# Langages réguliers et automates finis

Université Libre de Bruxelles
2008 - 2009

Denis BOIGELOT Sébastien COLLETTE Gilles GEERAERTS

# Coordonnées


- Denis Boigelot
- dboigelo@ulb.ac.be
- 2N8.211
- http://www.ulb.ac.be/di/ssd/tmassart/Compil
- http://student.ulb.ac.be/~dboigelo/


# Langage régulier

Soit un alphabet $\Sigma$ (fini)

Cas de base:

- $\emptyset$ est un langage régulier
- $\{\varepsilon\}$ est un langage régulier
- $\forall a \in \Sigma$, $\{a\}$ est un langage régulier

# Langage régulier

Soient L et K des langages réguliers

Inductions:

- $L \cdot K := \{ l \cdot k \mid l \in L \wedge k \in K\}$ est régulier
- $L \cup K$ est régulier
- $L^* := \{\varepsilon\} \cup \{ww\ldots w \mid w \in L\}$ est régulier

# Automate fini

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

- $Q$ est un ensemble fini des états;
- $\Sigma$ est l'alphabet des symboles à l'entrée;
- $\delta$ est la relation de transition: $Q \times \Sigma \times Q$;
- $q_0$ est l'état initial;
- $F \subseteq Q$ ensemble des états accepteurs.

# Propriété importante

Si R est un langage régulier, alors il existe un automate fini M tel que $L(M) = R$.

# 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.

# Solution 1.1

- $1 \in \Sigma$ 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.

# Solution 1.2

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

# Solution 2.1

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

# Solution 2.2

Réponse: Non!

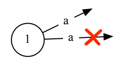
- Preuve par contradiction. Supposons que L est régulier.
- Donc, il existe un automate fini $A = \langle Q, \Sigma, \delta, q_0, F\rangle$.
- Intuitivement: comme Q est fini, il existe un mot de L qui est accepté en passant deux fois par le même état q. Par exemple: $w = 0^{2|Q|}1^{2|Q|}$.

# Solution 2.2

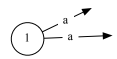
- Donc, il existe un chemin de $q_0$ à q labellé par $0^{k1}$, une boucle allant de q à q labellée par $0^{k2}$ et un chemin allant de q à $q' \in F$, labellé par par $0^{k3}1^{2|Q|}$, avec $k1 + k2 + k3 = 2|Q|$.
- Mais alors, on peut aussi accepter, par exemple le mot $0^{k1}0^{k3}1^{2|Q|}$, qui n'est pas dans L.
- Contradiction: A ne peut pas exister et donc L n'est pas régulier.

# Automate fini

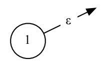
- DFA (deterministic finite automata)



- NFA (nondeterministic finite automata)



- ε-NFA (epsilon-transitions NFA)

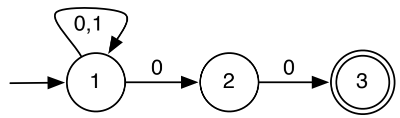


# Exercice 3

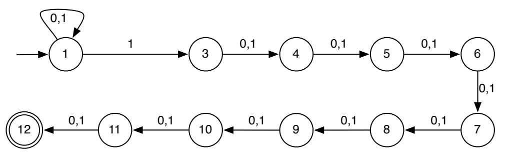
Donnez un automate non déterministe qui accepte chacun des langages suivants (définis sur l'alphabet $\Sigma = \{0, 1\}$):

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.

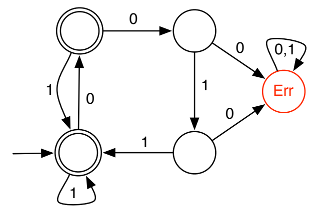
# Solution 3.1



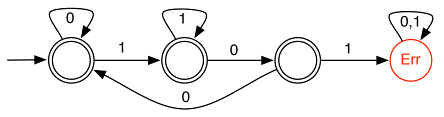
# Solution 3.2



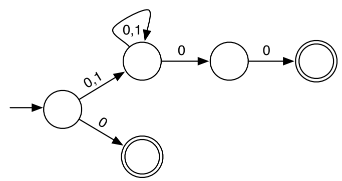

# Solution 3.3



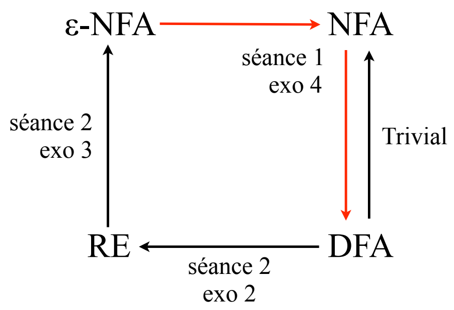

# Solution 3.4




# Solution 3.5



# Liens entre automates




# Déterminiser un automate

Technique: *subset construction*

NFA $N = \langle Q_N, \Sigma, \delta_N, q_0, F_N \rangle \longrightarrow$ DFA $D = \langle Q_D, \Sigma, \delta_D, \{q_0\}, F_D \rangle$ où

- $Q_D = \mathcal{P}(Q_N)$
- $F_D = \{S \mid S \subseteq Q_N \text{ tel que } S \cap F_N \neq \emptyset\}$
- $\forall\, S \subseteq Q_N, \forall a \in \Sigma, \delta_D(S,a) = \bigcup_{p \in S} \delta_N(p,a)$

# Supprimer les ε-transitions

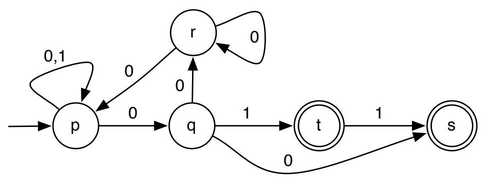
ε-NFA $E = \langle Q, \Sigma, \delta_E, q_0, F \rangle \longrightarrow$ NFA $N = \langle Q, \Sigma, \delta_N, q_0, F \rangle$ tel que:

soit $\alpha \in \Sigma$, si $q' \in \hat{\delta}_E(q, \alpha)$, alors $(q, \alpha, q') \in \delta_N$

c'est-à-dire: s'il existe un chemin de q à q' passant par un et un seul α (plus éventuellement des ε) dans le ε-NFA, alors on inclut la transition (q, α, q') au NFA

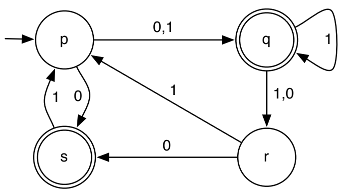
# Exercice 4.1

Déterminisez:



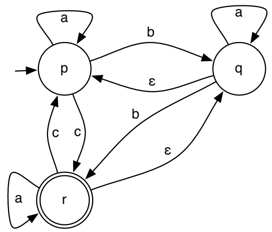
# Exercice 4.2

Déterminisez:




# Exercice 4.3

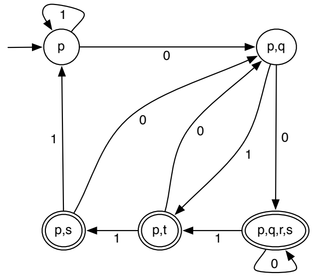
Déterminisez:




# Solution 4.1

| 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} |

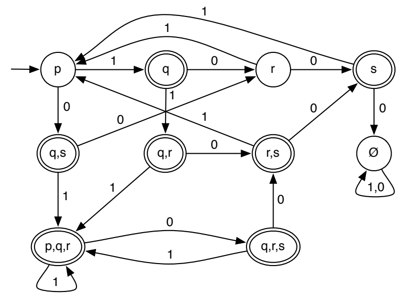
# Solution 4.1



# Solution 4.2

| δ | 1 | 0 |
|---|---|---|
| → {p} | {q} | {q,s} |
| * {q} | {q,r} | {r} |
| {r} | {p} | {s} |
| * {s} | {p} | ∅ |
| * {q,s} | {p,q,r} | {r} |
| * {q,r} | {p,q,r} | {r,s} |
| * {r,s} | {p} | {s} |
| * {p,q,r} | {p,q,r} | {q,r,s} |
| * {q,r,s} | {p,q,r} | {r,s} |
| ∅ | ∅ | ∅ |

# Solution 4.2

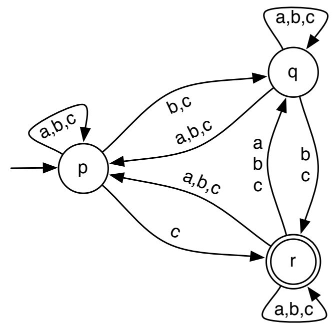



# Solution 4.3

Suppression des ε-transitions:

| | 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} |

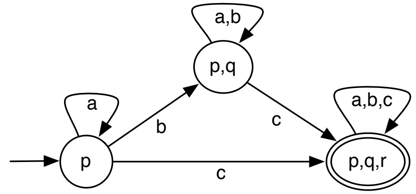
# Solution 4.3



# Solution 4.3

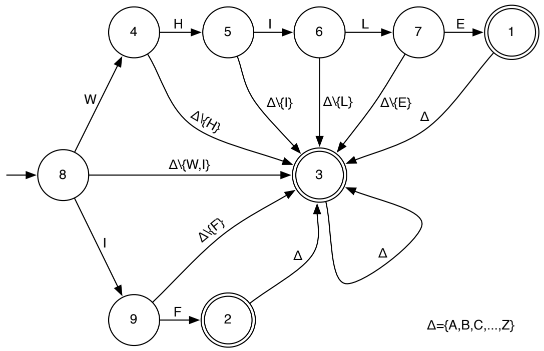
| δ | a | b | c |
|---|---|---|---|
| → {p} | {p} | {p,q} | {p,q,r} |
| {p,q} | {p,q} | {p,q,r} | {p,q,r} |
| * {p,q,r} | {p,q,r} | {p,q,r} | {p,q,r} |

# Solution 4.3




# Exercice 5

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




# Solution 5

```
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’);}
```

# Solution 5

```
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;
```

# Solution 5

```
            case 4:
                ...
            case 2:
                if (alpha(c)) state = 3;
                else return 2; // Pas de read_next!
                read_next();
                break;
        }
    }
```