Théorie des langages et de la compilation
Travaux pratiques

Sébastien COLLETTE Gilles GEERAERTS

Université Libre de Bruxelles

Coordonnées

- Gilles GEERAERTS, Tél. 02-650.55.96
  E-mail : gigeerae@ulb.ac.be
  Bureau : 2N8.211

Où trouver les documents

- La page du cours:
  `http://www.ulb.ac.be/di/ssd/tmassart/Compil`
- La page des t.p.:
  `http://www.ulb.ac.be/di/ssd/ggeeraer`

Langage régulier

Soit un alphabet Σ (fini).

- $\emptyset$ est un langage régulier ;
- $\{\varepsilon\}$ est un langage régulier ;
- Pour tout $a \in \Sigma$, $\{a\}$ est un langage régulier ;
- Si L_1 et L_2 sont réguliers, alors $L_1 \cdot L_2 = \{w_1w_2 \mid w_1 \in L_1 \wedge w_2 \in L_2\}$ est régulier ;

Langage régulier – suite

- Si L est régulier, alors $L^* = \{\varepsilon\} \cup \{www\cdots w \mid w \in L\}$ est régulier ;
- Si L_1 et L_2 sont réguliers, alors $L_1 \cup L_2$ est régulier.

Automate fini

$M = \langle Q, \Sigma, \delta, q_0, F\rangle$ où

- Q ensemble fini des états;
- Σ alphabet des symboles à l'entrée;
- δ fonction de transition;
- q_0 état initial;
- $F \subseteq Q$ ensemble des états accepteurs.

M est fini *déterministe* si la fonction $\delta : Q \times \Sigma \to Q$ est une fonction *totale*.

Exercice 1

Démontrez, à l'aide de la définition inductive des langages réguliers, que les deux langages suivants sont réguliers (l'alphabet considéré est $\Sigma = \{0, 1\}$):

1. L'ensemble des mots composés d'un nombre arbitraire de 1, suivis de 01, suivis d'un nombre arbitraire de 0.
2. L'ensemble des nombres binaires impairs.

Exercice 2

1. Démontrez que tout langage fini est régulier.
2. Le langage $L = \{0^n1^n \mid n = 0, 1, 2, \ldots\}$ est-il régulier ? Expliquez.

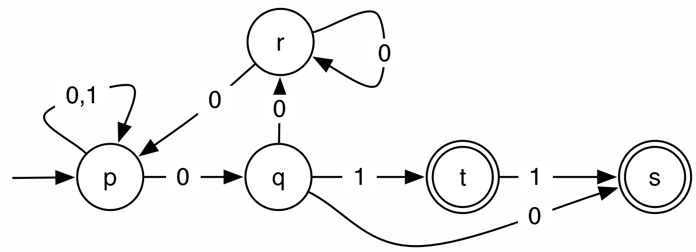
Exercice 3

Donnez un automate non déterministe qui accepte chacun des langages suivants (définis sur l'alphabet $\Sigma = \{0, 1\}$):

1. Toutes les chaînes qui se terminent par 00.
2. Toutes les chaînes dont le 10ème symbole, compté à partir de la fin de la chaîne, est un 1.
3. Ensemble de toutes les chaînes dans lesquelles chaque paire de 0 apparaît devant une paire de 1.
4. Ensemble de toutes les chaînes ne contenant pas 101.
5. Tous les nombres binaires divisibles par 4.

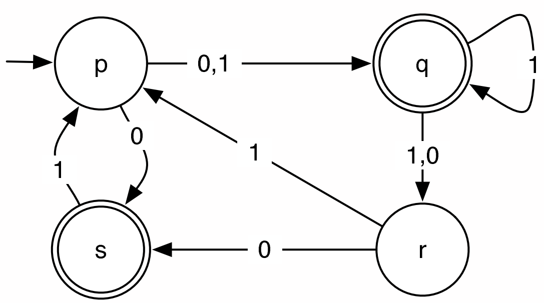
Exercice 4 – 1

Déterminisez:



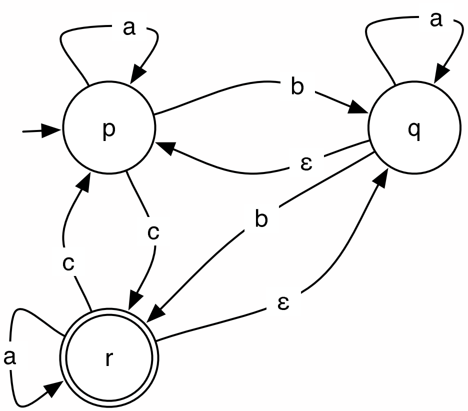
Exercice 4 – 2

Déterminisez:



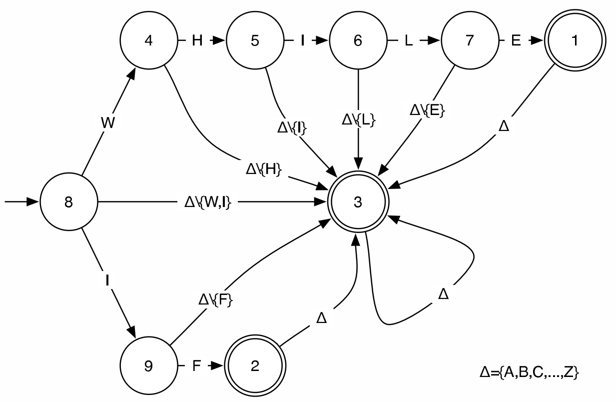
Exercice 4 – 3

Déterminisez:



Exercice 5

Écrivez une fonction C qui implémente cet automate et renvoie le numéro d'état accepteur.



Exercice 1.1 – correction

- $1 \in \Sigma$ et $0 \in \Sigma$. Donc $\{1\}$ et $\{0\}$ sont des langages réguliers.
- La fermeture de Kleene d'un langage régulier est un langage régulier. Donc $\{1\}^*$ et $\{0\}^*$ sont des langages réguliers.
- La concaténation de langages réguliers est un langage régulier. Donc $\{1\}^* \cdot \{0\} \cdot \{1\} \cdot \{0\}^*$ est un langage régulier.

Exercice 1.2 – correction

- Un nombre binaire impair se termine nécessairement par 1.
- $\{1\}$ et $\{0\}$ sont des langages réguliers.
- $\{1\} \cup \{0\}$ est régulier.
- $(\{1\} \cup \{0\})^*$ est régulier.
- $(\{1\} \cup \{0\})^* \cdot \{1\}$ est régulier.

Exercice 2.1 – correction

- Soit $L = \{w_1, w_2, \dots w_n\}$ un langage fini.
- Comme chaque mot w_i est une concaténation finie de caractères de Σ, il est clair que $\{w_i\}$ est régulier pour tout $1 \leq i \leq n$.
- Donc, $\{w_1\} \cup \{w_2\} \cup \dots \cup \{w_n\} = L$ est régulier.

Exercice 2.2 – correction

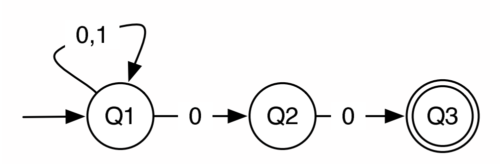
Non !

- Preuve par contradiction. Supposons que L est régulier.
- Donc, il existe un automate fini $A = \langle Q, \Sigma, \delta, q_0, F \rangle$.
- Intuitivement: comme Q est fini, il existe un mot de L qui est accepté en passant deux fois par le même état q. Par exemple: $w = 0^{2|Q|}1^{2|Q|}$.

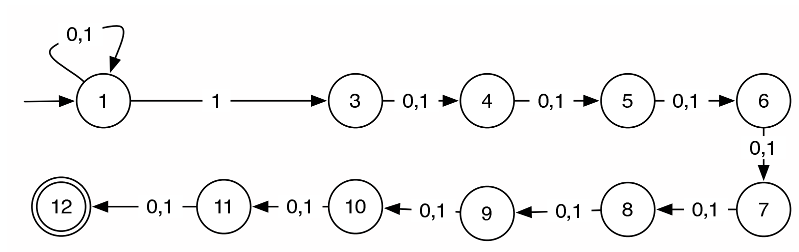
Exercice 2.2 – correction

- Donc, il existe un chemin de q_0 à q labellé par 0^{k_1}, une boucle allant de q à q labellée par 0^{k_2} et un chemin allant de q à $q' \in F$, labellé par par $0^{k_3}1^{2|Q|}$, avec $k_1 + k_2 + k_3 = 2|Q|$.
- Mais alors, on peut aussi accepter, par exemple le mot $0^{k_1}0^{k_3}1^{2|Q|}$, qui n'est pas dans L.
- Contradiction: A ne peut pas exister et donc L n'est pas régulier.

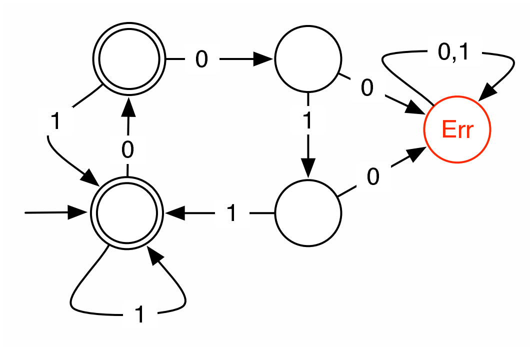
Exerice 3.1 – correction



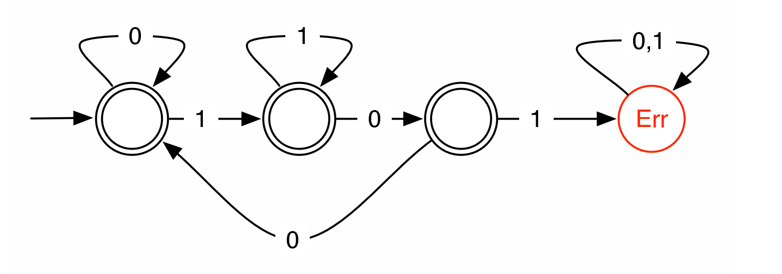
Exerice 3.2 – correction



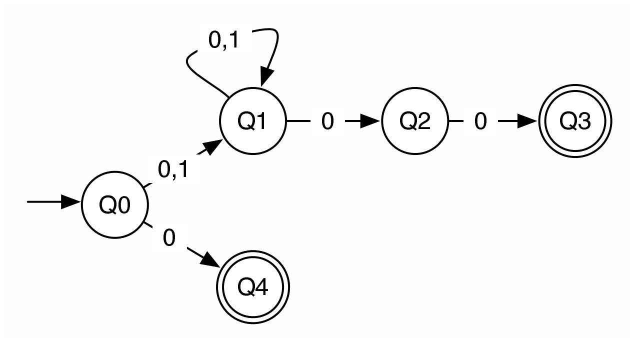
Exerice 3.3 – correction



Exerice 3.4 – correction



Exerice 3.5 – correction



Exercice 4.1 – correction

δ	1	0
$\{p\}$	$\{p\}$	$\{p,q\}$
$\{p,q\}$	$\{p,t\}$	$\{p,q,r,s\}$
$\{p,t\}$	$\{p,s\}$	$\{p,q\}$
$\{p,q,r,s\}$	$\{p,t\}$	$\{p,q,r,s\}$
$\{p,s\}$	$\{p\}$	$\{p,q\}$

État initial: $\{p\}$. États accepteurs: $\{p,t\}$, $\{p,s\}$ et $\{p,q,r,s\}$.

Exercice 4.2 – correction

δ	1	0
$\{p\}$	$\{q\}$	$\{q,s\}$
$\{q\}$	$\{q,r\}$	$\{r\}$
$\{r\}$	$\{p\}$	$\{s\}$
$\{s\}$	$\{p\}$	$\emptyset$
$\{q,s\}$	$\{p,q,r\}$	$\{r\}$
$\{r,s\}$	$\{p\}$	$\{s\}$
$\{p,q,r\}$	$\{p,q,r\}$	$\{q,r,s\}$
$\{q,r,s\}$	$\{p,q,r\}$	$\{r,s\}$
$\emptyset$	$\emptyset$	$\emptyset$

État initial: $\{p\}$. États accepteurs: $\{q\}$, $\{s\}$, $\{q,s\}$, $\{r,s\}$, $\{p,q,r\}$ et $\{q,r,s\}$.

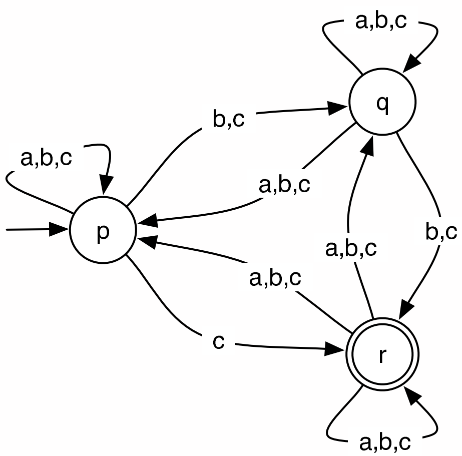
Exercice 4.3 – correction

Calcul des ε-closures:

	a	b	c
p	$\{p\}$	$\{p,q\}$	$\{p,q,r\}$
q	$\{p,q\}$	$\{p,q,r\}$	$\{p,q,r\}$
r	$\{p,q,r\}$	$\{p,q,r\}$	$\{p,q,r\}$

NB: cet automate est non-déterministe !

Exercice 4.3 – correction



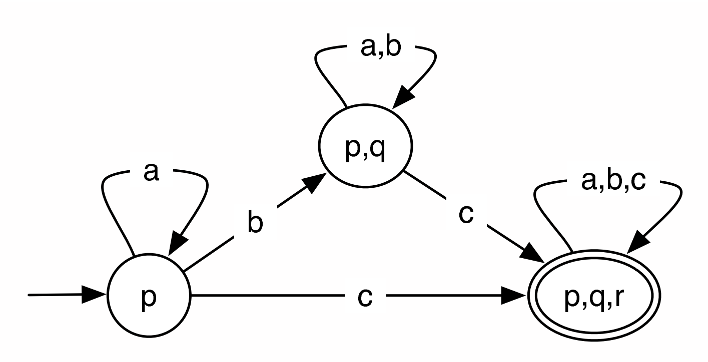
Exercice 4.3 – correction

Déterminisation:

δ	a	b	c
$\{p\}$	$\{p\}$	$\{p,q\}$	$\{p,q,r\}$
$\{p,q\}$	$\{p,q\}$	$\{p,q,r\}$	$\{p,q,r\}$
$\{p,q,r\}$	$\{p,q,r\}$	$\{p,q,r\}$	$\{p,q,r\}$

État initial: $\{p\}$.État accepteur: $\{p,q,r\}$.

Exercice 4.3 – correction



Exercice 5 – correction

```
char buffer ; // Initialise au 1er car de l'input

char next_char() {
   return buffer ;
}

void read_next(){
   buffer = getchar() ;
}

bool alpha(char c) {
 return (c >= 'A' && c <= 'Z') ;
}
```


Exercice 5 – correction

```
int automate() {
  int state = 8 ;
  char c ;
  while (true){
    switch state {
      case 8:
        if (next_char() == 'W') state = 4 ;
        else if (next_char() == 'I') state = 9 ;
        else if (alpha(c)) state = 3 ;
        else state = 8 ;
        read_next() ;
        break ;
      case 4:
        ...
```


Exercice 5 – correction

```
case 2 :
  if (alpha(c)) state = 3 ;
  else return 2 ; // Pas de read_next !
  read_next() ;
  break ;
```