fini et que l'ensemble des variables est infini. Par exemple, l'expression `z + 3` n'a pas de valeur dans l'état `[x = 5, y = 6]`. En pratique, cela signifie que la tentative de calculer la valeur de l'expression `z + 3` dans l'état `[x = 5, y = 6]` déclenche une erreur.

L'exécution d'une instruction dans un état produit un autre état, il est donc naturel de décrire ce qui se passe quand on exécute une instruction d'un langage par une fonction $\Sigma$, qui à une instruction `p` et un état initial `s` associe un état final $\Sigma$`(p,s)`. Cette fonction aussi est partielle. Elle n'est pas définie quand l'exécution de `p` dans l'état `s` produit une erreur ou ne termine pas.

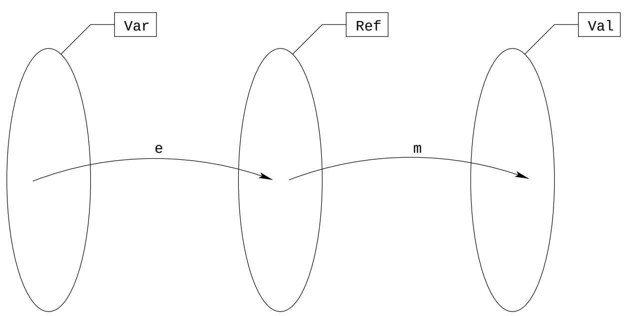
Dans le cas où l'instruction `p` est de la forme `x = t;`, cette fonction se définit de la manière suivante

$$\Sigma(\mathtt{x = t;,s}) = \mathtt{s + (x =} \ \Theta(\mathtt{t,s})\mathtt{)}.$$

Par exemple $\Sigma$`(x = x + 1;,[x = 5]) = [x = 6]`. Cette définition traduit la phrase « Exécuter l'instruction `x = t;` consiste à remplir la case `x` avec la valeur de l'expression `t` ».

### 1.3.2 La décomposition de l'état

Un état `s` est une fonction d'une partie finie de l'ensemble `Var` dans l'ensemble `Val`. Il sera utile, à partir du prochain chapitre, de décomposer cette fonction comme la composition de deux fonctions de domaine fini : la première, l'*environnement*, d'une partie finie de l'ensemble `Var` dans un ensemble intermédiaire `Ref`, dont les éléments sont appelés *références*, et la seconde, la *mémoire*, d'une partie finie de l'ensemble `Ref` dans l'ensemble `Val`.




Cela amène à poser deux ensembles infinis `Var` et `Ref` et un ensemble `Val` de valeurs. L'ensemble des *environnements* se définit comme l'ensemble des fonctions d'une partie finie de l'ensemble `Var` dans l'ensemble `Ref` et l'ensemble des *mémoires* comme l'ensemble des fonctions d'une partie finie de l'ensemble `Ref` dans l'ensemble `Val`. Sur l'ensemble des environnements, on définit une