

```
x = x + 1;
```

sont des instructions en Java, alors que les suites de symboles

```
y + 3 = x;
```

```
x + 2 = y + 5;
```

n'en sont pas.

Pour expliquer ce qui se passe quand on exécute l'instruction  $x = t$  ; on doit supposer qu'il y a, dans un recoin de la mémoire d'un ordinateur, une case appelée  $x$ . Exécuter l'instruction  $x = t$  ; consiste alors à remplir cette case avec la *valeur* de l'expression  $t$ . La valeur contenue antérieurement dans la case  $x$  est effacée. Si l'expression  $t$  est une constante, par exemple  $3$ , sa valeur est cette même constante. Si c'est une expression sans variables, comme  $3 + 4$ , sa valeur s'obtient en effectuant quelques opérations arithmétiques, ici une addition. Si l'expression  $t$  contient des variables, alors il faut aller chercher les valeurs correspondant à ces variables dans les cases de la mémoire de l'ordinateur. L'ensemble des contenus des cases de la mémoire de l'ordinateur s'appelle un *état*.

Nous allons considérer, dans un premier temps, que les expressions, comme  $x + 3$  et les instructions, comme  $y = x + 3$  ;, forment deux catégories disjointes. Nous serons cependant amenés à relativiser un peu ce postulat par la suite.

Dans ces exemples, les valeurs des expressions sont des nombres entiers relatifs. En fait, la valeur d'une expression ne peut pas être un entier arbitraire et une telle valeur appartient toujours à un intervalle fini. En Java, les valeurs entières sont comprises entre  $-2^{31}$  et  $2^{31} - 1$ , il y a donc  $2^{32}$  valeurs différentes. Quand les opérations arithmétiques amènent à sortir de cet intervalle, le résultat  $y$  est ramené en prenant son reste modulo  $2^{32}$ . Ainsi, en ajoutant  $1$  à  $2^{31} - 1$ , c'est-à-dire  $2147483647$ , on sort de l'intervalle et on y revient en retranchant  $2^{32}$ , ce qui donne  $-2^{31}$  soit  $-2147483648$ .

### Exercice 1.1

Quel est le contenu de la variable  $x$  à l'issue de l'exécution de l'instruction suivante ?

```
x = 2 * 1500000000;
```

*En Caml, l'affectation s'écrit  $x := t$ . Dans l'expression  $t$ , on désigne le contenu de la case  $x$  non par l'expression  $x$  elle-même, mais par l'expression  $!x$ . Ainsi, on écrit en Caml  $y := !x + 1$  ce que l'on écrit  $y = x + 1$  ; en Java.*

*En C, l'affectation s'écrit comme en Java.*