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 26/09/2018

Exercice 5. On considère une entrée donnée de la façon suivante :

- un entier $n \in \mathbb{N}^*$ est stocké dans `X0` ;
- une liste de n entiers, $(x_i)_{1 \leq i \leq n}$, est stockée dans les compteurs suivants : `Xi` contient x_i.

Décrire une SRAM qui calcule la somme $\sum_{i=1}^{n} x_i$ des valeurs de la liste.

Correction. Ici l'important est de comprendre le fonctionnement de l'adressage indirect des SRAM, analogue aux pointeurs en C/C++. `X`j `:= <X`i`>` va chercher la valeur de `X`i – disons que `X`i `=` p – puis s'en sert comme "adresse" c'est-à-dire comme indice d'une case mémoire : `X`j reçoit la valeur de `X`p. On peut donc comparer ça à écrire `X`j `= *X`i`;` en C.
Voici une solution utilisant les registres de travail `R0`, `R1`, ... séparés des `X`i. L'idée est d'itérer sur les éléments de la liste à partir de la fin. En effet comme `X0` contient n au début, `<X0>` donne `X`n. Ensuite on décrémente pour que `X0` prenne les valeurs $n, n-1, \ldots, 2$, ainsi `<X0>` parcourt `X`n`,` `X`$(n-1)$ `...,` `X2`. On accumule la somme dans `X1`, comme ça on calcule bien `X1 + X2 + ...+ X`n ; puis on renvoie le résultat copié dans `X0`. D'où le code :

```
1 : if X0 = 0 goto 6 else 2,
2 : R0 := <X0>,
3 : X1 := X1 + R0,
4 : X0 := X0 - 1,
5 : goto 1,
6 : X0 := X1,
7 :
```

Supposons maintenant qu'on veuille faire le calcul *sans recours aux registres* `R`i. On va voir qu'on peut quand même s'en sortir (mais c'est un peu tordu, ça ne sera pas demandé au partiel). On utilise `X1` à la place de `R0`. Il faut alors sauvegarder cet élément `X1` du tableau quelque part... Mettons-le dans `X`$(n+1)$; ainsi le tableau sera constitué des éléments `X2, ..., X`$(n+1)$. (On a changé l'ordre des valeurs, le 1er élément étant devenu le dernier ; ici ce n'est pas grave car l'addition ne dépend pas de l'ordre, par contre en général il faut se méfier.) Le code :

```
1 : X0 := X0 + 1, // pour avoir un pointeur sur n + 1
2 : <X0> := X1,
3 : X0 := X0 - 1,
4 : if X0 <= 1 goto 9 else 5,
5 : X1 := <X0>,
6 : X2 := X2 + X1,
7 : X0 := X0 - 1,
8 : goto 4,
9 : X0 := X2,
10 :
```

Notons que la macro `6 : X2 := X2 + X1` peut être dépliée en le code suivant, ce qui permet de vérifier qu'il n'y a pas d'arnaque puisqu'elle ne cache aucun autre registre de travail :

```
6.1 : if X1 = 0 goto 7 else 6.2,
6.2 : X1 := X1 - 1,
6.2 : X2 := X2 + 1,
6.4 : goto 6.1,
```
Ce qu'on n'aurait pas pu faire, c'est utiliser `X`$(n+1)$ à la place de `R0`, puisque `X`$(n+1)$ n'est pas manipulable directement : en effet n n'est pas une constante, mais fait partie des données, et donc on doit employer l'adressage indirect (autrement dit un pointeur) pour accéer à `X`$(n+1)$, mais où stocker ce pointeur ?

Exercice 7. Soient deux fonctions $f : \mathbb{N} \to \mathbb{N}$ et $g : \mathbb{N} \to \mathbb{N}$, calculées par les deux SRAM respectives $\mathcal{M}_f$ et $\mathcal{M}_g$. Ces machines prennent toutes deux leur entrée sur `X0` et écrivent également toutes deux leur sortie sur `X0`.
Comment réaliser la composition $f \circ g$ par une SRAM ?

Correction. Au départ, `X0` contient une entrée x. En exécutant $\mathcal{M}_g$, `X0` contient $g(x)$. Puis en exécutant $\mathcal{M}_f$ juste après, `X0` contient $f(g(x))$. Il suffit donc de mettre les codes de $\mathcal{M}_g$ et $\mathcal{M}_f$ à la suite !
Une subtilité cependant : il faut que $\mathcal{M}_f$ suppose que les registres qu'elle utilise ne sont pas forcément nuls au début, et les réinitialise à 0 lors de leur première utilisation si besoin. On peut toujours modifier le code de $\mathcal{M}_f$ pour garantir ça. Sinon, le fait que $\mathcal{M}_g$ ait été exécutée avant, et ait touché à des registres, peut poser problème.
Sur une machine à compteurs, comme on connaît à l'avance les registres utilisés, c'est simple de tous les remettre à 0 entre l'appel à $\mathcal{M}_g$ et celui à $\mathcal{M}_f$. Pour une SRAM, l'ensemble des registres utilisés varie en fonction de l'entrée...