

chine abstraite n'est pas motivée uniquement par des raisons pédagogiques : les principaux compilateurs de Caml et de Java, par exemple, compilent vers de telles machines abstraites. Les programmes compilés sont ensuite exécutés par un programme qui simule le fonctionnement de cette machine, ou bien ils sont traduits, lors d'une seconde passe, dans le langage machine d'une machine concrète.

4.1 Un interpréteur écrit dans un langage sans fonctions

Au chapitre 2, nous avons présenté les règles de la sémantique opérationnelle à grands pas de PCF et nous en avons tiré les principes de l'écriture d'un interpréteur. Par exemple, la règle

$$\frac{e \vdash u \hookrightarrow q \quad e \vdash t \hookrightarrow p}{e \vdash t + u \hookrightarrow n} \text{ si } p + q = n$$

nous amène à écrire, en Caml, le bout d'interpréteur suivant

```
let rec interp env p = match p with
| Plus(t,u) ->
  let w = interp env u
  in let v = interp env t
     in (match (v,w) with | (Const(n), Const(m)) -> Const(n + m)
                       | ...)
| ...
```

Le fait que Caml permette des définitions locales fait que l'on peut calculer la valeur du terme `interp env t` et retrouver la valeur `w` intacte après ce calcul, même si la variable `w` a été liée à d'autres valeurs durant ce calcul.

Si nous voulions écrire un interpréteur en langage machine, ou dans n'importe quel langage qui n'a pas de définitions locales, nous devrions utiliser un mécanisme pour mémoriser la valeur `w`, par exemple une pile : cela demanderait d'interpréter le terme `u`, puis de mettre le résultat sur le sommet de la pile, d'interpréter ensuite le terme `t` et enfin de retirer le sommet de la pile en l'ajoutant au résultat de cette interprétation.

Avec un tel interpréteur, interpréter le terme $((((1 + 2) + 3) + 4) + 5) + 6$ demande de mettre l'entier 6, puis l'entier 5, ..., puis l'entier 2 sur le sommet de la pile, puis d'ajouter le sommet 2 de la pile à l'entier 1, puis le sommet 3 de la pile à l'entier obtenu, puis... puis le sommet 6 de la pile à l'entier obtenu, ce qui donne le résultat : 21.

4.2 De l'interprétation à la compilation

Cet interpréteur peut se découper en deux programmes. Le premier est un objet muni de deux champs : un champ qui contient un entier et que l'on appelle