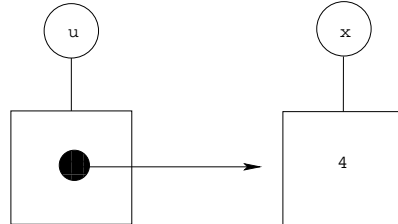


associe une référence r' à une référence r , une solution est d'écrire dans la case r les coordonnées de l'endroit où l'on a dessiné la référence r' sur la feuille. Une meilleure solution est de dessiner dans la case r une flèche qui désigne la référence r' .



Si t est une expression de type T^* alors le langage C comporte une nouvelle construction d'affectation $*t = u$; , homologue à la construction $:=$ de Caml : si la valeur de t est une référence r et celle de u est v , alors l'exécution de l'instruction $*t = u$; associe la valeur v à la référence r dans la mémoire.

Exercice 2.13

Montrer que l'exécution de l'instruction $x = u$; a le même effet que celle de l'instruction $*\&x = u$. Qu'en déduire?

Ces constructions permettent alors d'écrire une fonction `swap` qui prend en arguments deux références et permute les valeurs associées à ces références dans la mémoire.

```
void swap (int* const x, int* const y) {
    int z;
    z = *x;
    *x = *y;
    *y = z;}

```

Pour permuter le contenu des variables `a` et `b`, il suffit maintenant d'appliquer cette fonction aux références `&a` et `&b`.

```
int main () {
    a = 4;
    b = 7;
    swap(&a,&b);
    printf("%d\n",a);
    printf("%d\n",b);
    return 0;}

```