Automates

Propriétés des langages réguliers

Damien Nouvel

Plan

- Lois sur les opérateurs de langage
- Démonstrations sur les langages
- Langages réguliers
- Expressions régulières et automates

Plan

- Lois sur les opérateurs de langage
- Démonstrations sur les langages
- Langages réguliers
- Expressions régulières et automates

Propriétés des langages réguliers
Lois sur les opérateurs de langage

- Union
  - **Commutative** : L+M = M+L
  - **Associative** : (L+M)+N = L+(M+N)
  - **Idempotente** : L+L = L
  - Opérateur **identité** Ø : L+Ø = L
- Concaténation
  - **Non-commutative** : L.M ≠ M.L
  - **Associative** : (L.M).N = L.(M.N)
  - **Non-idempotente** : L.L ≠ L (=L^2)
  - **Distributive** par rapport à l'union : L.(M+N) = L.M+L.N
  - Opérateur **identité** ε : L.ε = L, **annulateur** Ø : L.Ø = Ø

- Propriétés de la fermeture
  - $(L^*)^* = L^*$
  - $\varnothing^* = \varepsilon$
  - $\varepsilon^* = \varepsilon$
  - $L^+ = L.L^* = L^*.L$
  - $L^* = L^+ + \varepsilon$
  - $L? = L + \varepsilon$

Plan

- Lois sur les opérateurs de langage
- Démonstrations sur les langages
- Langages réguliers
- Expressions régulières et automates

- Déterminer si deux langages sont **équivalents**
  - Par les expressions régulières « concrètes »
  - Par les **automates**
  - Raisonnement par **récursivité** sur la longueur du mot
  - Raisonnement par **décomposition** du mot
  - Invalidation par un **contre-exemple**
- Passage aux expressions régulière « **concrètes** »
  - Suppression des variables, remplacées par des symboles
- **Equivalence** entre L et M suppose
  - Quelque soit w ∈ L alors w ∈ M
  - Quelque soit w ∈ M alors w ∈ L

- Par exemple, prouver que $(L+M)^* \leftrightarrow (L^*M^*)^*$
  - **Expressions régulières concrètes** : $(a+b)^* = (a^*b^*)^*$
  - Implication $\rightarrow$
    - Soit $w \in (a+b)^*$ alors nous **décomposons** : $w = w_1w_2...w_n$
    - Pour tout i, $w_i = a$ ou $w_i = b$, dans tous les cas $w_i \in (a^*b^*)$
    - Alors $w = w_1w_2...w_n \in (a^*b^*)^*$
  - Implication $\leftarrow$
    - Soit $w \in (a^*b^*)^*$ alors nous **décomposons** : $w = w_1w_2...w_n$
    - Pour tout i, $w_i = a^jb^k = x.y$ tels que $x \in (a+b)^*$ et $y \in (a+b)^*$
    - Ainsi $x.y \in (a+b)^*.(a+b)^* = (a+b)^*$
    - Alors $w = w_1w_2...w_n \in ((a+b)^*)^* = (a+b)^*$

Propriétés des langages réguliers
Plan

- Lois sur les opérateurs de langage
- Démonstrations sur les langages
- Langages réguliers
- Expressions régulières et automates

- Langage régulier
  - Langage régulier reconnu par un **automate**
  - Langage régulier reconnu par une **expression régulière**
- La combinaison de **langages réguliers** est **régulier**
  - Union
  - Concaténation
  - Fermeture
  - Intersection
  - Complémentaire
  - Différence
  - Homomorphisme

- Union, concaténation, fermeture
  - Soit L un langage régulier défini par l'expression régulière E et M un langage régulier défini par l'expression régulière F, par définition des **opérateurs** « ∪ », « . » et « * » :

    $L \cup M = L(E+F)$

    $L . M = L(EF)$

    $L^* = L(E)^* = L(E^*)$
  - Alors « $L \cup M$ », « $L . M$ » et « L^* » sont représentés par des **expressions régulières** et sont donc des **langages réguliers**

- Intersection
  - Soient L et M deux **langages réguliers**, représentés par les automates déterministes :

$$A_L = (Q_L, \Sigma_L, \delta_L, q_{0L}, F_L)$$

$$A_M = (Q_M, \Sigma_M, \delta_M, q_{0M}, F_M)$$

  - Construisons l'**automate déterministe** A_N défini comme suit

$$A_N = (Q_N, \Sigma_N, \delta_N, q_{0N}, F_N)$$

$$\Sigma_N = \Sigma_L \cup \Sigma_M$$

$$Q_N = Q_L \times Q_M = \{q_i, q_j \mid q_i \in Q_L, q_j \in Q_M \}$$

$$q_{0N} = \{q_{0L}, q_{0M}\}$$

$$F_N = F_L \times F_M = \{q_i, q_j \mid q_i \in F_L, q_j \in F_M \}$$

- Intersection (suite)
  - La **fonction de transition** de l'automate A_N est définie selon celle des automates A_L et A_M - pour chaque élément (q_i, q_j) de Q_N et pour chaque symbole $a \in \Sigma_N$

    $$\delta_N(\{q_i, q_j\}, a) = \{\delta_L(q_i, a), \delta_M(q_j, a)\}$$

  - L'automate **accepte un langage** reconnu par A_L et par A_M
    - Supposons $w = x.a$ tel quel $\overline{\delta}_N(\{q_i, q_j\}, x) = \{\overline{\delta}_L(q_i, x), \overline{\delta}_M(q_i, x)\}$
    - Alors par définition de l'automate déterministe

      $$\overline{\delta}_N(\{q_i, q_j\}, x.a) = \delta_N(\overline{\delta}_N(\{q_i, q_j\}, x), a) = \delta_N(\{\overline{\delta}_L(q_i, x), \overline{\delta}_M(q_j, x)\}, a)$$

    - Et par construction de l'automate A_N

      $$\overline{\delta}_N(\{q_i, q_j\}, x.a) = \{\delta_L(\overline{\delta}_L(q_i, x), a), \delta_M(\overline{\delta}_M(q_j, x), a)\}$$

      $$\overline{\delta}_N(\{q_i, q_j\}, x.a) = \{\overline{\delta}_L(q_i, x.a), \overline{\delta}_M(q_j, x.a)\}$$

- Intersection (suite)
  - L'automate accepte un langage accepté par A_L et par A_M
    - Ainsi, la **fonction de transition étendue** définissant le langage

$$L = \{w \mid \overline{\delta}_L(q_{0L}, w) \in F_L\}$$

$$M = \{w \mid \overline{\delta}_M(q_{0M}, w) \in F_M\}$$

    - Nous avons pour l'automate A_L

$$N = \{w \mid \overline{\delta}_N(q_{0N}, w) \in F_N\}$$

$$N = \{w \mid \overline{\delta}_N(\{q_{0L}, q_{0M}\}, w) \in \{q_i, q_j \mid q_i \in F_L, q_j \in F_M\}\}$$

$$N = \{w \mid \{\overline{\delta}_L(q_{0L}, w), \overline{\delta}_M(q_{0M}, w)\} \in \{q_i, q_j \mid q_i \in F_L, q_j \in F_M\}\}$$

$$N = \{w \mid \overline{\delta}_L(q_{0L}, w) \in F_L, \overline{\delta}_M(q_{0M}, w) \in F_M\}\}$$

  - L'intersection de deux langages peut-être représentée par un langage et est donc un langage **régulier**

- Complémentaire
  - Soit L un langage régulier défini sur un alphabet Σ, alors le complémentaire correspond à $\Sigma^* - L$
  - On considère l'automate reconnaissant L

    $A_L = (Q_L, \Sigma_L, \delta_L, q_{0L}, F_L)$
  - Nous construisons l'automate

    $A_M = (Q_L, \Sigma_L, \delta_L, q_{0L}, Q_L - F_L)$

    Même automate, sauf que les états acceptants deviennent non-acceptants, et vice-versa
  - Tout mot accepté par A_L ne l'est **pas** par A_M et inversement :

    $M = \{ w \mid \overline{\delta}_L(q_{0L}, w) \notin F_L \}$
  - Le langage est accepté par un automate et est régulier

- Différence
  - On remarque que la différence entre deux langages L et M peut-être réalisée en utilisant l'**intersection** et le **complémentaire**

    $$L - M = L \cap \overline{M}$$

  - Le passage au complémentaire et l'intersection de langages réguliers donnent un langage **régulier**

- Homomorphisme
  - Ou « **morphisme de groupes** » : application qui « préserve la structure des groupes », par ex. pour des opérateurs * et ∘ :

    $h(x*y) = h(x) \circ h(y)$
  - Par exemple pour les langages :
    - Conversion de caractères alphabétiques → numériques
    - Suppression des voyelles
    - Remplacement des espaces par des sauts de lignes
    - ...
  - Formellement pour les langages

    $\mathbf{h(x.y.z) = h(x).h(y).h(z)}$

    $\mathbf{h(L) = \{\ h(w) \mid w \in L\ \}}$

- Homomorphisme (suite)
  - Application d'un homomorphisme à une expression régulière
    - Chaque **symbole** de l'expression
  - Soit un langage L représenté par l'expression régulière E

    h(L(E)) = L(h(E))
  - Preuve par **récursivité**
    - Pour un **symbole** simple E = a alors L(h(a)) = h(a) = h(L(a))
    - Selon la **forme de l'expression régulière**
      - Si E = F+G, en supposant L(h(F)) = h(L(F)) et L(h(G)) = h(L(G)) alors

        L(h(E) = L(h(F+G)) = L(h(F) + h(G)) = L(h(F)) ∪ L(h(G))

        h(L(E) = h(L(F+G)) = h(L(F) ∪ L(G)) = h(L(F)) ∪ h(L(G))

        Par induction : h(L(E) = L(h(E)

- Homomorphisme (suite)
  - Preuve par **récursivité** (suite)
    - Selon la **forme de l'expression régulière** (suite)
      - Si $E = F.G$, en supposant $L(h(F)) = h(L(F))$ et $L(h(G)) = h(L(G))$ alors

        $h(E) = h(F.G) = h(F).h(G)$
        $L(E) = L(F).L(G)$
        $L(h(E)) = L(h(F.G)) = L(h(F).h(G)) = L(h(F)).L(h(G))$
        $h(L(E)) = h(L(F).L(G)) = h(L(F)).h(L(G))$
        Par induction, $L(h(E)) = h(L(E))$
      - Si $E = F^*$, en supposant $L(h(F)) = h(L(F))$

        $h(E) = h(F^*) = h(F)^*$
        $L(h(E)) = L(h(F)^*) = L(h(F))^*$
        $h(L(E)) = h(L(F^*)) = h(L(F))^*$
        Par induction, $L(h(E)) = h(L(E))$
  - Un langage, lors de sa transformation par homomorphisme reste reconnu par une expression régulière et est **régulier**

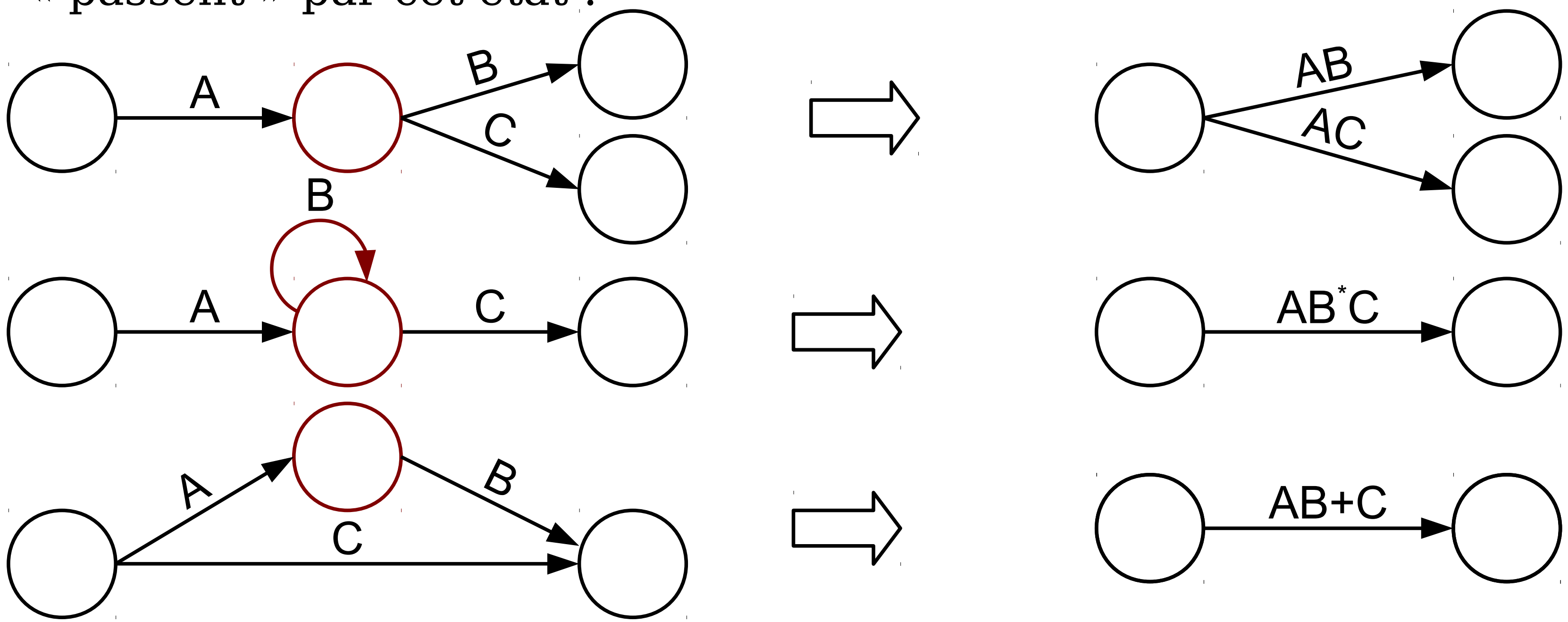
- Lois sur les opérateurs de langage
- Démonstrations sur les langages
- Langages réguliers
- Expressions régulières et automates

Propriétés des langages réguliers
Expressions régulières et automates

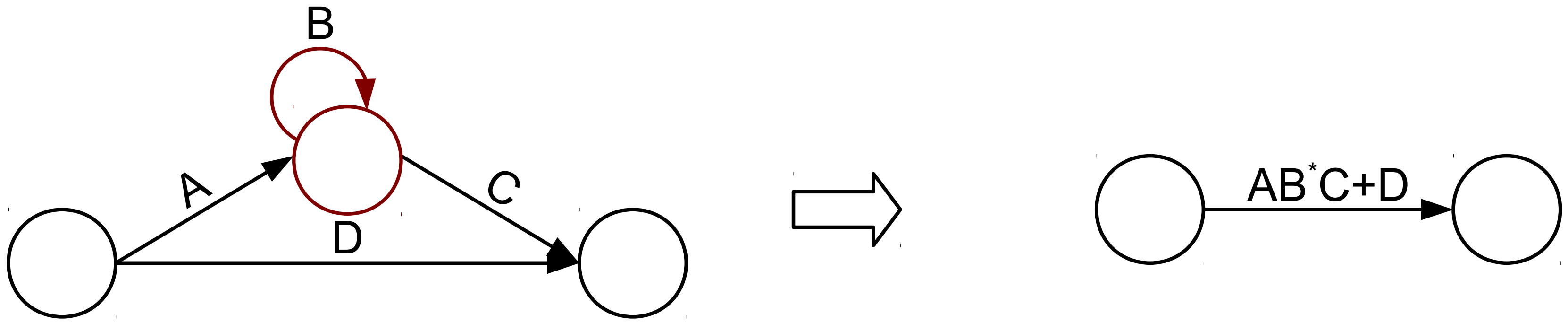
- Langages des expressions régulières et des automates
  - Théorème : **pour tout automate**, il existe **une expression régulière reconnaissant le même langage**
    - Possibilité de transformer un automate en expression régulière
  - Théorème : **pour toute expression régulière**, il existe **un automate reconnaissant le même langage**
    - Possibilité de transformer une expression régulière en automate
- Passage par une représentation « intermédiaire »
  - Procédure **précise** pour réaliser la transformation
  - Ressemble aux automates (noeuds = états, arcs = transitions)
  - Transitions **notées comme des expressions régulières**
  - Ce n'est plus un automate au sens DFA / FNA / ε-NFA

Propriétés des langages réguliers
Expressions régulières et automates

- Automate → expression régulière
  - Remplacement des « , » par des « + » sur les transitions
  - Élimination des états « intermédiaires » (sauf initial / finaux)
    - Pour un état, déterminer l'expression régulière des transitions qui « passent » par cet état :



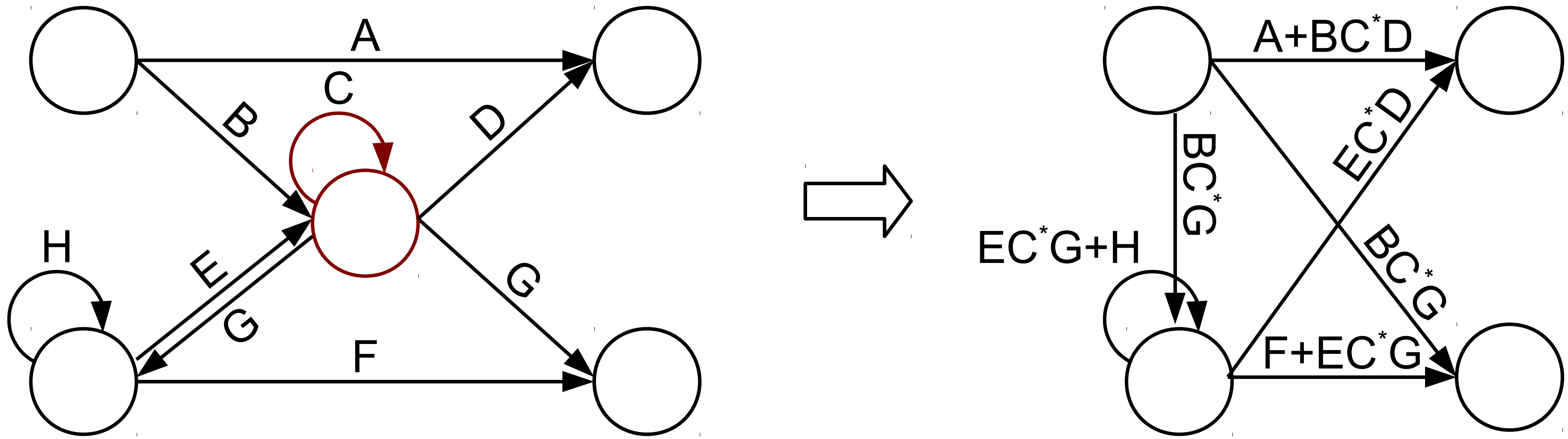
- Automate → expression régulière (suite)
  - Dans le cas général :



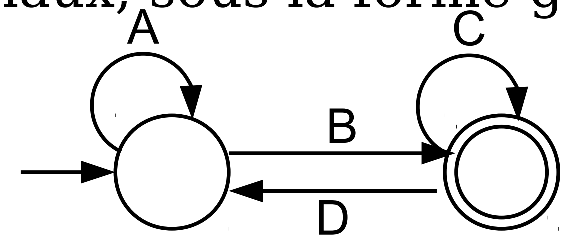
  - Cas particulier :

Propriétés des langages réguliers
Expressions régulières et automates

- Automate → expression régulière (suite)
  - On considère toutes les paires d'états entrants / sortants :

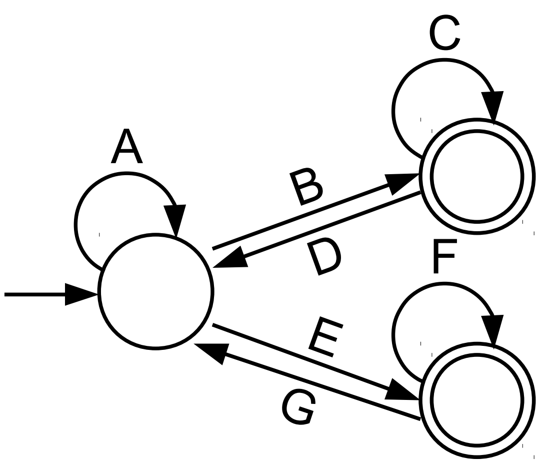


- Automate → expression régulière (suite)
  - Pour terminer, ne sont conservés que l'état initial et les états finaux, sous la forme générale :



⇒ $(A+BC^*D)^*BC^*$

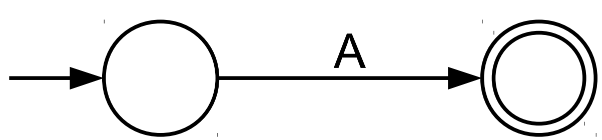
  - Pour chaque état final, on supprimer les autres états finaux, puis on fait la disjonction entre les expressions obtenues :



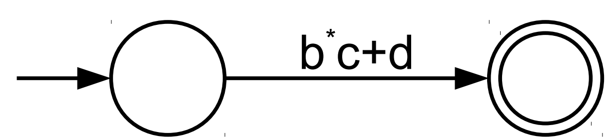
⇒ $((A+EF^*G)+BC^*D)^*BC^*$
$+$
$((A+BC^*D)+EF^*G)^*EF^*$

Propriétés des langages réguliers
Expressions régulières et automates

- Expression régulière → automate
  - Séparer les parties de l'expression régulière pour créer les nœuds de l'automate
    - Il faut une procédure dans laquelle à chaque étape les nœuds au départ / à la fin de l'expression régulière sont identifiés
    - L'utilisation des transitions ε permet de décomposer l'expression régulière de manière automatique
  - Au départ, on considère l'expression régulière entière comme une seule transition :



Si $A = b^*c+d$:



- Expression régulière → automate
  - On décompose récursivement les parties de l'expression régulière, en prenant garde au parenthèses et à la précédence des opérateurs :