Raphaël Charrondière
raphael.charrondiere@inria.fr

Correction TD 4: Testons ce code !

Où l'on continue commence à manipuler les cfg.

```
S ← 0;
i ← 0;
n ← taille(Seq);
répéter
    suivant Seq[i] faire
        cas où A faire
            S ← 1 + S;
        fin
        cas où C faire
            S ← 3 + S;
        fin
        cas où T faire
            S ← 4 + S;
        fin
    fin
    i ← 1 + i;
jusqu'à i = n;
si M vaut double alors
    S ← S%2;
sinon
    S ← S%4;
fin
retourner S vaut V;
```

Algorithme 1 : Contrôle d'intégrité

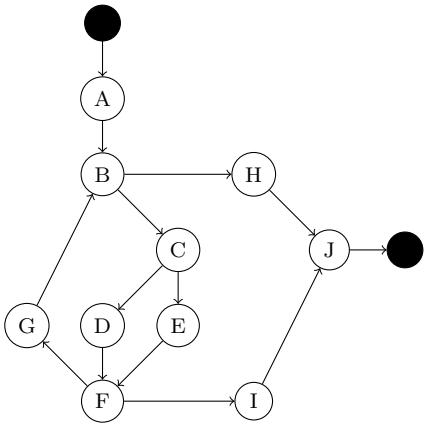


Figure 1 : Le graphe du parti

```
Entrées : l une liste
Output : l triée
n ←longueur(l);
faire
    stop ← vrai;
    pour i de 1 à n − 1 faire
        si l[i] > l[i + 1] alors
            Inverser l[i] et l[i + 1];
            stop ← faux;
        fin
    fin
tant que ¬stop;
retourner l;
```

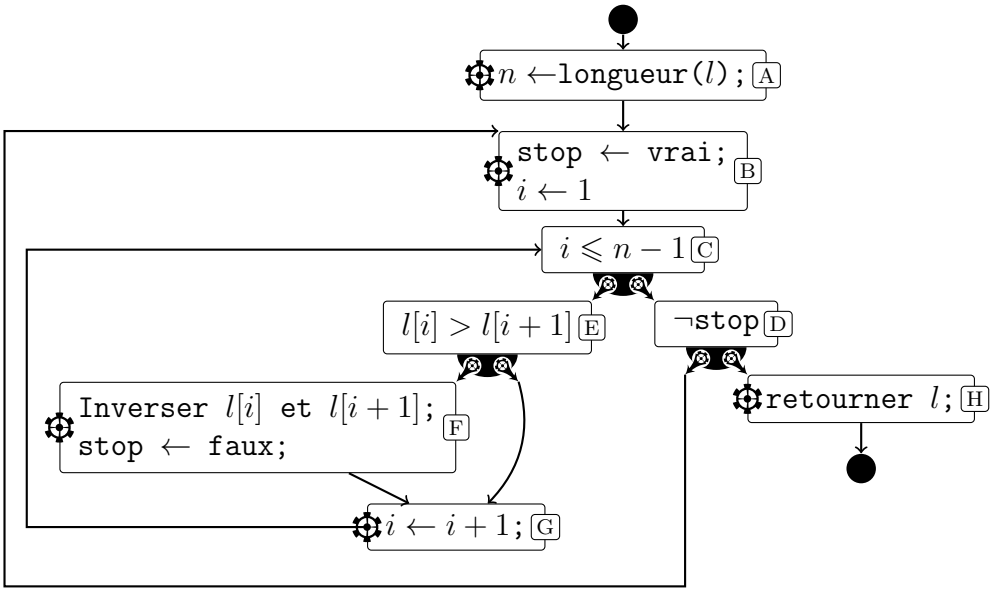
Algorithme 2 : Tri bulle

Exercice 1 Des poissons dans l'eau

1 Donner son graphe de flot de contrôle.

`Correction :`

Raphaël Charrondière
raphael.charrondiere@inria.fr



2 Donner les chemins sous forme algébrique.

`Correction :` AB(C(EF ?GC)*DB)*(C(EF ?GC)*D)H

3 Donner des données de test pour le critère de couverture « tous les nœuds »

`Correction :` Par exemple DT :[l=(2,1,0)], le flot passe alors par le chemin ABCEFGCEFGCDB-CEFGCEGCDBCEGCEGCDH

4 Donner des données de test pour le critère de couverture « tous les arcs »

`Correction :` Le même exemple fonctionne DT :[l=(2,1,0)]

5 Donner des données de test pour le critère de couverture par chemins indépendants.

`Correction :` Il faut d'abord calculer la complexité cyclomatique. Il y a 10 nœuds et 12 arcs, donc elle est de $12-10+2 = 4$. On doit trouver quatre chemins indépendants, et les données de tests correspondantes.

— DT1 :[l=()] on a le chemin 1 : ABCDH
— DT2 :[l=(0,1)] on a le chemin 2 : ABCEGCDH
— DT3 :[l=(1,0)] on a le chemin 3 : ABCEFGCDBCEGCDH
— DT4 :[l=(2,1,0)] on a le chemin 4 : ABCEFGCEFGCDBCEFGCEGCDBCEGCEGCDH

On a 4 chemins, s'ils sont indépendants alors ces 4 DT couvrent le critère de couverture par chemins indépendants vu que la complexité cyclomatique est de 4.

On a le tableau comme défini en cours du nombre de chaque arc pour les différents chemins. On rappelle qu'un chemin est indépendant des autres s'il ne s'écrit pas comme combinaison linéaire des autres.

	1	2	3	4
A→B	1	1	1	1
B→C	1	1	2	3
C→D	1	1	2	3
C→E	0	1	2	6
D→B	0	0	1	2
D→H	1	1	1	1
E→F	0	0	1	3
E→G	0	1	1	3
F→G	0	0	1	3
G→C	0	1	2	6

On commencer par montrer que le chemin 1 est indépendant des 3 autres, cad qu'on résout l'équation $C1 = \alpha_2 C2 + \alpha_3 C3 + \alpha_4 C4$.

— Pour la ligne A→B, l'équation devient E1 : $1 = \alpha_2 + \alpha_3 + \alpha_4$
— Pour la ligne D→B, l'équation devient E2 : $0 = \alpha_3 + 2\alpha_4$

Raphaël Charrondière
raphael.charrondiere@inria.fr

— Pour la ligne F→G, l'équation devient E3 : $0 = \alpha_3 + 3\alpha_4$
— Pour la ligne G→C, l'équation devient E4 : $0 = \alpha_2 + 2\alpha_3 + 6\alpha_4$

On peut combiner E3-E2 pour trouver $\alpha_4 = 0$, puis en réinjectant dans E2 $\alpha_3 = 0$. E1 nous indique alors que $\alpha_2 = 1$. E4 se traduit alors par $0 = 1$, ce qui indique que l'équation **n'a pas** de solution, donc le chemin 1 est indépendant des trois autres.

Montrons maintenant que le chemin 2 est indépendant des chemins 3 et 4 (plus besoin de considérer le chemin 1). On cherche alors à résoudre l'équation $C2 = \beta_3 C3 + \beta_4 C4$.

— Pour la ligne A→B, l'équation devient E1 : $1 = \beta_3 + \beta_4$
— Pour la ligne D→B, l'équation devient E2 : $0 = \beta_3 + 2\beta_4$
— Pour la ligne E→F, l'équation devient E3 : $0 = \beta_3 + 3\beta_4$

E3-E2 nous indique que $\beta_4 = 0$, puis E3 que $\beta_3 = 0$. En réinjectant dans E1 on trouve $1 = 0$, l'équation n'a donc pas de solution, C2 est bien indépendant de C3 et C4.

Finalement il nous reste à montrer que C3 et C4 sont indépendants, c'est à dire qu'ils ne sont pas proportionnel, ne qui n'est clairement pas le cas.

Conclusion : les données de test proposées couvrent bien un ensemble maximal de chemins indépendants

6➤ Donner des données de test pour le critère « toutes-les-définitions »

`Correction :` On peut lister un chemin dr-strict pour chaque définition :
— n dans nœud A : ABC
— stop dans nœud B : BCD
— i dans nœud B : BC
— stop dans nœud F : FGCD
— i dans G : GC

On propose alors les DT : [l=(1,0)] et [l=()]

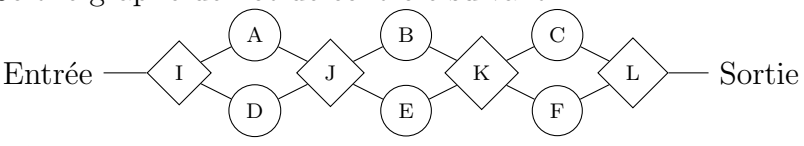
7➤ Donner des données de test pour le critère « tous-les-utilisateurs »

`Correction :` On peut lister un chemin dr-strict pour chaque définition-utilisation accessible :
— n dans nœud A utilisation nœud C : ABC
— stop dans nœud B utilisation nœud D : BCD
— i dans nœud B utilisation nœud C : BC
— i dans nœud B utilisation nœud E : BCE
— i dans nœud B utilisation nœud F : BCEF
— i dans nœud B utilisation nœud G : BCEG
— stop dans nœud F utilisation nœud C : FGCD
— i dans nœud G utilisation nœud C : GC
— i dans nœud G utilisation nœud E : GCE
— i dans nœud G utilisation nœud F : GCEF
— i dans nœud G utilisation nœud G : GCEG

On propose alors les DT : [l=(2,1,0)] et [l=()]

Exercice 2 — Entraînement pour l'indépendance

Soit le graphe de flot de contrôle suivant :



1➤ Donner les chemins du graphe sous forme algébrique.

`Correction :` I(A+D)J(B+E)K(C+F)L

Raphaël Charrondière
raphael.charrondiere@inria.fr

(2) Trouver une base de chemins indépendants. *Indice* : vous trouverez une manière de connaître leur nombre dans le cours.

`Correction :` On peut calculer la complexité cyclomatique égale à 4 (par exemple il y a 4 zones). On propose les chemins C1=IAJBKCL, C2=IDJBKCL, C3=IAJEKCL, C4=IAJBKFL. Il faut juste montrer que ces chemins sont indépendant.

— C2 est indépendant des autres car c'est le seul à contenir l'arc ID
— C3 est indépendant des autres car c'est le seul à contenir l'arc JE
— C4 est indépendant des autres car c'est le seul à contenir l'arc KF

(Comme tout les autres sont indépendants, C1 est indépendant des autres)

(3) Exprimer les autres chemins comme une combinaison des chemins de la base.

`Correction :` Les autres chemins sont

— C5 : IDJEKC = C2+C3-C1
— C6 : IDJBKF = C2+C4-C1
— C7 : IAJEKF = C3+C4-C1
— C8 : IDJEKF = C2+C3+C4-2C1

Exercice 3 Mais non, je ne vois pas de problème !

Les tests sont importants, ils ne sont pas suffisants.

(1) Donner un exemple différent de celui du cours où le critère « tous les arcs » et même le critère « condition-décision multiple » sont couverts, mais où il y a tout de même un bug. Nous avons donné deux situations où cela pouvait se produire en cours, les rappeler.

`Correction :` On a donné deux situations suivantes :

— s'il y a deux conditions indépendantes et qu'il y a tout de même une dépendance entre les commandes. Par exemple en C `if(a>1)data=malloc(a);else a=20;...if(b>1)free(data);` avec les DT :[a=30,b=10] et [a=0,b=0], on couvre tout les nœuds/arcs mais on ne détecte pas que l'on peut faire un `free` sans `malloc`.
— S'il y a une boucle, avec une initialisation dans la boucle elle-même. Exemple pour le calcul de la factorielle `int res;for(i=0,i<n,i++) if(i==0)res=1; else res=res*i;return res;`, la variable `res` n'est pas initialisée, et si `n` vaut 0 il y a une erreur. Si on a une DT $[n > 1]$ par exemple DT$[n = 2]$ on a la couverture par nœud/arc.

(2) Reprendre l'exemple et tester le critère « tous les chemins indépendants ».

`Correction :` Pour le premier exemple : on va forcément avoir l'une des deux situations : appel malloc sans free ou l'inverse. Dans le second exemple on ajoutera la DT[n=0] pour obtenir la couverture par chemin indépendant.

(3) On veut créer un exemple où le critère « tous les chemins indépendants » est insuffisant. Le programme en question devra vérifier l'intégrité d'un tableau Seq. Celui-ci contient des éléments valant G, A, T ou C.

Pour cela on attribuera une valeur à chaque élément : G vaut 0, A vaut 1, T vaut 2 et C vaut 3. Ensuite on calculera la somme S de toutes ces valeurs. Puis on comparera cette valeur à une clé (V,M) fournie en entrée. V est un nombre entre 0 et 4, M vaut 'double' ou 'quadruple'.

Si M vaut 'double', le programme vérifie que S vaut V modulo 2, si M vaut 'quadruple', le programme vérifie que S vaut V modulo 4.

— Lire l'algorithme 1 et repérer l'erreur
— Construire le graphe de flot de contrôle correspondant.
— Calculer le nombre de chemins indépendants.
— On propose les données de tests suivantes :
  — {Seq=[G], V=0, M=double}

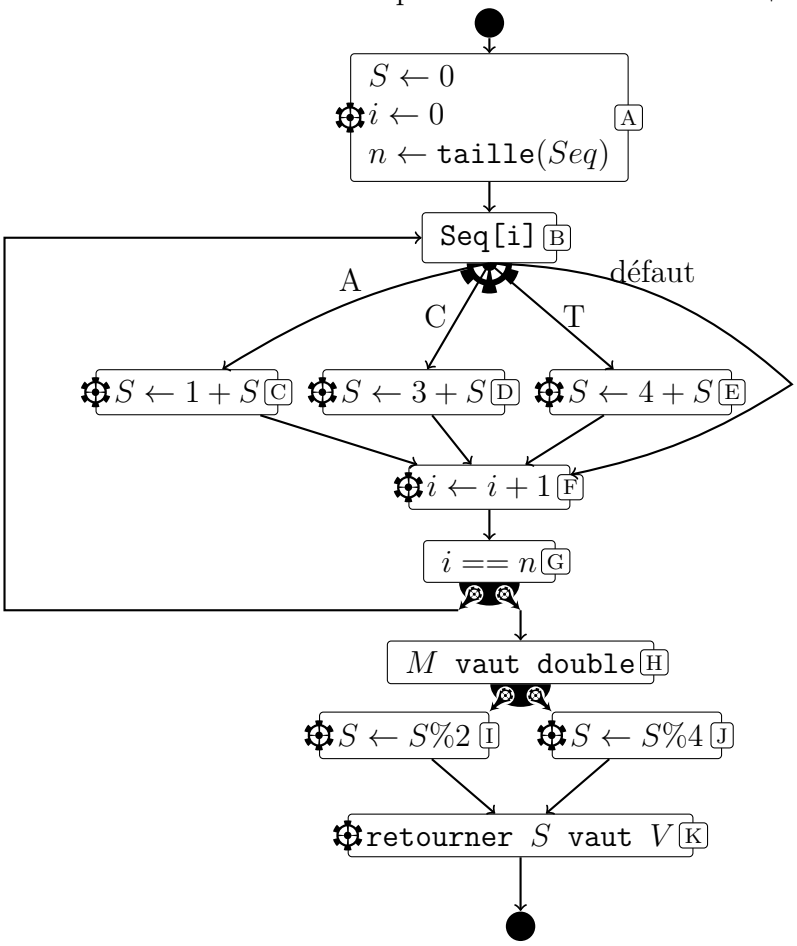
Raphaël Charrondière
raphael.charrondiere@inria.fr

— {Seq=[A], V=1, M=double}
— {Seq=[T], V=0, M=double}
— {Seq=[C], V=1, M=double}
— {Seq=[G], V=0, M=quadruple}
— {Seq=[A,C], V=0, M=double}

Couvre-t-on une base de chemins indépendants ? Repère-t-on l'erreur ?

— Conclure

`Correction :` L'erreur est pour T il est écrit "$S \leftarrow 4 + S$" au lieu de $S \leftarrow 2 + S$.



La complexité cyclomatique vaut 6 Les DT correspondent à une base de chemins indépendants mais on ne voit pas d'erreur. On a donc bien un cas où couverture par chemins indépendants ne suffit pas à détecter une erreur.

Exercice 4 Recyclage

On reprend l'algorithme 1 de l'exo 3, question 3. Vous avez construit son GFC.

1 Donner des données de test pour couvrir le critère « toutes-les-définitions »

`Correction :` On liste les définitions :

— S au nœud A,C,D,E,I et J
— i au nœud A,F
— n au nœud A

Dans ce code toute variable définie est utilisée. On peut donner les DT [Seq=ACT, M=double] et [Seq=G,M=quac

2 Donner des données de test pour couvrir le critère « tous-les-utilisateurs »

`Correction :` On liste les définitions/utilisations accessibles par les triplets (variable/nœud définition/nœud

Raphaël Charrondière
raphael.charrondiere@inria.fr

utilisation) : (i/A/F) (i/F/G) (n/A/G) (S/A/C) (S/A/D) (S/A/E) (S/A/I) (S/A/J) (S/C/C) (S/C/D) (S/C/E) (S/C/I) (S/C/J) (S/D/C) (S/D/D) (S/D/E) (S/D/I) (S/D/J) (S/E/C) (S/E/D) (S/E/E) (S/E/I) (S/E/J).

On trouve alors les DT suivantes :
[Seq=[],M=Double], [Seq=[],M=Quadruple], [Seq=[A,A],M=Double], [Seq=[C,C,A],M=Quadruple], [Seq=[T,A,C],M=Double], [Seq=[T,T,C],M=Quadruple], [Seq=[A,T],M=Double], [Seq=[C,T],M=Quadruple],

---

Exercice 5 Parti indépendantiste

Cette partie porte sur les chemins indépendants.

1 Donnez une base de chemin indépendants du graphe en figure 1. Justifiez votre réponse.

`Correction :` On peut calculer la complexité cyclomatique pour savoir combien une base a de chemins indépendants. Ici il y a 12 nœuds et 14 arcs, donc elle est de 4. Proposons les chemins ① =ABHJ, ② = ABCDFIJ, ③ = ABCEFIJ et ④ = ABCDFGBHJ.

Montrons que ces chemins sont indépendants : le chemin ④ est le seul à contenir l'arc FG, il est donc indépendant des trois autres. Parmis les trois restants seul ③ passe par l'arc CE, et ② par CD, donc ces 2 chemins sont aussi indépendants des autres. Finalement il ne reste qu'un seul chemin, donc il est forcément indépendant.

2

Soit un graphe $\mathcal{G} = (N, A)$ où N est l'ensemble de ses $n = \#N$ nœuds et A l'ensemble de ses $a = \#A$ arcs. On rappelle que la complexité cyclomatique se définit par $v(\mathcal{G}) = 2 + a - n$.

Montrez qu'en notant ϕ le nombre de nœuds de décision, alors $v(\mathcal{G}) = \phi + 1$.

`Correction :` Les arcs sont orientés, pour les compter il suffit de compter les arcs sortants pour chaque nœud. Pour un nœud d'entrée il y a un arc sortant, pour un nœud de sortie il n'y a pas d'arc sortant, pour un nœud de commande il y a un arc sortant, pour un nœud de décision il y a deux arcs sortants. Si l'on note ψ le nombre de nœuds de commandes, étant donné qu'il y a un unique nœud d'entrée et un unique nœud de sortie on a $n = \phi + \psi + 1 + 1$, le nombre d'arc est $a = \phi + 2\psi + 1$. On peut alors calculer $v(\mathcal{G}) = 2 + a - n = 2 + \psi + 2\phi + 1 - \phi - \psi - 1 - 1 = \phi + 1$
