Galilée – Master 1 Informatique

19/09/2018

Exercice 6. Décrire une MTI pour la fonction successeur ($x \mapsto x + 1$) sur une représentation binaire des entiers.

Correction. L'idée est la suivante :

1. on déplace la tête tout à droite ;
2. si le chiffre de poids faible est 0, on écrit 1 et c'est fini ;
3. sinon, il y a une retenue à propager : on écrit 0 ($1 + 1 = (10)_2$: je retiens 1 et je pose 0), on se déplace vers la gauche et on reboucle à l'étape précédente.

```
1 : right,  
2 : if B goto 3 else 1,  
3 : left,  
4 : if 1 goto 5 else 8,  
5 : write 0,  
6 : left,  
7 : goto 4,  
8 : write 1,  
9 :
```

Exercice 8. Décrire une MTI à 2 ou 3 rubans qui teste l'égalité de deux mots $m, m' \in \{0, 1\}^*$.

Correction. Précision sur l'énoncé : les mots sont tous deux donnés sur un même ruban d'entrée, le ruban 1, et séparés par un symbole blanc. Autrement dit, au début, le ruban d'entrée contient $\dots BBmBm'BB\dots$

Les étapes à implémenter sont les suivantes :

1. recopier m sur le ruban 2 (lignes de code 1 à 7 ci-dessous);
2. sauter le caractère blanc entre m et m' (ligne 8) ;
3. parcourir simultanément m sur le ruban 2 et m' sur le ruban 1 (à partir de la ligne 9)
 - si un caractère est différent, ou si une chaîne s'arrête avant l'autre, interrompre, effacer le ruban 1 et écrire 0 en sortie ;
 - si on finit le parcours sans différence, effacer le ruban 1 et écrire 0 en sortie.

Voici le début du code, la partie "effacer le ruban 1 et écrire la sortie" n'étant pas codée :

```
1 : right1,  
2 : if1 B goto 8 else 3,  
3 : if1 0 goto 4 else 6,  
4 : write2 0,  
5 : goto 1,
```

```
6 : write2 1,  
7 : goto 1,  
8 : right1,  
9 : if1 0 goto 10 else 11,  
10 : if2 0 goto 8 else <adresse du code pour écrire 0 en sortie>,  
11 : if1 1 goto 12 else 13,  
12 : if2 1 goto 8 else <adresse du code pour écrire 0 en sortie>,  
13 : if2 B goto <adr pour écrire 1 en sortie> else <adr pour écrire 0 en sortie>,  
14 : <suite du code, contenant les adresses pour écrire 0/1 en sortie>
```