

- $\Theta(t - u, e, m) = \Theta(t, e, m) - \Theta(u, e, m)$,
- $\Theta(t * u, e, m) = \Theta(t, e, m) * \Theta(u, e, m)$,
- $\Theta(t / u, e, m) = \Theta(t, e, m) / \Theta(u, e, m)$,
- $\Theta(t \% u, e, m) = \Theta(t, e, m) \% \Theta(u, e, m)$,
- si $\Theta(b, e, m) = \text{true}$ alors
 - $\Theta((b) ? t : u, e, m) = \Theta(t, e, m)$
- si $\Theta(b, e, m) = \text{false}$ alors
 - $\Theta((b) ? t : u, e, m) = \Theta(u, e, m)$

À première vue, cette définition peut sembler circulaire, car pour définir la valeur d'une expression de la forme $t + u$, nous utilisons la valeur des expressions t et u . Cette définition est cependant bien formée car les expressions t et u sont de plus petite taille que $t + u$. Cette définition est donc une définition par récurrence sur la taille des expressions.

La première clause de cette définition indique que la valeur d'une expression qui est une variable mutable est $m(e(x))$. On applique donc d'abord la fonction e à la variable x , ce qui produit une référence, puis la fonction m à cette référence, ce qui produit une valeur. Si la variable est finale, en revanche, on trouve directement la valeur dans l'environnement.

La définition de la fonction Θ de Caml est identique, sauf, précisément, dans le cas des variables où l'on a une unique clause

- $\Theta(x, e, m) = e(x)$,

que la variable x soit finale ou mutable.

Par exemple, si e est l'environnement $[x = r]$ et m la mémoire $[r = 4]$ et que la variable x est mutable dans e , la valeur $\Theta(x, e, m)$ est 4 en Java, mais c'est r en Caml.

Caml comporte ensuite une construction ! telle que

- $\Theta(!t, e, m) = m(\Theta(t, e, m))$

Si x est une variable, alors la valeur de $!x$ est $\Theta(!x, e, m) = m(\Theta(x, e, m)) = m(e(x))$ qui est la valeur de x en Java. On explique ainsi que l'on écrive $y := !x + 1$ en Caml, ce que l'on écrit $y = x + 1$; en Java.

En Caml, les références susceptibles d'être associées à un entier dans la mémoire sont de type `int ref`. Par exemple, la variable x et la valeur r de cet exemple sont de type `int ref`. À la différence de la variable x , les expressions $!x, !x + 1, \dots$ sont de type `int`.

La définition de la fonction Θ de C est la même que celle de Java.

Exercice 1.4

Donner la définition de la fonction Θ pour les expressions de la forme $t \& u$ et $t | u$.

À la différence de l'opérateur booléen $\&$ qui évalue ses deux arguments, l'opé-