

l'*accumulateur* et un champ qui contient une liste d'entiers et que l'on appelle la *pile*. Sur cet objet, on peut exécuter les opérations

- **Ldi**  $n$  : mettre l'entier  $n$  dans l'accumulateur,
- **Push** : mettre le contenu de l'accumulateur sur le sommet de la pile,
- **Add** : ajouter le sommet de la pile et l'accumulateur, mettre le résultat dans l'accumulateur et retirer le sommet de la pile.

Cet objet est notre machine abstraite, et ces trois instructions constituent son langage machine. Les champs d'une machine abstraite s'appellent des *registres*.

Le second programme prend en argument un terme de PCF et, en fonction de ce terme, il donne des instructions à la machine, que celle-ci exécute au fur et à mesure. Si  $t$  est un terme de PCF, on note  $|t|$  la suite d'instructions données par ce programme à la machine abstraite lors de l'interprétation de ce terme. Par exemple, pour le terme  $((((1 + 2) + 3) + 4) + 5) + 6$ , les instructions suivantes sont données à la machine abstraite : **Ldi 6, Push, Ldi 5, Push, Ldi 4, Push, Ldi 3, Push, Ldi 2, Push, Ldi 1, Add, Add, Add, Add, Add**.

**Exercice 4.1** *Quelles sont les instructions données à la machine abstraite pour interpréter le terme  $1 + (2 + (3 + (4 + (5 + 6))))$  ?*

Ce partage du travail peut être comparé à celui qui s'opère entre le passager et le conducteur d'une voiture dans une ville inconnue : le passager lit la carte et donne des instructions à voix haute au conducteur, qui les exécute sans savoir où il va.

S'il détermine ses instructions en regardant uniquement la carte, le passager peut enregistrer la suite d'instructions sur un disque compact, que le conducteur écoute sur l'autoradio. Ainsi, le passager n'a pas réellement besoin d'être dans la voiture pour guider le conducteur. De même, un interpréteur peut mettre la suite d'instructions  $|t|$  dans un fichier, et ce fichier peut être exécuté par la machine abstraite à un autre moment. On a ainsi transformé l'interpréteur en un compilateur.

En général, on considère que la machine abstraite contient, outre l'accumulateur et la pile, un troisième registre : le *code* qui est la liste des instructions à exécuter. Quand on l'allume, la machine abstraite va donc chercher une instruction dans le registre code, l'exécute, puis va chercher une autre instruction... jusqu'à ce que le registre contenant le code soit vide. Comme nous allons le voir, le fait que l'exécution d'une instruction puisse ajouter des instructions au registre code permet de faire des boucles et des définitions récursives.

## 4.3 Une machine abstraite pour PCF

### 4.3.1 L'environnement

Pour le moment nous n'avons compilé qu'un fragment de PCF, qui contient les nombres entiers et l'addition. Comment généraliser ce principe à PCF en entier ?