Université de Yaoundé I | Faculté des Sciences
The University of Yaounde I | Faculty of Sciences

Département d'Informatique

Compilation : Fiche de TD N°1

Année académique 2023-2024

Etienne Kouokam

EXERCICE 0 **[Erreurs de compilation.]**

On considère les programmes suivants :

```
class A {
int x?;
public static void main(){
}
}
```

```
class B {
int x;
public static void main(){
String s = "bonjour;
System.out.println(s);
}
}
```

```
class C {
public static void main(){
int y=3;
System.out.println(x+y);
}
}
```

```
class D {
static void m(int x){
System.out.println(x+1);
}
public static void main(){
int y=3;
m(y=y+4);
m(y==y+4);
}
}
```

```
class E {
static int m(int x){
if (x>0) return 1;
else
if (x<=0) return 2;
else return 3;
return 4;
}
public static void main(){
m(42);
}
}
```

```
class F {
static int m(int x){
return (x+1);
}
public static void main(){
int final=42;
m(final+1);
}
}
```

```
class G {
public static void main(){
short s = 35000;
System.out.println(s);
}
}
```

```
class H {
static int m(int x, boolean b,
double f){
return b?x+(int)f:3;
}
public static void main(){
m(4.0,true,3.14);
}
}
```

```
class I {
static int m(int x){
final int y=x+1;
y++;
return y;
}
public static void main(){
m(4);
}
}
```

0.1 Dire si les programmes du tableau ci-dessus sont corrects. Quand un programme ne l'est pas, indiquer à quel moment de la compilation l'erreur est détectée.

0.2 Dire à quelle phase de la compilation on peut détecter les erreurs suivantes ?
a. Identificateur mal formé : 12K3 en C ? **b.** Conflit de type sin('a') **c.** Instruction non atteignable **d.** Variable non déclarée **e.** Commentaire non fermé **f.** Parenthèse non fermée **g.** BEGIN non fermé **h.** Mot clé utilisé comme identificateur **i.** Non conformité entre le nombre de paramètres de définition et d'appel d'une procédure. **j.** Tentative de modifier une constante

EXERCICE 1 **[Notion de longueur.]**

1.1 Soit $\Sigma = \{a, b\}$ et les langages $L = \{abb, b, a, \epsilon\}$ et $L' = \{ba, baa\}$.
Calculer les langages suivants : $L \cup L'$, $L \cap L'$, LL', $L'L$, L^0, L^2.

1.2 Étant donné un alphabet Σ, définir par induction sur la structure des mots la fonction $|w|_a$ qui calcule le nombre d'occurrences d'une lettre $a \in \Sigma$ dans le mot $w \in \Sigma^*$.

1.3 Vérifier que : $|u.v| = |u| + |v|$; $|u.v|_a = |u|_a + |v|_a$.

EXERCICE 2 **[Propriétés de l'itération.]**

2.1 Terminer la démonstration des propriétés 8) de l'itération, c'est-à-dire, montrer que, quels que soient $L, M \subseteq \Sigma^*$:

1. $(L + M)^* \subseteq (L^* + M^*)^*$
2. $(L^* + M^*)^* \subseteq (L^*M^*)^*$
3. $(L^*M^*)^* \subseteq L^*(ML^*)^*$
4. $L^*(ML^*)^* \subseteq (L + M)^*$

2.2 Montrer que, quels que soient $L, M \subseteq \Sigma^*$:

1. $(L^*M)^* = \epsilon + (L + M)^*M$
2. $(LM^*)^* = \epsilon + L(L + M)^*$

EXERCICE 3 **[Facteurs gauches.]**

L'ensemble $fg(u)$ des facteurs gauches d'un mot $u \in \Sigma^*$ se définit par la récurrence suivante : $fg(\epsilon) = \epsilon \quad fg(ux) = fg(u) + ux \; \forall u \in \Sigma^*$ et tout $x \in \Sigma$.

3.1 Vérifier que cette définition est bien la bonne, c'est-à-dire que

$$(v \in fg(u)) \Leftrightarrow (\exists w \in \Sigma^* u = vw)$$

3.2 Vérifier que $fg(uv) = fg(u) + ufg(v) \; \forall u \in \Sigma^*$ et tout $v \in \Sigma^*$

c. Exprimer $fg(LM)$ et $fg(L^*)$ pour $L, M \subseteq \Sigma^*$ quelconques.

EXERCICE 4 **[Facteurs droits.]**

En mettant la définition de l'ensemble $fg(u)$ des facteurs gauches du mot $u \in \Sigma^*$ devant un miroir on peut obtenir facilement une définition de l'ensemble $fd(u)$ de ses facteurs droits, par une récurrence basée sur l'ajout des lettres à gauche : $fd(\epsilon) = \epsilon \quad fd(xu) = xu + fd(u) \; \forall u \in \Sigma^*$ et tout $x \in \Sigma$.
Si l'on s'obstine à vouloir ajouter les lettres à droite, on est conduit à poser la définition par récurrence suivante :

$$fd(\epsilon) = \epsilon \quad fd(ux) = \epsilon + fd(u)x \; \forall u \in \Sigma^* \text{ et tout } x \in \Sigma$$

4.1 Vérifier que cette définition est bien la bonne, c'est-à-dire que

$$(v \in fg(u)) \Leftrightarrow (\exists w \in \Sigma^* u = wv)$$

4.2 Vérifier que $fd(uv) = fd(v) + fd(u)v \; \forall u \in \Sigma^*$ et tout $v \in \Sigma^*$

4.3 Exprimer $fd(LM)$ et $fd(L^*)$ pour $L, M \subseteq \Sigma^*$ quelconques.

EXERCICE 5 **[Facteurs.]**

L'ensemble $fact(u)$ des facteurs du mot $u \in \Sigma^*$ peut se définir par récurrence de la façon suivante, si l'on ajoute des lettres "Ã droite et à gauche" :
$fact(\epsilon) = \epsilon \quad fact(x) = \epsilon + x$ pour tout $x \in \Sigma$,
$fact(xuy) = fact(xu) + fact(uy) + xuy \; \forall x, y \in \Sigma$ et tout $u \in \Sigma^*$. Si l'on s'obstine à vouloir ajouter les lettres à droite, on est conduit à poser la définition par récurrence suivante :

$$fact(\epsilon) = \epsilon \quad fact(ux) = fact(u) + fd(u)x \; \forall u \in \Sigma^* \text{ et tout } x \in \Sigma$$

5.1 Vérifier que cette définition est bien la bonne, c'est-à-dire que

$$(v \in fact(u)) \Leftrightarrow \left(\exists w \in \Sigma^* u = wvw'\right)$$

5.2 Vérifier que $fact(uv) = fact(u) + fd(u)fg(v) + fact(v) \; \forall u \in \Sigma^*$ et tout $v \in \Sigma^*$

5.3 Exprimer $fd(LM)$ et $fd(L^*)$ pour $L, M \subseteq \Sigma^*$ quelconques.

EXERCICE 6 **[Application.]**

Soient a et b des symboles distincts l'un de l'autre.
Calculer $fg(L_i)$, $fd(L_i)$ et $fact(L_i)$ ($1 \leq i \leq 6$) dans chacun des cas suivants :

$L_1 = a^*b^* = \{a^m b^n \mid 0 \leq n, m\}$ $\qquad L_4 = \{a^m b^n \mid 0 \leq m < n\}$
$L_2 = \{a^m b^m \mid 0 \leq m\}$ $\qquad L_5 = \{a^m b^n \mid 0 \leq n \leq m\}$
$L_3 = \{a^m b^n \mid 0 \leq m \leq n\}$ $\qquad L_6 = \{a^m b^n \mid 0 \leq n < m\}$

EXERCICE 7 **[Système d'équations linéaires en langages.]**

Résoudre les Systèmes d'équations suivants :

$$\left\{\begin{array}{lcl} X_0 & = & bX_0 + aX_1 \\ X_1 & = & aX_1 + bX_3 \\ X_2 & = & aX_1 + bX_3 + \epsilon \\ X_3 & = & bX_1 + aX_3 \end{array}\right. \quad \left\{\begin{array}{lcl} X_0 & = & aX_1 + nX_2 \\ X_1 & = & aX_1 + nX_3 \\ X_2 & = & aX_4 + \epsilon \\ X_3 & = & X_3 + \epsilon \\ X_4 & = & aX_4 + \epsilon \end{array}\right. \quad \left\{\begin{array}{lcl} X_1 & = & bX_2 + aX_3 \\ X_2 & = & bX_1 + aX_4 \\ X_3 & = & \epsilon + aX_4 + bX_2 \\ X_4 & = & \epsilon + (a+b)X_4 \end{array}\right. \quad \left\{\begin{array}{lcl} S & = & 0 + X_0 \\ X_0 & = & 0X_0 + 1X_1 \\ X_1 & = & 0X_2 + 1X_0 + 1 \\ X_2 & = & 0X_1 + 1X_2 \end{array}\right.$$

Le dernier Système d'équations correspondant au langages des multiples de 3 écrits dans l'ordre inverse. Pourriez-vous en déduire un système d'équations du même langage dans l'ordre normal ?

EXERCICE 8 **[Mélange de mots.]**

Cette opération consiste à insérer de nouvelles occurrences de caractères à un mot.

8.1 Mélange de deux mots. Soit $mel(u, v)$ l'ensemble des mélanges de $u \in A^*$. et $v \in A^*$ défini par la (double) récurrence suivante :

$$mel(u, \epsilon) = u \;\; mel(\epsilon, v) = v \;\; mel(ux, vy) = mel(u, vy)x + mel(ux, v)y$$

1. Intuitivement, $w \in mel(u, v)$ est un mot de longueur $|u| + |v|$ obtenu en faisant "glisser" les caractères de v dans u, en respectant leur ordre relatif. Pour s'en persuader, calculer $mel(abc, def)$.
2. Vérifier que $mel(u, v) = mel(v, u)$.

8.2 Mélange de deux langages. L'application $mel : A^* \times A^* \to \mathcal{P}(\mathcal{A}^*)$ se prolonge aux langages par :

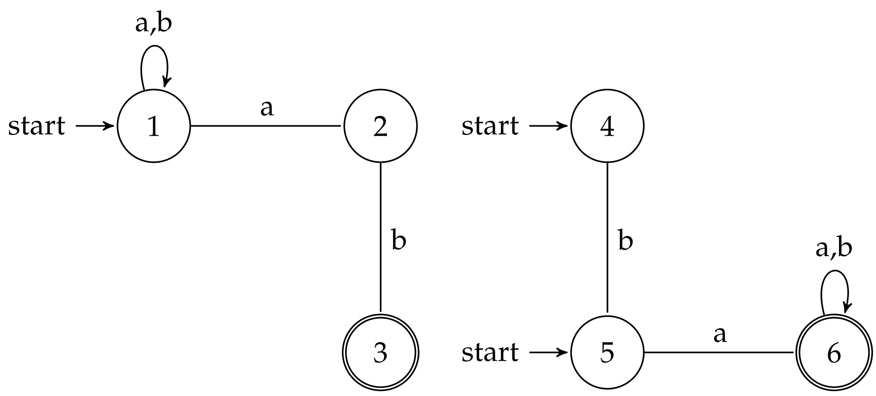
$$mel(L, M) = \sum_{(v,w) \in L \times M} mel(v, w)$$

c'est-à-dire : $u \in mel(L, M)$ ssi il existe $v \in L$ et $w \in M$ tels que $u \in mel(v, w)$.
Calculer $mel(a^*, b^*)$ et $mel(ab, a^*b^*)$.

EXERCICE 9 **[Automates non-déterministes et déterminisation]**

9.1 Construire un automate non déterministe reconnaissant tous les mots sur $\{a, b, c\}$ qui finissent par aba. Déterminiser l'automate obtenu.

9.2 Déterminiser l'automate de la figure ci-dessous :



9.3 Soit $\Sigma = \{a, b\}$ un alphabet. Donner des automates non-déterministes reconnaissant les langages :

1. $L_1 = \{w \in \Sigma^* \mid$ commence / ab, termine / bb et contient 3 occurrences successives de a$\}$
2. $L_1 = \{w \in \Sigma^* \mid |w|_a \equiv 1(3)\}$
3. $L_2 = \{w \in \Sigma^* \mid |w|_a \equiv 0(2)\ et\ |w|_b \equiv 0(2)\}$
4. $L_3 = \{a^p b^q\ |p \geq q\ et\ q \leq 5\}$
5. $L_4 = \{w \in \Sigma^* \mid$ contient le mot abaaabab$\}$

9.4 Déterminiser chacun des automates obtenus à la question précédente

EXERCICE 10 **[Les nombres vus comme des mots sur $\{0, 1\}$.]**

Un nombre ne commence jamais par un 0, sauf le nombre 0.
Donner des automates déterministes complets reconnaissant les codages en binaire :

10.1 Des entiers impairs.

10.2 Des puissances de 2. **10.3** Des entiers de la forme $4^n + 3$ pour $n \geq 0$.

10.3 Des sommes de deux puissances de 4.

10.4 Des successeurs des multiples de 3

EXERCICE 11 [Langage des commentaires.]

Dans notre langage de programmation, les commentaires ont la forme : $/ * w * /$, où le commentaire proprement dit w ne peut pas contenir le facteur $*/$, sauf si il est immédiatement précédé du caractère d'échappement $\%$. On se restreindra à l'alphabet $\{/, *, \%, c\}$, c représentant tous les autres caractères.
Donner un automate déterministe reconnaissant le langage des commentaires.

EXERCICE 12 [langages reconnus par des automates.]

On considère les automates représentés aux figures 1 et 2.

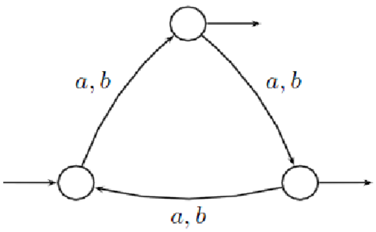


FIGURE 1 – Automate A_3

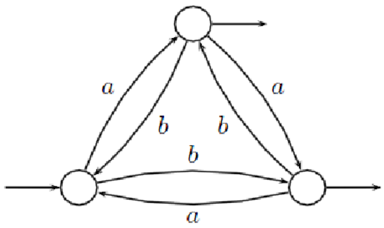


FIGURE 2 – Automate A_4

12.1 Les mots abab, ababa et ababab sont-ils reconnus par l'automate A_3 ?

12.2 Décrire le langage reconnu par l'automate A_3.

12.3 Les mots a^4b^3, a^4b^2 et a^4b sont-ils reconnus par l'automate A_4 ?

12.4 Décrire le langage reconnu par l'automate A_4.

EXERCICE 13 [Digicode]

On veut écrire 2 automates déterministes qui reconnaissent l'entrée du "mot de passe" d'un digicode. Il n'y a que des chiffres possibles en entrée et le code est **11654**.

13.1 Construire un automate qui lit un code de taille 5, l'accepte si c'est le bon, refuse sinon, et permet ensuite de retenter sa chance.

13.2 Construire un automate qui arrive dans un état final pour toute séquence tapée qui finit par le bon code.

EXERCICE 14 [Les automates savent-ils compter ?]

On veut réaliser un changeur de petite monnaie au moyen d'un automate. L'alphabet sera constitué des pièces jaunes grises (5, 10 et 25 francs CFA).

14.1 Construire un automate qui reconnaît toutes les suites de pièces jaunes dont la somme vaut 100 FCFA.

14.2 Construire l'automate qui reconnait les suites dont la somme vaut 100 FCFA et qui contiennent une pièce de 25 FCFA.

14.3 Construire un automate qui reconnaît toutes les suites de pièces jaunes dont la somme vaut 60 FCFA et comportant au moins autant de pièces de 5 que de 10 FCFA.

14.4 Supposons qu'il existe un automate reconnaissant toutes les suites de pièces (peu importe la somme totale) comportant au moins autant de pièces de 5 que de 10. Cet automate reconnaît en particulier la suite commençant par autant de pièces de 10 qu'il y a d'états dans l'automate, puis deux fois plus de pièces de 10 que de 5. Que peut-on en déduire ?

EXERCICE 15 [Le décompte des points au tennis.]

Au tennis, au cours d'un jeu, le score d'un joueur vaut successivement 0, 15, 30 puis 40 au fur et à mesure qu'il marque des points. Si un seul joueur est à 40, il lui suffit de remporter encore un point pour remporter le jeu. En revanche, si les deux sont à 40 (cas dit d'égalité), le joueur qui remporte le point suivant ne gagne qu'un avantage. Il peut alors remporter le jeu s'il gagne également le point suivant ; s'il le perd, les joueurs reviennent à égalité.
Construire un automate représentant l'évolution des scores au cours d'un jeu. Les lettres a et b représenteront un point gagné par le joueur A ou par le joueur B.

EXERCICE 16 [Construction d'automates.]

Donner des automates reconnaissant les langages suivants :

16.1 $L_1 = \{u \in A^* \mid$ toute occurrence de b dans u est immédiatement suivie d'au moins deux de a$\}$

16.2 $L_2 = \{u \in A^* \mid$ u ne contient pas deux a successifs$\}$

16.3 $L_3 = \{u \in A^* \mid$ le nombre d'occurrences de a dans u est pair$\}$

16.4 $L_4 = \{u \in A^* \mid$ les blocs de a dans u sont alternativement de longueur paire et impaire$\}$

EXERCICE 17 [Union, intersection, complémentaire ...]

Soit $\Sigma = \{a, b\}$ et soient deux langages

$$L_1 = \{u \in \Sigma^* \mid |u| \equiv 0 \, mod \, 3\} \; et$$

$$L_2 = \{u \in \Sigma^* \, | unecontientpaslefacteura^2\}$$

En utilisant les constructions vues en cours, construire les automates reconnaissant les langages suivants : $L_1 \cap L_2, L_1 \cup L_2$ *et* $\overline{L_1 \cap L_2}$?

EXERCICE 18 [Intersection de langages et automate produit.]

On considère les automates A, B et C de la figure 3 ci-dessous.

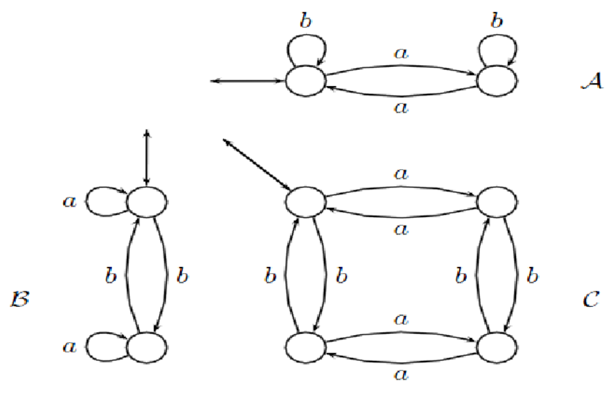


FIGURE 3 – Les automates A, B et C

18.1 Décrire les langages reconnus par les automates A et B.

18.2 Après avoir lu un nombre pair de b, dans quels états peut se trouver l'automate C ?

18.3 Décrire le langage reconnu par l'automate C et conclure.

EXERCICE 19 [Des expressions régulières aux automates.]

Soient donnés les langages et/ou expressions régulières suivants :

$E_1 = (a + b)^*(abb + \epsilon)$ $\quad E_4 = (a + b)^*ba(a + b)^*$ $\quad E_6 = a^* + a^*ba^* + a^*ba^*ba^*b(a + b)^*$

$E_3 = (ab^*a)^*$ $\quad E_5 = a(a + b + c)^*bc$ $\quad E_7 = a^*ba^*ba^*$

Pour chacun de ces cas :

19.1 Construire l'automate de Thompson "pur" correspondant

19.2 Contruire l'automate obtenu après application de l'algorithme de Glushkov

19.3 Transformer les différents automaates ainsi obtenus en AFD s'il y a lieu

19.4 Appliquer l'algorithme de minimisation du cours à l'AFD ainsi obtenu

19.5 Produire l'automate canonique correspondant

19.6 Caractériser le langage décrit par l'expression régulière

EXERCICE 20 [Lemme de pompage ou Myhill Nerode ?]

Les langages suivants sont-ils réguliers ? Justifier à chaque fois.

20.1 $L_1 = \{0^{2n}/n \geq 1\}$

20.2 $L_2 = \{0^{2^n}/n \geq 1\}$

20.3 $L_3 = \{0^n1^n/n \geq 1\}$

20.4 $L_4(n) = \{x \in \{0, 1\}^*/n_0(x) \equiv n_1(x)\ [n]\}$ où $n_0(x)$ (resp. $n_1(x)$) est égal au nombre de 0 (resp. de 1) dans l'écriture de x.

20.5 $L_5 = \{x \in \{0, 1\}^*/$x n'a pas 3 zéros consécutifs$\}$

20.6 $L_6 = \{0^n/n\ premier\}$

EXERCICE 21 [Encore des Automates .]

Pour chacun des automates produits dans les exercices précédents, donner l'automate minimal et l'automate canonique correspondants.

21.1 Considérons l'alphabet : $A = \{the, old, man, men, is, are, here, and\}$

- **a)** Construire un automate sur A qui accepte le langage : {the man is here , the men are here}
- **b)** Idem pour : {the man is here , the men are here , the old man is here ,
  the old men are here , the old old man is here , the old old men are here , . . .}
- **c)** Construire l'automate qui accepte toutes les phrases de b), plus celles obtenues par la conjonction and.

21.2 On considère $A = \{a, b\}$

- **a)** Construire un automate qui accepte toute suite qui ne contient que des a (pas la suite vide).
- **b)** Construire un automate qui accepte toute suite qui contient un nombre impair de a, suivi d'un nombre arbitraire (éventuellement nul) de b.
- **c)** Construire un automate qui accepte à la fois les phrases acceptées par l'automate pour a) et celles acceptées par l'automate pour b).

Bon Courage ! ! !