Correction

Après s'être assuré qu'un algorithme termine (cours sur la terminaison), on doit s'assurer que l'algorithme réalise bien la tâche qui lui est confiée. Ceci se réalise grâce à des démonstrations assez identiques aux démonstrations que vous pouvez faire en cours de mathématiques : une démonstration se compose d'applications de théorèmes et de phases de calculs ; la difficulté étant de savoir comment appliquer les théorèmes.

Pour les preuves de corrections nous allons avoir 6 grands thorèmes, le reste étant un enchaînement de formules logique. Un théorème de correction se présente de la façon suivante : $\{P\}\ c\ \{Q\}$. P et Q sont respectivement les pré et post-conditions. c est un ensemble d'instructions exprimées dans notre langage restreint. P représente l'état de la mémoire avant l'exécution des instructions c, et Q l'état de la mémoire après leur exécution.

1 – On note $\{P\} = \{y = 5\}$. Quel est l'état de la mémoire après l'exécution de l'instruction $x \leftarrow y$? Et après l'exécution de l'instruction $y \leftarrow 3$? Et si l'on lui applique les deux instructions $x \leftarrow y; y \leftarrow 3$?

Nous allons ici décrire les différents théorèmes qui permettent de vérifier la correction d'un algorithme.

I Affectation de renommage

Définition 1 :

On appelle affectation de renommage une instruction du type $X \leftarrow a$ où a ne contient pas X. On donne donc un nom à l'expression a.

Par exemple $X \leftarrow 3 + 2$ est une affectation de renommage, tout comme $X \leftarrow Y$ ou $X \leftarrow Y + 5$. Mais $X \leftarrow X + 1$ non.

Définition 2 : substitution

Si P est une condition (un état mémoire) on note $P[a/X]$ la même condition dans laquelle toutes les occurrences de X sont remplacées par a.

Remarque :Si on connait P, on peut facilement décrire $P[a/X]$. Par contre P s'obtient en remplaçant certaines occurrences de a par X : il n'y a pas unicité.

2 – On note $P = \{X = Y \wedge Y = 5\}$, donner $P[Z/X]$.

3 – Si $P[a/X]$ est $\{a = 3 \wedge b = a\}$, donner les possibilités pour P.

Théorème 1

Dans le cas d'une affectation de renommage, le théorème qui s'applique est le suivant :

$$\{P[a/X]\}\ X \leftarrow a\ \{P\}.$$

4 – Écrire la preuve de l'algorithme suivant :

$\{x = X \wedge y = Y\}$
$t \leftarrow x;$
$x \leftarrow y;$
$y \leftarrow t;$
$\{y = X \wedge x = Y\}$

On doit s'assurer avant d'appliquer le théorème que la pré-condition ne contienne plus aucune occurrence de X. Il faudra parfois faire une étape de calcul pour cela.

II Affectation d'incrémentation

Définition 3 : incrémentation

On appelle affectation d'incrémentation une instruction du type $X \leftarrow a$ où a contient au moins une occurrence de X.

L'exemple le plus simple d'affectation d'incrémentation est l'instruction $X \leftarrow X + 1$.

Théorème 2

Le théorème d'affectation d'incrémentation est le même mais l'utilisation diffère : pour appliquer le théorème

$$\{P[a/X]\}\ X \leftarrow a\ \{P\}$$

on remplace *toutes* les occurences de a par X.

5 – Donner la post-condition obtenue après exécution de $X \leftarrow X + 5$ à la pré-condition $\{X = 4 \wedge Y = X + 5\}$.

6 – Écrire la preuve de l'algorithme suivant. Quel est son rôle ? Quelle est sa particularité ?

$\{x = X \wedge y = Y\}$
$x \leftarrow x + y;$
$y \leftarrow x - y;$
$x \leftarrow x - y;$
$\{y = X \wedge x = Y\}$

III Enchaînements, conséquences

Comme nous l'avons vu, on peut enchaîner les instructions. Ceci se traduit par un théorème simple :

Théorème 3

Si $\{P_1\}\ c_1\ \{P_2\}$ et $\{P_2\}\ c_2\ \{P_3\}$ sont deux preuves correctes, alors $\{P_1\}\ c_1; c_2\ \{P_3\}$ est une preuve correcte.

C'est ce théorème qui permet de parler de preuve d'un programme. Nous l'avons en fait implicitement utilisé jusque là.

On peut aussi, comme dans une vraie démonstration, effectuer de simples étapes de calcul, qui permettent de de simplifier des conditions par exemple. Le théorème associé est le suivant :

Théorème 4

Si $P' \Rightarrow P$, si $\{P\}\ c\ \{Q\}$ est une preuve correcte et si $Q \Rightarrow Q'$, alors $\{P'\}\ c\ \{Q'\}$ est une preuve correcte.

7 – Effectuer la preuve suivante :

{X=n}
$X \leftarrow X + 1;$
{X=n+1}

8 – Montrer que la preuve suivante est correcte :

{}
$X \leftarrow 5;$
{X=5}

La règle de conséquences permet donc en particulier de gérer l'initialisation des variables. Lorsque l'on démarre un algorithme, l'état de la mémoire est {}.

IV Conditions

On s'attaque maintenant aux instructions conditionnelles du type *c :=Si* b *alors* c_1 *sinon* c_2 *finsi* où b est un test et c_1 et c_2 sont deux blocs d'instructions. Le théorème associé est le suivant :

Théorème 5

Si $\{P \wedge b\}\ c_1\ \{Q_1\}$ et $\{P \wedge \neg b\}\ c_2\ \{Q_2\}$ sont deux preuves correctes, alors $\{P\}\ c\ \{Q_1 \vee Q_2\}$ est une preuve correcte.

Il faut donc s'intéresser à la preuve de chacune des sous-instructions.

9 – Montrer que l'algorithme suivant renvoie toujours 0.

Entrées : x un entier tel que $1 \leq x \leq 3$
$\{\}$
si $x == 1$ **alors**
 | $y \leftarrow x - 1$
sinon si $x == 2$ **alors**
 | $y \leftarrow x - 2$
sinon
 | $y \leftarrow x - 3$
fin
$\{y = 0\}$
retourner y

10 – Montrer que l'algorithme suivant renvoie le maximum entre les deux entrées.

Entrées : Deux réels X et Y
$\{\}$
si $X \leq Y$ **alors**
 | $MAX \leftarrow Y$
sinon
 | $MAX \leftarrow X$
fin
$\{MAX = \max(X, Y)\}$
retourner MAX

11 – Montrer que l'algorithme suivant calcule la valeur absolue de l'entrée.

Entrées : un réel x
Sorties : la valeur absolue de x
$\{x \in \mathbb{R}\}$
si $x < 0$ **alors**
 | $absx \leftarrow -x$
sinon
 | $absx \leftarrow x$
fin
$\{absx = |x|\}$
retourner *absx*

V Boucles

On se concentre sur les boucles du type *Tant que*, car les boucles *Pour* n'en sont que des cas particuliers. On pose de manière générale une instruction c sous la forme d'une boucle : *c :=Tant que* b *faire* c_0 *fintantque*.

Définition 4 : invariant

On appelle invariant d'un bloc d'instructions c_0 sous la contrainte b une condition P telle que

$$\{P \wedge b\}\ c_0\ \{P\}$$

soit une preuve correcte.
Si c_0 est le corps d'une boucle *Tant que* b *faire* c_0 *fintantque*, on dit que P est un *invariant de boucle* pour cette boucle.

Théorème 6

Si P est une invariant de boucle pour la boucle c, alors la preuve :

$$\{P\} \text{ Tant que } b \text{ faire } c_0 \text{ fintantque} \{P \wedge \neg b\}$$

est une preuve correcte.

Pour faire la preuve de correction d'une boucle, il faut donc déterminer un invariant : il n'y a pas de méthode générale pour effectuer ce travail, il faut chercher ce qui convient le mieux. Il n'y a d'ailleurs pas unicité de l'invariant de boucle.

12 – Montrer que dans l'algorithme suivant, la condition $\{somme = 1 + \cdots + i \wedge i - 1 < n\}$ est un invariant de boucle.

```
Entrées : un entier n
Sorties : la somme des n premiers entiers
i ← 0;
somme ← 0;
tant que i < n faire
    i ← i + 1;
    somme ← somme + i;
fin
retourner somme
```

13 – Montrer alors que cet algorithme calcule la somme des n premiers entiers.

14 – Montrer que dans l'algorithme suivant, $\{X = R + Q \times Y \wedge R \geq 0\}$ est un invariant de boucle.

```
Entrées : X et Y deux entiers positifs.
Sorties : R et Q
R ← X;
Q ← 0;
tant que Y ≤ R faire
    R ← R − Y;
    Q ← Q + 1;
fin
retourner (R, Q)
```

15 – Que fait cet algorithme ? En faire la preuve.

VI Document réponse

$\{x = X \wedge y = Y\}$
$t \leftarrow x;$
$\{t = X \wedge y = Y\}$
$x \leftarrow y;$
$\{t = X \wedge x = Y\}$
$y \leftarrow t;$
$\{y = X \wedge x = Y\}$

$\{x = X \wedge y = Y\}$
$x \leftarrow x + y;$
$\{x - y = X \wedge y = Y\}$
$y \leftarrow x - y;$
$\{y = X \wedge x - y = Y\}$
$x \leftarrow x - y;$
$\{y = X \wedge x = Y\}$

$\{X = n\}$
$X \leftarrow X + 1;$
$\{X - 1 = n\}$
$\{X = n + 1\}$

$\{\}$
$\{5 = 5\}$
$X \leftarrow 5;$
$\{X = 5\}$

Entrées : x un entier tel que $1 \leq x \leq 3$
$\{x = 1 \vee x = 2 \vee x = 3\}$
si $x == 1$ **alors**
  $\{x = 1 \wedge (x = 1 \vee x = 2 \vee x = 3)\}$
  $\{x = 1\}$
  $y \leftarrow x - 1$
  $\{y + 1 = 1\}$
  $\{y = 0\}$
sinon si $x == 2$ **alors**
  $\{x = 2 \wedge (x = 1 \vee x = 2 \vee x = 3)\}$
  $\{x = 2\}$
  $y \leftarrow x - 2$
  $\{y + 2 = 2\}$
  $\{y = 0\}$
sinon
  $\{x \neq 1 \wedge x \neq 2 \wedge (x = 1 \vee x = 2 \vee x = 3)\}$
  $\{x = 3\}$
  $y \leftarrow x - 3$
  $\{y + 3 = 3\}$
  $\{y = 0\}$
fin
$\{(y = 0) \vee (y = 0) \vee (y = 0)\}$
$\{y = 0\}$
retourner y

Entrées : Deux réels X et Y
$\{\}$
si $X \leq Y$ **alors**
  $\{X \leq Y\}$
  $\{X \leq Y \wedge Y = Y\}$
  $MAX \leftarrow Y$
  $\{X \leq Y \wedge MAX = Y\}$
sinon
  $\{X > Y\}$
  $\{X > Y \wedge X = X\}$
  $MAX \leftarrow X$
  $\{X > Y \wedge MAX = X\}$
fin
$\{(X \leq Y \wedge MAX = Y) \vee (X > Y \wedge MAX = X)\}$
$\{MAX = \max(X, Y)\}$
retourner MAX

Entrées : un réel x
Sorties : la valeur absolue de x
$\{x \in \mathbb{R}\}$
si $x < 0$ **alors**
  $\{x \in \mathbb{R} \wedge x < 0\}$
  $\{x \in \mathbb{R} \wedge x < 0 \wedge x = x\}$
  $absx \leftarrow -x$
  $\{x \in \mathbb{R} \wedge x < 0 \wedge -absx = x\}$
  $\{x \in \mathbb{R} \wedge x < 0 \wedge absx = -x\}$
sinon
  $\{x \in \mathbb{R} \wedge x \geq 0\}$
  $\{x \in \mathbb{R} \wedge x \geq 0 \wedge x = x\}$
  $absx \leftarrow x$
  $\{x \in \mathbb{R} \wedge x \geq 0 \wedge absx = x\}$
fin
$\{x \in \mathbb{R} \wedge ((x < 0 \wedge absx = -x) \vee (x \geq 0 \wedge absx = x))\}$
$\{x \in \mathbb{R} \wedge absx = |x|\}$
retourner *absx*

Entrées : un entier n
Sorties : la somme des n premiers entiers
$\{\}$
$\{0 = 0\}$
$i \leftarrow 0;$
$\{i = 0\}$
$\{i = 0 \wedge 0 = 0\}$
$somme \leftarrow 0;$
$\{i = 0 \wedge somme = 0\}$
$\{somme = \sum_{k=0}^{i} k \wedge i - 1 < n\}$
tant que $i < n$ **faire**
  $\{somme = \sum_{k=0}^{i} k \wedge i - 1 < n \wedge i < n\}$
  $i \leftarrow i + 1;$
  $\{somme = \sum_{k=0}^{i-1} k \wedge i - 2 < n \wedge i - 1 < n\}$
  $somme \leftarrow somme + i;$
  $\{somme - i = \sum_{k=0}^{i-1} k \wedge i - 2 < n \wedge i - 1 < n\}$
  $\{somme = \sum_{k=0}^{i} k \wedge i - 1 < n\}$
fin
$\{somme = \sum_{k=0}^{i} k \wedge i - 1 < n \wedge i \geq n\}$
$\{somme = \sum_{k=0}^{i} k \wedge n \leq i < n + 1\}$
$\{somme = \sum_{k=0}^{i} k \wedge i = n\}$
$\{somme = \sum_{k=0}^{n} k\}$
retourner $somme$

Entrées : X et Y deux entiers positifs.
Sorties : R et Q
$\{X \geq 0 \wedge Y \geq 0\}$
$\{X = X \wedge X \geq 0 \wedge Y \geq 0\}$
$R \leftarrow X;$
$\{X = R \wedge R \geq 0 \wedge Y \geq 0\}$
$\{X = R \wedge R \geq 0 \wedge Y \geq 0 \wedge 0 = 0\}$
$Q \leftarrow 0;$
$\{X = R \wedge R \geq 0 \wedge Y \geq 0 \wedge Q = 0\}$
$\{X = R + Q * Y \wedge R \geq 0\}$
tant que $Y \leq R$ **faire**
  $\{X = R + Q * Y \wedge R \geq 0 \wedge Y \leq R\}$
  $R \leftarrow R - Y;$
  $\{X = R + Y + Q * Y \wedge R + Y \geq 0 \wedge Y \leq R + Y\}$
  $\{X = R + (Q + 1) * Y \wedge R \geq 0\}$
  $Q \leftarrow Q + 1;$
  $\{X = R + Q * Y \wedge R \geq 0\}$
fin
$\{X = R + Q * Y \wedge R \geq 0 \wedge Y > R\}$
$\{X = R + Q * Y \wedge 0 \leq R < Y\}$
retourner (R, Q)

VII Document répons élève

$t \leftarrow x;$

$x \leftarrow y;$

$y \leftarrow t;$

$x \leftarrow x + y;$

$y \leftarrow x - y;$

$x \leftarrow x - y;$

$X \leftarrow X + 1;$

$X \leftarrow 5;$

Entrées : x un entier tel que $1 \leq x \leq 3$

si $x == 1$ **alors**

    $y \leftarrow x - 1$

sinon si $x == 2$ **alors**

    $y \leftarrow x - 2$

sinon

    $y \leftarrow x - 3$

fin

retourner y

Entrées : Deux réels X et Y

si $X \leq Y$ **alors**

    $MAX \leftarrow Y$

sinon

    $MAX \leftarrow X$

fin

retourner MAX

Entrées : un réel x
Sorties : la valeur absolue de x

si $x < 0$ **alors**

    $absx \leftarrow -x$

sinon

    $absx \leftarrow x$

fin

retourner *absx*

Entrées : un entier n
Sorties : la somme des n premiers entiers

$i \leftarrow 0$;

$somme \leftarrow 0$;

tant que $i < n$ **faire**

$i \leftarrow i + 1$;

$somme \leftarrow somme + i$;

fin

retourner $somme$

Entrées : X et Y deux entiers positifs.
Sorties : R et Q

$R \leftarrow X$;

$Q \leftarrow 0$;

tant que $Y \leq R$ **faire**

  $R \leftarrow R - Y$;

  $Q \leftarrow Q + 1$;

fin

retourner (R, Q)