

2.2 La sémantique des fonctions

Venons-en maintenant à l'extension de la définition de la fonction Σ . Nous devons faire face à plusieurs nouveautés. Tout d'abord, la fonction Σ doit prendre en arguments, outre une instruction, un environnement et une mémoire, un *environnement global* \mathbf{G} . Cet environnement global est constitué d'un environnement \mathbf{e} qui contient les variables globales et d'une fonction de domaine fini qui, à chaque nom de fonction, associe sa définition, c'est-à-dire ses arguments formels et le corps de la fonction à exécuter à chaque appel.

Ensuite, nous devons prendre en compte le fait que, puisque les fonctions peuvent modifier la mémoire, l'évaluation d'une expression peut modifier la mémoire. De ce fait, le résultat de l'évaluation d'une expression, quand il existe, ne sera plus simplement une valeur, mais un couple formé d'une valeur et d'une mémoire.

Nous devons également expliquer ce qui se passe quand on exécute l'instruction **return**, en particulier le fait que l'exécution de cette instruction interrompt l'exécution du corps de la fonction. Cela nous amène à reconsidérer la définition de la fonction Σ dans le cas de la séquence

$$\Sigma(\{p_1 \ p_2\}, \mathbf{e}, \mathbf{m}, \mathbf{G}) = \Sigma(p_2, \mathbf{e}, \Sigma(p_1, \mathbf{e}, \mathbf{m}, \mathbf{G}), \mathbf{G})$$

selon laquelle exécuter une séquence $\{p_1 \ p_2\}$ consiste à exécuter p_1 puis p_2 .

En effet, si p_1 est de la forme **return** t ;, ou plus généralement si l'exécution de p_1 amène à exécuter un **return**, alors l'instruction p_2 ne doit pas être exécutée. Nous allons donc considérer que le résultat $\Sigma(p_1, \mathbf{e}, \mathbf{m}, \mathbf{G})$ de l'exécution de p_1 dans l'état \mathbf{e} , \mathbf{m} n'est pas simplement une mémoire, mais un objet plus complexe. Une première partie de cet objet est un booléen qui indique si l'exécution de p_1 s'est déroulée normalement, ou si un **return** a été rencontré. Si l'instruction s'est déroulée normalement, la seconde partie de cet objet est la mémoire produite par cette exécution. Si une instruction **return** a été rencontrée, alors la seconde partie de cet objet est formée d'une part de la valeur à renvoyer et d'autre part de la mémoire produite par cette exécution. L'ensemble d'arrivée de la fonction Σ sera donc désormais $(\{\mathbf{normal}\} \times \mathbf{Mem}) \cup (\{\mathbf{return}\} \times \mathbf{Val} \times \mathbf{Mem})$ où \mathbf{Mem} est l'ensemble des mémoires, c'est-à-dire des fonctions d'une partie finie de l'ensemble \mathbf{Ref} dans l'ensemble \mathbf{Val} .

Nous devons enfin prendre en compte le fait qu'une fonction peut être appelée non seulement depuis le programme principal — la fonction **main** — mais aussi depuis une autre fonction, mais laissons cela pour le moment.