

Exercice 1

On rappelle qu'exécuter l'instruction $x = t$; dans l'état s , produit l'état $s + (x = V)$ où V est la valeur de l'expression t dans l'état s et où la notation $s + (x = V)$ désigne l'état identique à s sauf pour la variable x qui prend la valeur V .

On rappelle aussi qu'exécuter $p1$ $p2$ dans l'état s , produit l'état s'' , produit en exécutant $p2$ dans l'état s' , produit en exécutant $p1$ dans l'état s .

Soit s l'état $[x = 7, y = 8, z = 9]$.

1. Quel est l'état produit par l'exécution dans s de l'instruction

$x = 4$;

?

2. Quel est l'état produit par l'exécution dans s de l'instruction

$x = 4$;

$y = 5$;

?

3. Quel est l'état produit par l'exécution dans s de l'instruction

$y = 5$;

$x = 4$;

?

4. Quel est l'état produit par l'exécution dans s de l'instruction

$x = y$;

$y = x$;

?

Le contenu des variables x et y a-t-il été interverti?

5. Donner une instruction qui intervertit le contenu des variables x et y .
C'est-à-dire qui exécutée dans l'état $[x = v, y = v', z = v']$ produit un état dans lequel le contenu de la variable x est v' et celui de la variable y est v .

Exercice 2

On dit que deux instructions p et q sont *équivalentes* si elles ont la même sémantique, c'est-à-dire si dans tout état s , exécuter p et exécuter q produit le même état.

1. Donner deux instructions p et q telles que les instructions $p \ q$ et $q \ p$ soient équivalentes.
2. Donner deux instructions p et q telles que les instructions $p \ q$ et $q \ p$ ne soient pas équivalentes.
3. Montrer que les instructions $\{p \ q\} \ r$ et $p \ \{q \ r\}$ sont toujours équivalentes.

Exercice 3

1. Comment exprimer la sémantique de l'instruction `skip` qui ne fait rien ?
2. Le test incomplet (sans `else`) est une instruction de la forme `if (t) p`. Sa sémantique est la suivante :
 - si la valeur de t dans s est `true`, alors exécuter `if (t) p` dans l'état s produit l'état s' , produit par l'exécution de p dans l'état s ,
 - si la valeur de t dans s est `false`, alors exécuter `if (t) p` dans l'état s produit l'état s .
 Montrer que cette construction est redondante, c'est-à-dire que toute instruction qui utilise cette construction peut se transformer en une instruction équivalente qui ne l'utilise pas.

Exercice 4

1. Montrer que l'instruction


```
u = 1;
x = 0;
repeat {
  u = u * 2;
  x = x + 1;
}
until (x == 3)
```

 peut se transformer en une instruction équivalente qui n'utilise pas de boucle.
2. La boucle est-elle redondante ? Donner un exemple d'instruction qui peut s'exprimer avec une boucle, mais qu'on ne peut pas transformer en une instruction équivalente sans boucle.

Exercice 5

Donner une instruction p et deux états s et s' tels que p termine quand on l'exécute dans l'état s mais pas quand on l'exécute dans l'état s' .