

I3 – Algorithmique et programmation
Introduction à l'algorithmique
Cours n°2
Structures de contrôle

Camille Coti
camille.coti@iutv.univ-paris13.fr

IUT de Villetaneuse, département R&T

2011 – 2012

Slides des cours, versions électroniques des polys, TD, TP...
<http://www.lipn.fr/~coti/cours>

Booléen

Booléen

Un booléen est une variable à deux états : **Vrai** ou **Faux**. On représente aussi parfois ces deux états par 1 et 0.

Les opérateurs de comparaison renvoient un booléen. Exemple :

- $1 > 0$ renvoie Vrai
- $1 == 0$ renvoie Faux

Lorsqu'une condition est évaluée, on regarde sa valeur (booléen). Exemple :

```

1 début
2   maVar : Entier
3   maVar ← 0
4   si maVar < 5 alors
5     | maVar ← maVar + 1
6   fin si
7 fin
  
```

- *maVar* est initialisé à 0
- On teste $maVar < 5$
 - Équivalent à tester $0 < 5$
- La condition vaut **Vrai**
- Donc on exécute le bloc d'instruction après le mot-clé **alors**

Un peu d'algèbre de Boole

On peut évaluer des expressions booléennes en utilisant des opérateurs :

Opérateurs logiques

- ET logique : \cdot
- OU logique : $+$
- OU exclusif : \oplus
- Négation : $\bar{\quad}$ ou $!$

Exemples :

- $a \cdot b$
- $a + b$
- $a \oplus b$
- $\overline{a + b} = !(a + b)$
- $a + \bar{b} = a + !b$

Tables de vérité :

\cdot	0	1
0	0	0
1	0	1

$+$	0	1
0	0	1
1	1	1

\oplus	0	1
0	0	1
1	1	0

La **négation** transforme un Vrai en Faux et inversement :

- $\overline{\text{Vrai}} = \text{Faux}$
- $\overline{\text{Faux}} = \text{Vrai}$

Un peu d'algèbre de Boole (suite)

Prenons $a = Vrai$ et $b = Faux$.

Exemples d'expressions booléennes :

- $a + b = Vrai + Faux = Vrai$
- $a + \bar{b} = Vrai + \overline{Faux} = Vrai + Vrai = Vrai$
- $\overline{a \cdot b} = \overline{Vrai \cdot Faux} = \overline{Vrai \cdot Vrai} = \overline{Vrai} = Faux$

Attention aux parenthèses !

- $(a + b) \cdot (c + d)$

Exemples de compositions d'expressions booléennes sur des variables en algorithmique :

- $(a > b) ET (a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ donc l'expression vaut $Faux$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
- $(a > b) OU (a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$

Un peu d'algèbre de Boole (suite)

Prenons $a = Vrai$ et $b = Faux$.

Exemples d'expressions booléennes :

- $a + b = Vrai + Faux = Vrai$
- $a + \bar{b} = Vrai + \overline{Faux} = Vrai + Vrai = Vrai$
- $\overline{a \cdot b} = \overline{Vrai \cdot Faux} = \overline{Vrai \cdot Vrai} = \overline{Vrai} = Faux$

Attention aux parenthèses !

- $(a + b) \cdot (c + d)$

Exemples de compositions d'expressions booléennes sur des variables en algorithmique :

- $(a > b) ET (a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ donc l'expression vaut $Faux$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
- $(a > b) OU (a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$

Un peu d'algèbre de Boole (suite)

Prenons $a = Vrai$ et $b = Faux$.

Exemples d'expressions booléennes :

- $a + b = Vrai + Faux = Vrai$
- $a + \bar{b} = Vrai + \overline{Faux} = Vrai + Vrai = Vrai$
- $\overline{a \cdot b} = \overline{Vrai \cdot Faux} = \overline{Vrai \cdot Vrai} = \overline{Vrai} = Faux$

Attention aux parenthèses !

- $(a + b) \cdot (c + d)$

Exemples de compositions d'expressions booléennes sur des variables en algorithmique :

- $(a > b) ET (a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ donc l'expression vaut $Faux$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
- $(a > b) OU (a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$

Un peu d'algèbre de Boole (suite)

Prenons $a = Vrai$ et $b = Faux$.

Exemples d'expressions booléennes :

- $a + b = Vrai + Faux = Vrai$
- $a + \bar{b} = Vrai + \overline{Faux} = Vrai + Vrai = Vrai$
- $\overline{a \cdot b} = \overline{Vrai \cdot Faux} = \overline{Vrai \cdot Vrai} = \overline{Vrai} = Faux$

Attention aux parenthèses !

- $(a + b) \cdot (c + d)$

Exemples de compositions d'expressions booléennes sur des variables en algorithmique :

- $(a > b) ET (a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ donc l'expression vaut $Faux$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
- $(a > b) OU (a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$

Un peu d'algèbre de Boole (suite)

Prenons $a = Vrai$ et $b = Faux$.

Exemples d'expressions booléennes :

- $a + b = Vrai + Faux = Vrai$
- $a + \bar{b} = Vrai + \overline{Faux} = Vrai + Vrai = Vrai$
- $\overline{a \cdot b} = \overline{Vrai \cdot Faux} = \overline{Vrai \cdot Vrai} = \overline{Vrai} = Faux$

Attention aux parenthèses !

- $(a + b) \cdot (c + d)$

Exemples de compositions d'expressions booléennes sur des variables en algorithmique :

- $(a > b) ET (a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ donc l'expression vaut $Faux$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
- $(a > b) OU (a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$

Un peu d'algèbre de Boole (suite)

Prenons $a = Vrai$ et $b = Faux$.

Exemples d'expressions booléennes :

- $a + b = Vrai + Faux = Vrai$
- $a + \bar{b} = Vrai + \overline{Faux} = Vrai + Vrai = Vrai$
- $\overline{a \cdot b} = \overline{Vrai \cdot Faux} = \overline{Vrai \cdot Vrai} = \overline{Vrai} = Faux$

Attention aux parenthèses !

- $(a + b) \cdot (c + d)$

Exemples de compositions d'expressions booléennes sur des variables en algorithmique :

- $(a > b) ET (a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ donc l'expression vaut $Faux$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
- $(a > b) OU (a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$

Un peu d'algèbre de Boole (suite)

Prenons $a = Vrai$ et $b = Faux$.

Exemples d'expressions booléennes :

- $a + b = Vrai + Faux = Vrai$
- $a + \bar{b} = Vrai + \overline{Faux} = Vrai + Vrai = Vrai$
- $\overline{a \cdot b} = \overline{Vrai \cdot Faux} = \overline{Vrai \cdot Vrai} = \overline{Vrai} = Faux$

Attention aux parenthèses !

- $(a + b) \cdot (c + d)$

Exemples de compositions d'expressions booléennes sur des variables en algorithmique :

- $(a > b)ET(a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ donc l'expression vaut $Faux$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
- $(a > b)OU(a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$

Un peu d'algèbre de Boole (suite)

Prenons $a = Vrai$ et $b = Faux$.

Exemples d'expressions booléennes :

- $a + b = Vrai + Faux = Vrai$
- $a + \bar{b} = Vrai + \overline{Faux} = Vrai + Vrai = Vrai$
- $\overline{a \cdot b} = \overline{Vrai \cdot Faux} = \overline{Vrai \cdot Vrai} = \overline{Vrai} = Faux$

Attention aux parenthèses !

- $(a + b) \cdot (c + d)$

Exemples de compositions d'expressions booléennes sur des variables en algorithmique :

- $(a > b)ET(a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ donc l'expression vaut $Faux$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
- $(a > b)OU(a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$

Un peu d'algèbre de Boole (suite)

Prenons $a = Vrai$ et $b = Faux$.

Exemples d'expressions booléennes :

- $a + b = Vrai + Faux = Vrai$
- $a + \bar{b} = Vrai + \overline{Faux} = Vrai + Vrai = Vrai$
- $\overline{a \cdot b} = \overline{Vrai \cdot Faux} = \overline{Vrai \cdot Vrai} = \overline{Vrai} = Faux$

Attention aux parenthèses !

- $(a + b) \cdot (c + d)$

Exemples de compositions d'expressions booléennes sur des variables en algorithmique :

- $(a > b)ET(a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ donc l'expression vaut $Faux$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
- $(a > b)OU(a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$

Un peu d'algèbre de Boole (suite)

Prenons $a = Vrai$ et $b = Faux$.

Exemples d'expressions booléennes :

- $a + b = Vrai + Faux = Vrai$
- $a + \bar{b} = Vrai + \overline{Faux} = Vrai + Vrai = Vrai$
- $\overline{a \cdot b} = \overline{Vrai \cdot Faux} = \overline{Vrai \cdot Vrai} = \overline{Vrai} = Faux$

Attention aux parenthèses !

- $(a + b) \cdot (c + d)$

Exemples de compositions d'expressions booléennes sur des variables en algorithmique :

- $(a > b)ET(a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ donc l'expression vaut $Faux$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
- $(a > b)OU(a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$

Un peu d'algèbre de Boole (suite)

Prenons $a = Vrai$ et $b = Faux$.

Exemples d'expressions booléennes :

- $a + b = Vrai + Faux = Vrai$
- $a + \bar{b} = Vrai + \overline{Faux} = Vrai + Vrai = Vrai$
- $\overline{a \cdot b} = \overline{Vrai \cdot Faux} = \overline{Vrai \cdot Vrai} = \overline{Vrai} = Faux$

Attention aux parenthèses !

- $(a + b) \cdot (c + d)$

Exemples de compositions d'expressions booléennes sur des variables en algorithmique :

- $(a > b)ET(a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ donc l'expression vaut $Faux$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
- $(a > b)OU(a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$

Un peu d'algèbre de Boole (suite)

Prenons $a = Vrai$ et $b = Faux$.

Exemples d'expressions booléennes :

- $a + b = Vrai + Faux = Vrai$
- $a + \bar{b} = Vrai + \overline{Faux} = Vrai + Vrai = Vrai$
- $\overline{a \cdot b} = \overline{Vrai \cdot Faux} = \overline{Vrai \cdot Vrai} = \overline{Vrai} = Faux$

Attention aux parenthèses !

- $(a + b) \cdot (c + d)$

Exemples de compositions d'expressions booléennes sur des variables en algorithmique :

- $(a > b)ET(a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ donc l'expression vaut $Faux$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
- $(a > b)OU(a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$

Un peu d'algèbre de Boole (suite)

Prenons $a = Vrai$ et $b = Faux$.

Exemples d'expressions booléennes :

- $a + b = Vrai + Faux = Vrai$
- $a + \bar{b} = Vrai + \overline{Faux} = Vrai + Vrai = Vrai$
- $\overline{a \cdot b} = \overline{Vrai \cdot Faux} = \overline{Vrai \cdot Vrai} = \overline{Vrai} = Faux$

Attention aux parenthèses !

- $(a + b) \cdot (c + d)$

Exemples de compositions d'expressions booléennes sur des variables en algorithmique :

- $(a > b)ET(a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ donc l'expression vaut $Faux$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
- $(a > b)OU(a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$

Un peu d'algèbre de Boole (suite)

Prenons $a = Vrai$ et $b = Faux$.

Exemples d'expressions booléennes :

- $a + b = Vrai + Faux = Vrai$
- $a + \bar{b} = Vrai + \overline{Faux} = Vrai + Vrai = Vrai$
- $\overline{a \cdot b} = \overline{Vrai \cdot Faux} = \overline{Vrai \cdot Vrai} = \overline{Vrai} = Faux$

Attention aux parenthèses !

- $(a + b) \cdot (c + d)$

Exemples de compositions d'expressions booléennes sur des variables en algorithmique :

- $(a > b)ET(a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ donc l'expression vaut $Faux$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
- $(a > b)OU(a > 0)$
 - Si $a = 1$ et $b = 2$: $(a > b) = Faux$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$
 - Si $a = 3$ et $b = 2$: $(a > b) = Vrai$ et $(a > 0) = Vrai$ donc l'expression vaut $Vrai$

Les boucles

Condition d'arrêt

La condition d'arrêt d'une boucle est une condition (une expression booléenne) qui détermine le moment où une boucle doit arrêter d'exécuter le bloc d'instructions.

Une boucle sert à **répéter** un bloc d'instructions tant qu'une condition de continuation est satisfaite ou que la condition d'arrêt n'est pas satisfaite.
Exemples d'utilisation :

- Parcours d'un tableau, calcul itératif...

Une boucle utilise un bloc d'instructions : c'est tout le bloc d'instructions correspondant qui est répété.

Il est possible que le bloc d'instructions ne soit pas exécuté du tout (si la condition d'arrêt est déjà satisfaite) ou un nombre infini de fois (souvent un bug).

Boucle **Pour**

On définit un compteur et :

- Une initialisation de ce compteur
 - $i \leftarrow 0$
- Une condition d'arrêt pour sortir de la boucle
 - à 9
- Un pas qui modifie le compteur à **la fin** de chaque itération
 - pas 1

On termine le bloc avec **finpour**

Exemple : algorithme de remplissage d'un tableau de 10 cases.

```
1 début  
2   | tab[10] : Tableau d'entiers  
3   | pour  $i \leftarrow 0$  à 9 pas 1 faire  
4   |   |  $tab[i] \leftarrow 2 * i$   
5   | finpour  
6 fin
```

Boucle **Pour** (suite)

Le **pas** est n'importe quel modificateur sur le compteur : il peut être négatif, non linéaire...

```

1 début
2   | pour  $i \leftarrow 10$  à 0 pas -2 faire
3   |   | afficher(  $i$  )
4   |   finpour
5 fin

```

Affichage par le programme :

10
8
6
4
2
0

```

1 début
2   | pour  $i \leftarrow 1$  à 35 pas *2 faire
3   |   | afficher(  $i$  )
4   |   finpour
5 fin

```

Affichage par le programme :

1
2
4
8
16
32

La boucle s'exécute tant que i est inférieur à 35 : on s'arrête à 32.

Boucle **Tant que... faire**

La boucle *Tant que* exécute un bloc d'instructions tant qu'une condition est vraie : c'est la **condition de boucle**.

- La condition de boucle est définie après le mot-clé **Tant que**
- L'action à effectuer est donnée entre les mot-clés **faire** et **fintq** : on définit un bloc d'instructions qui est répété

Attention aux boucles infinies !

- Il faut que la condition de boucle finisse par être invalidée...

Exemple : algorithme de calcul des puissances de 2 inférieures à 50.

1	début	
2		<i>puissance</i> : Entier
3		<i>puissance</i> ← 1
4		tant que <i>puissance</i> < 50 faire
5		afficher(<i>puissance</i>)
6		<i>puissance</i> ← <i>puissance</i> * 2
7		fintq
8	fin	

Affichage :

1
2
4
8
16
32

Boucle **Tant que... faire** (suite)

```
1 début
2   |  puissance : Entier
3   |  puissance ← 1
4   |  tant que puissance < 50 faire
5   |  |  afficher( puissance )
6   |  |  puissance ← puissance * 2
7   |  fintq
8 fin
```

Détail de l'exécution :

- *puissance* vaut 1
- *puissance* est-il inférieur à 50 ? oui donc on exécute le bloc
- Affichage : 1
- *puissance* vaut 2
- Retour à la ligne 4 : *puissance* est-il inférieur à 50 ? oui donc on exécute le bloc
- Affichage : 2
- *puissance* vaut 4
- ...
- Lorsque *puissance* prend la valeur 64 : la condition ligne 4 n'est plus satisfaite et on sort de la boucle

Équivalence entre les boucles **for** et **tant que ... faire**

On peut écrire une boucle **faire ... tant que** équivalente à une boucle **for** :

- Le compteur est initialisé avant d'entrer dans la boucle **faire ... tant que**
- La condition de boucle est la même que la condition d'arrêt de la boucle **for**
- Le compteur est modifié à la fin du bloc d'instruction exécuté par la boucle **faire ... tant que**

```

1 début
2   | pour  $i \leftarrow 0$  à 9 pas 1 faire
3   |   |  $tab[i] \leftarrow 2 * i$ 
4   | finpour
5 fin
  
```

```

1 début
2   |  $i : entier$ 
3   |  $i \leftarrow 0$ 
4   | tant que  $i < 10$  faire
5   |   |  $tab[i] \leftarrow 2 * i$ 
6   |   |  $i \leftarrow i + 1$ 
7   | fintq
8 fin
  
```

Boucle **Faire ... tant que**

- La boucle **Faire ... tant que** exécute un bloc d'instructions **puis** évalue une condition de boucle
- Le bloc d'instructions est répété si la condition de boucle est satisfaite

```

1 début
2   | puissance : Entier
3   | puissance ← 1
4   | faire
5   |   | afficher( puissance )
6   |   | puissance ← puissance * 2
7   |   | tant que puissance < 20 ;
8   | fin

```

Détail de l'exécution :

- *puissance* vaut 1
- Affichage : 1
- *puissance* vaut 2
- *puissance* est-il inférieur à 20 ? oui
donc on ré-exécute le bloc : retour à la ligne 4
- Affichage : 2
- *puissance* vaut 4
- ...
- Lorsque *puissance* a pris la valeur 32 :
la condition ligne 7 n'est plus satisfaite
et on sort de la boucle

Différence entre les boucles **Faire ... tant que** et **Tant que ... faire**

- La boucle **Faire ... tant que** commence par évaluer la condition de boucle
 - **Puis** elle exécute le bloc d'instructions si la condition de boucle est validée
- La boucle **Tant que ... faire** exécute le bloc d'instructions **puis elle évalue** la condition de boucle

Algorithme d'attente à un stop :

```

1 début
2   vitesse : Entier
3   faire
4     | vitesse ← 0
5   tant que voituresArrivent == Vrai ;
6     vitesse ← 50
7 fin
  
```

Algorithme d'attente à un feu rouge :

```

1 début
2   vitesse : Entier
3   tant que couleurFeu == rouge faire
4     | vitesse ← 0
5   fintq
6   vitesse ← 50
7 fin
  
```

Avec un stop on s'arrête, puis on regarde si on peut avancer. À un feu de circulation, on s'arrête si le feu est rouge ; si le feu n'est pas rouge, on avance.