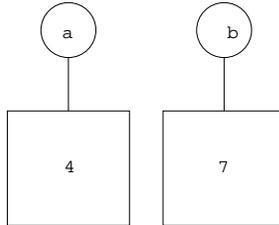


L'affectation d'un champ mutable s'écrit `x.c <- 4`.

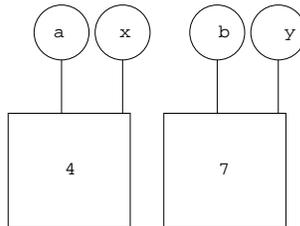
La fonction

```
let swap x y = let z = x.c in (x.c <- y.c; y.c <- z)
```

intervertit le contenu de deux variables de type `int_wrap`. En effet, quand on exécute l'instruction `swap a b` dans l'état formé de l'environnement `[a = r, b = r']` et la mémoire `[r = 4, r' = 7]`



les valeurs associées aux variables `x` et `y` sont les références `r` et `r'` et non les entiers `4` et `7`



et donc intervertir le contenu de `x` et `y` intervertit le contenu de `a` et `b`.

On peut donc éviter d'utiliser la construction `ref` en utilisant une variable finale de type enveloppé, comme en Java. En fait, le type `int ref` de Caml n'est pas primitif, mais c'est une abréviation pour un type enveloppé comme celui-ci.

Exercice 4.4

Quel est l'état construit par les déclarations suivantes ?

```
type point = {
  mutable latitude : float;
  mutable longitude : float;
  mutable altitude : float;}
```

```
let x = {latitude = 48.715; longitude = 2.208; altitude = 156.0;}
```

et par les déclarations suivantes ?