rateur `&&` n'évalue son second argument que si l'évaluation du premier a donné le résultat `true`. Donner la définition de la fonction Θ pour les expressions de la forme `t && u`.

Même question pour l'opérateur booléen `||`, qui n'évalue son second argument qui si l'évaluation du premier a donné le résultat `false`.

1.3.5 L'exécution des instructions

La fonction Σ associe maintenant une mémoire à des triplets formés d'une instruction, d'un environnement et d'une mémoire. La fonction Σ de Java se définit ainsi.

- Quand l'instruction `p` est une déclaration de variable mutable de la forme `{T x = t ; q}`, la fonction Σ se définit ainsi
  Σ`({T x = t ; q},e,m) =` Σ`(q,e + (x = r),m + (r =` Θ`(t,e,m)))`
  où `r` est une référence quelconque qui n'apparaît pas dans `e` et `m`.
- Quand l'instruction `p` est une déclaration de variable finale de la forme `{final T x = t ; q}`, la fonction Σ se définit ainsi
  Σ`({final T x = t ; q},e,m) =` Σ`(q,e + (x =` Θ`(t,e,m)),m)`
- Quand l'instruction `p` est une affectation de la forme `x = t ;`, cette fonction se définit ainsi
  Σ`(x = t ;,e,m) = m + (e(x) =` Θ`(t,e,m))`.
- Quand l'instruction `p` est une séquence de la forme $\{p_1\ p_2\}$, la fonction Σ se définit ainsi
  $\Sigma(\{p_1\ p_2\},e,m) = \Sigma(p_2,e,\Sigma(p_1,e,m))$
- Quand l'instruction `p` est un test de la forme `if (b)` p_1 `else` p_2, la fonction Σ se définit ainsi. Si Θ`(b,e,m) = true` alors
  Σ`(if (b)` p_1 `else` p_2`,e,m) =` $\Sigma(p_1,e,m)$
  Si Θ`(b,e,m) = false` alors
  Σ`(if (b)` p_1 `else` p_2`,e,m) =` $\Sigma(p_2,e,m)$
- Venons-en maintenant au cas où l'instruction `p` est une boucle de la forme `while (b) q`. Nous avons vu qu'en introduisant une instruction fictive `skip ;` telle que Σ`(skip ;,e,m) = m`, on peut définir l'instruction `while (b) q` comme une abréviation pour l'instruction infinie

```
if (b) {q if (b) {q if (b) {q if (b) ...
                                      else skip;}
                            else skip;}
                  else skip;}
        else skip;
```

  Quand on doit traiter avec de tels objets infinis, on cherche souvent à les approcher comme limites d'approximations finies. Ainsi, en introduisant