

# TD 1 : Machines à registres

## 1 Définition

Une machine à registres est composée d'une **infinité** de registres :

$$R_1, R_2, \dots, R_i, \dots$$

Chaque registre peut contenir un entier positif ou nul. On note  $(R_i)$  le contenu du registre  $R_i$ .

Les seules actions élémentaires exécutables par la machine sont

— Remise à zéro d'un registre :

$$\{(R_i) = n\} R_i \leftarrow 0 \{(R_i) = 0\}$$

— Incrément de 1 :

$$\{(R_i) = n\} R_i ++ \{(R_i) = n + 1\}$$

— La répétitive "tant que" :

$$\mathbf{tant\ que\ } R_i \neq R_j \mathbf{\ faire\ } \langle \text{Programme} \rangle \mathbf{\ ftq}$$

Un **programme** pour la machine à registre est une suite d'actions élémentaires, séparées par des points-virgules. Par exemple :

```
R1 ← 0 ;
R2 ← 0 ;
R1 ++ ;
R1 ++ ;
tant que R1 ≠ R2 faire
    R2 ++ ;
ftq ;
```

**Premier exercice** : que fait ce programme ?

## 2 Quelques utilitaires arithmétiques

**Présentation des exercices** : les exercices sont présentés sous la forme

{ État initial} Programme demandé {État final}

Le travail demandé consiste à écrire le programme pour une machine à registre qui permet de passer de l'état initial à l'état final. On pourra ré-utiliser tout programme précédemment défini.

### 2.1 Initialisation

$n$  est un entier positif;  $\{(R_i) = m\} R_i \leftarrow n \{(R_i) = n\}$

### 2.2 Affectation

$\{(R_i) = n, (R_j) = m\} R_i \leftarrow R_j \{(R_i) = m, (R_j) = m\}$

### 2.3 Décrémentation

$\{(R_i) = n\} R_i -- \{(R_i) = n - 1\}$ ; si  $n = 0$  le résultat est indéterminé.

## 2.4 Addition

$\{(R_1) = n, (R_2) = m, (R_3) = p\} R_1 \leftarrow R_2 + R_3 \{(R_1) = m + p, (R_2) = m, (R_3) = p\}$   
(au moins 2 solutions possibles)

## 2.5 Soustraction

$\{(R_1) = n, (R_2) = m, (R_3) = p\} R_1 \leftarrow R_2 - R_3 \{(R_1) = m - p, (R_2) = m, (R_3) = p\}$   
Si  $(R_2) < (R_3)$  le résultat est indéterminé.

## 2.6 Quelques idées pour la suite

- Multiplication
- Exponentiation
- ...

# 3 Construction d'objets algorithmiques

Pour chacun des exercices ci-dessous, écrire le programme pour une machine à registres qui simule l'instruction demandée.

## 3.1 Conditionnelle simple

si  $R_i \neq R_j$  alors  $I$  ( $I$  est un programme).

## 3.2 Conditionnelle complète

si  $R_i \neq R_j$  alors  $I_1$  sinon  $I_2$  ( $I_1$  et  $I_2$  sont des programmes).  
En déduire la conditionnelle si  $R_i = R_j$  alors  $I_1$  sinon  $I_2$

## 3.3 Test d'infériorité

Écrire le programme qui met 1 dans le registre  $R_{\text{TEST}}$  si  $R_X \leq R_Y$  et 0 sinon.  
En déduire le programme qui met 1 dans le registre  $R_{\text{TEST}}$  si  $R_X > R_Y$  et 0 sinon.  
Écrire les répétitives

**tant que  $R_i \leq R_j$  faire < Programme > fintq**

et

**tant que  $R_i < R_j$  faire < Programme > fintq**

# 4 Pour aller plus loin

- Division euclidienne
- PGCD
- Coefficients de Bezout (ATTENTION : il faut utiliser des nombres négatifs. Si vous souhaitez programmer le calcul des coefficients de Bezout, vous pourrez supposer que la machine à registres les connaît...)
- ...

On pourra (c'est même indispensable!) utiliser les objets algorithmiques de la section précédente.