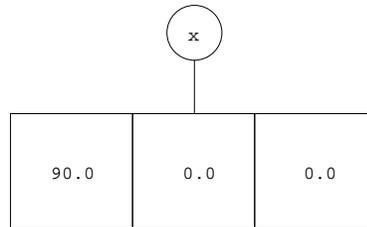


Et, avec une variable finale, la déclaration `let x = {latitude = 90.0; longitude = 0.0; altitude = 0.0;}` crée l'état



Une variable `x` de type `point ref` est toujours associée à une référence, elle même associée dans la mémoire à un enregistrement et il n'y a pas de valeur null en Caml.

### 4.3.3 L'accès aux champs

Si la valeur de l'expression `t` est une référence `r'` associée dans la mémoire à un enregistrement, la valeur de l'expression `t.l` est le champ `l` de cet enregistrement.

En particulier, quand `t` est de la forme `!x` où `x` est une variable mutable, sa valeur est `m(e(x))` et donc la valeur de l'expression `(!x).latitude` est le champ `latitude` de l'enregistrement `m(m(e(x)))`, comme la valeur de l'expression `x.latitude` en Java. Et quand `t` est une variable finale `x`, sa valeur est `e(x)` et donc la valeur de l'expression `x.latitude` est le champ `latitude` de l'enregistrement `m(e(x))` comme en Java.

### 4.3.4 L'affectation des champs

En Caml, tous les champs sont finaux par défaut et on doit indiquer quand ils sont mutables avec le mot clé `mutable`.

```
type int_wrap = {mutable c : int}
```