Chapitre II
Langages réguliers et Automates finis

1. Grammaire régulières
2. Automates Finis
3. Automates finis indéterministes
4. Automates finis déterministes
5. Expressions régulières

Grammaire régulière

Grammaire régulière

Une grammaire $G = (T, N, S, R)$ est régulière

- À droite, si les règles de R sont de la forme :
  $A \to aB$ ou $A \to a$ avec $A, B \in N$ et $a \in T$

- À gauche, si les règles de R sont de la forme:
  $A \to Ba$ ou $A \to a$ avec $A, B \in N$ et $a \in T$

Exemple : $G_1 = (T_1, N_1, S_1, R_1)$ avec

$$T_1 = \{a, b\}$$
$$N_1 = \{S_1, U_1\}$$
$$R_1 = \{S_1 \to aS_1 \mid aU_1,$$
$$U_1 \to bU_1 \mid b\}$$

Grammaire régulière

Exemple

- Une grammaire régulière à droite:

$G_1 = (T_1, N_1, S_1, R_1)$ avec

$$T_1 = \{a, b\}$$
$$N_1 = \{S_1, U_1\}$$
$$R_1 = \{S_1 \rightarrow aS_1 \mid aU_1,\ U_1 \rightarrow bU_1 \mid b\}$$

- Une grammaire régulière à gauche:

$G_2 = (T_2, N_2, S_2, R_2)$ avec

$$T_2 = \{a, b\}$$
$$N_2 = \{S_2, U_2\}$$
$$R_2 = \{S_2 \rightarrow S_2b \mid U_1b,\ U_2 \rightarrow U_2a \mid a\}$$

G_1 et G_2 engendre le même langage :

$$\mathcal{L}(G_1) = \mathcal{L}(G_2) = \{a^n b^m \ n > 0 \ et \ m > 0\}$$

Langage régulier

Langage régulier

Un langage est régulier si et seulement s'il existe une grammaire régulière générant ce langage.

Les grammaires et langages réguliers sont la base de la lexicographie. c-à-d, l'ensemble des :

- mots-clés,
- identificateurs,
- constantes numériques,
- etc

d'un langage de programmation (tel que C++) appartiennent à un langage régulier décrit par une grammaire régulière.

Langage régulier

Analyse descendante des mots

Si on lit les symboles du mot à analyser de la gauche vers la droite, alors une grammaire régulière à droite sera utilisée pour une ***analyse descendante***, de l'axiome vers le mot;

Exemple: *pour analyser le mot* $\boldsymbol{aaabb}$ *avec la grammaire* G_1:

$G_1 = (T_1, N_1, S_1, R_1)$ avec

$T_1 = \{a, b\}$

$N_1 = \{S_1, U_1\}$

$R_1 = \{S_1 \rightarrow aS_1 \mid aU_1,$

$U_1 \rightarrow bU_1 \mid b\}$

on construira la dérivation:

$$S_1 \Rightarrow aS_1 \Rightarrow aaS_1 \Rightarrow aaaU_1 \Rightarrow aaabU_1 \Rightarrow aaabb$$

Langage régulier

Analyse ascendante des mots

Si on lit les symboles du mot à analyser de la droite vers la gauche, alors une grammaire régulière à gauche sera utilisée pour une ***analyse ascendante***, du mot vers l'axiome;

Exemple: *pour analyser le mot* $\boldsymbol{aaabb}$ *avec la grammaire* G_2:

$G_2 = (T_2, N_2, S_2, R_2)$ avec

$T_2 = \{a, b\}$

$N_2 = \{S_2, U_2\}$

$R_2 = \{S_2 \rightarrow S_2 b \mid U_2 b,$

$U_2 \rightarrow U_2 a \mid a\}$

on construira la dérivation:

$$aaabb \Leftarrow U_2 aabb \Leftarrow U_2 abb \Leftarrow U_2 bb \Leftarrow S_2 b \Leftarrow S_2$$

Langage régulier

Propriétés des langages réguliers

Étant donné un alphabet A, on appel langage régulier sur A un langage sur A défini de la façon suivante :

- $\emptyset$ (l'ensemble vide) est langage régulier sur A,
- $\{\epsilon\}$ est langage régulier sur A,
- $\{a\}$ est langage régulier sur A pour tout $a \in A$,
- Si P et Q sont des langages réguliers sur A, alors les langages suivants sont des langages réguliers:
  - $P \cup Q$
  - PQ
  - P^*

Expressions régulières

Exemples des langes réguliers

- Pour tout mot $u \in A^*$, le langage $\{u\}$ est régulier.
  - Si u s'écrit $a_1 a_2 \dots a_n$ sur A, alors le langage $\{u\}$ s'écrit comme la concaténation $\{u\} = \{a_1\}\{a_2\} \dots \{a_n\}$.
  - $\{u\}$ est régulier car chaque $\{a_i\}$ est régulier,

- Tout langage fini est régulier.
  - Un ensemble fini de mots $\mathcal{L} = \{u_1, u_2, \dots, u_n\}$ s'écrit:
    $$\mathcal{L} = \{u_1\} \cup \{u_2\} \cup \dots \cup \{u_n\}$$
  - $\mathcal{L}$ est régulier car chaque $\{u_i\}$ est régulier, et leurs union donne un langage régulier.

- Sur l'alphabet $\{a, b\}$, l'ensemble $\{a^n b^m \ / \ n, m \in \mathbb{N}\}$ est régulier.
  - Le langage $\{a^n \ / \ n \in \mathbb{N}\} = \{a\}^*$ est régulier,
  - De même, $\{b^m \ / \ m \in \mathbb{N}\} = \{b\}^*$ est régulier,
  - Le langage $\{a^n b^m \ / \ n, m \in \mathbb{N}\}$, la concaténation des deux précédents, est donc régulier.

Automates Finis

Automate Fini

Un ***automate*** est une procédure effective (un algorithme) permettant de déterminer si un mot donné appartient à un langage.

Un ***Automate fini*** est une construction mathématique abstraite:

- utilisée seulement pour la reconnaissance des langages réguliers,
- caractérisée par un ***nombre fini d'états***,
- mais peut être dans un seul état à la fois (***l'état courant***),
- le passage d'un état à un autre est activé par ***une transition***.

Alors, un automate fini est défini par l'ensemble de ses états et l'ensemble de ses transitions.

Automates Finis Indéterministes

Définition (AFI)

Un automate fini indéterministe est défini par un quintuplet (K, T, M, I, F) tel que:

- K est un ensemble fini d'états.
- T est le vocabulaire terminal (correspondant à l'alphabet sur lequel est défini le langage).
- M est une relation dans $K \times T \times K$ appelée relation de transition.
- $I \subseteq K$ est l'ensemble des états initiaux.
- $F \subseteq K$ est l'ensemble des états finaux.

Les éléments de M sont de la forme (S_i, a, S_j), où S_i et S_j sont des états de K, et a est un symbole du vocabulaire terminal T.

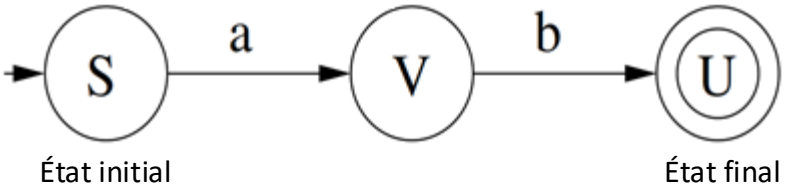
Automates Finis Indéterministes

Représentation graphique d'un automate fini

On représente un AFI par un graphe orienté dont les arcs sont étiquetés.

Dans cette représentation:

- Chaque état par un sommet du graphe,
- A chaque transition $(S_i, a, S_j) \in M$ on associe un arc du sommet S_i vers le sommet S_j étiqueté par a.
- Les sommets correspondant à des états initiaux de l'automate sont repérés par une pointe de flèche.
- Les sommets correspondant à des états finaux sont entourés de deux cercles.



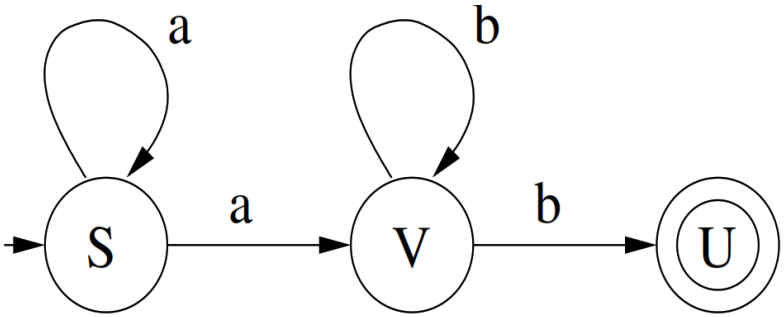
Automates Finis Indéterministes

Représentation graphique d'un AFI

Par exemple, l'AFI (K, T, M, I, F) tel que

- $K = \{S, V, U\}$,
- $T = \{a, b\}$,
- $M = \{(S, a, S), (S, a, V), (V, b, V), (V, b, U)\}$,
- $I = \{S\}$,
- $F = \{U\}$

sera représenté graphiquement par le graphe:



Automates Finis Indéterministes

Fonctionnement d'un AFI

De façon informelle, un mot u est accepté par un AFI s'il existe un chemin d'un sommet initial vers un sommet final tel que la concaténation des étiquettes des arcs empruntés par le chemin soit égale à u.

Sur l'exemple précédent, le langage des mots acceptés par l'automate est $\mathcal{L} = \{a^n b^m \ / \ n > 0 \ et \ m > 0\}$

Automates Finis Indéterministes

Non déterminisme

Un automate est dit indéterministe si :

- il peut y avoir plusieurs états initiaux,
- il peut exister plusieurs transitions possibles partant du même sommet $S_i \in K$ étiquetées par un même symbole terminal $a \in T$,

Parmi les plusieurs possibilités, si l'automate arrive à terminer la dérivation avec une transition jusqu'à un état final, alors le mots obtenu est accepté.

Si l'automate n'arrive pas à terminer la dérivation, alors retourne jusqu'au dernier point de choix (backtrack) et recommence avec une autre possibilité pour emprunter une autre route.

Automates Finis Indéterministes

Inconvénients du non déterminisme

L'exécution d'un automate fini indéterministe peut s'avérer très inefficace s'il comporte beaucoup de points de choix.

Autrement dit, si à chaque état l'automate a le choix entre deux transitions, alors pour analyser un mot de longueur n :

- Dans le pire des cas, il faudra envisager 2^n transitions,
- Dans le meilleur des cas, si on choisit toujours la "bonne" dérivation, on pourra trouver une dérivation en n transitions,

Automates Finis Déterministes

Pour éliminer ces points de choix, et rendre l'exécution efficace, il faut que l'automate soit déterministe, c'est-à-dire :

- Il ait un seul état initial,
- En partant d'un état $S_i \in K$ et d'un symbole $a \in T$, il existe une seule transition possible,

Automates Finis Déterministes

Définition (AFD)

Un automate fini déterministe est défini par un quintuplet (K, T, M, S_0, F) tel que:

- K est un ensemble fini d'états.
- T est le vocabulaire terminal.
- M est une relation dans $K \times T$ dans K,
- $S_0 \subseteq K$ est l'état initial.
- $F \subseteq K$ est l'ensemble des état finaux.

Dans AFD, la fonction de transition $M(S_i, a)$ donne l'état ***unique*** S_j dans lequel l'automate doit allez quand il se trouve dans l'état S_i et que le mot à analyser commence par le symbole a.

Automates Finis Déterministes

Exemple d'un AFD

Par exemple, l'AFI (K, T, M, S, F) tel que

- $K = \{S, V, U, E\}$,
- $T = \{a, b\}$,
- $M = \{(S, a) \to V, (S, b) \to E, (V, a) \to V, (V, b) \to U, (U, a) \to E, (U, b) \to U, (E, a) \to E, (E, b) \to E\}$,
- $I = \{S\}$,
- $F = \{U\}$

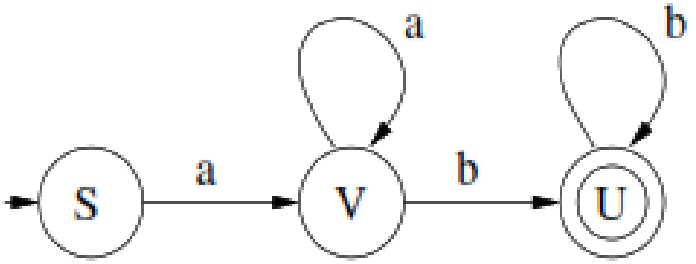
Cet AFD accepte le langage $\mathcal{L} = \{a^n p^m \ / \ n > 0, m > 0\}$

Automates Finis Déterministes

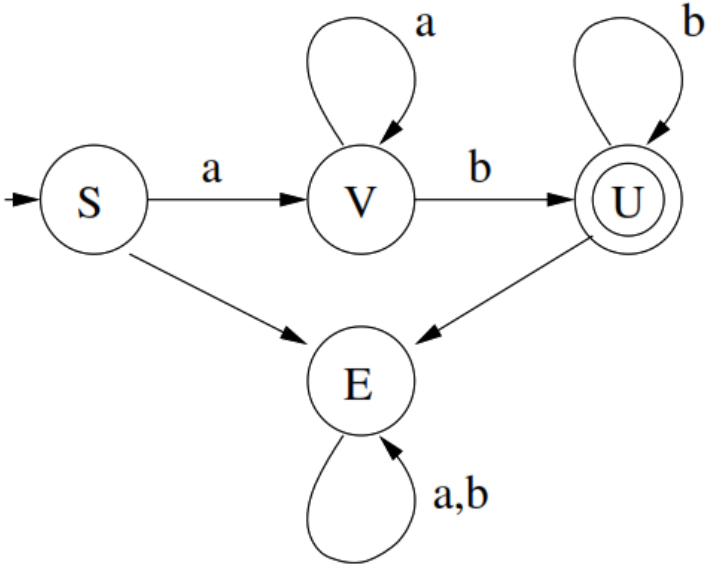
Représentation graphique d'un AFD

L'AFD est représenté généralement graphiquement par le graphe (a). Dans le graphe (b), on peut inclure l'état E qui correspond à un état d'erreur. L'automate vas dans E lors qu'il reconnait que le mot ne fait pas partie du langage.

Graphe (a)



Graphe (b)



Automates Finis Déterministes

L'exécution d'un automate fini déterministe est résumée dans la procédure "accepte" suivante :

```
procédure accepte
entrée : un AFD (K, T, M, S_0, F)
         un tableau de caractères u indicé de 1 à n
sortie : retourne vrai si u[1..n] appartient au langage, faux sinon
debut
    etatCrt ← S_0
    i ← 1
    tant que i ≤ n faire
        etatCrt ← M(etatCrt, u[i])
        i ← i + 1
    fin tant que
    si etatCrt ∈ F alors retourne vrai sinon retourne faux
fin
```

Expressions régulières

Expressions régulières

Une notation pratique pour dénoter des langages réguliers sur A, que l'on appelle expressions régulière sur A :

- $\emptyset$ est une expressions régulière dénotant le langage régulier $\emptyset$,
- ϵ est une expressions régulière dénotant le langage régulier $\{\epsilon\}$,
- a (tel que $a \in A$) est une expressions régulière dénotant le langage régulier $\{a\}$,
- Si p et q sont des expressions régulières dénotant respectivement les langages réguliers P et Q alors:
  - $(p + q)$ est une expression régulière dénotant le langage régulier $P \cup Q$
  - (pq) est une expression régulière dénotant le langage régulier PQ
  - $(p)^*$ est une expression régulière dénotant le langage régulier P^*
- Rien d'autre n'est une expression régulière.

Expressions régulières

Expressions régulières

Alors, une expression E est régulière sur A si et seulement si :

- $E = \emptyset$ ou,
- $E = \epsilon$ ou,
- $E = a$ (avec $a \in A$) ou,
- $E = E_1 \mid E_2$ et E_1 et E_2 sont deux expressions régulières sur A ou,
- $E = E_1.E_2$ et E_1 et E_2 sont deux expressions régulières sur A ou,
- $E = E_1^*$et E_1 est une expression régulière sur A,

Expressions régulières

Langage décrit par une expression régulière

Le langage $\mathcal{L}(E)$ décrit par une expression régulière E définie sur une alphabet A est définit par :

- $\mathcal{L}(E) = \emptyset$ si $E = \emptyset$,
- $\mathcal{L}(E) = \{\epsilon\}$ si $E = \epsilon$,
- $\mathcal{L}(E) = \{a\}$ si $E = a$,
- $\mathcal{L}(E) = \mathcal{L}(E_1) \cup \mathcal{L}(E_2)$ si $E = E_1 \mid E_2$,
- $\mathcal{L}(E) = \mathcal{L}(E_1).\mathcal{L}(E_2)$ si $E = E_1.E_2$,
- $\mathcal{L}(E) = \mathcal{L}(E_1)^*$ si $E = E_1^*$,

Où E_1 et E_2 sont deux expressions régulières sur A.

Priorités: Afin d'alléger les expressions régulières, on introduit les priorités suivantes:

$$priorité(*) > priorité(.) > priorité(+)$$

donc, $0 + 10^* \equiv \big(0 + (1(0)^*)\big)$

Expressions régulières

Exemples :

- E_1 Étant une expression régulière, on notera $\mathcal{L}(E_1)$ le langage dénoté par E_1: $\mathcal{L}\big(0 + (1(0)^*)\big) = \{0,1,10,100, \dots\}$
- L'expression régulière $\big(0 + (1(0)^*)\big)$ définie sur l'alphabet $\{0,1\}$ dénote le langage $\{0\} \cup \big\{\{1\}(\{0\})^*\big\}$
- C'est la langage formé du mot 0 et des mots composés d'un 1 suivi d'un nombre quelconque de 0.
- $E_2 = a^*bbc^*$ décrit le langage $\mathcal{L}(E_2) = \{a^n bbc^m \ / \ n \geq 0, m \geq 0\}$
- $E_3 = (a \mid b \mid c)^*(bb \mid cc)a^*$ décrit le langage $\mathcal{L}(E_2) = \{wbba^n, wcca^n \ / \ w \in A^*, n \geq 0\}$

Expressions régulières

Exemples :

$$0^*10^* = \{m \in \{0,1\}^* \mid m \text{ a exactement un } 1\}$$
$$(0+1)^*1(0+1)^* = \{m \in \{0,1\}^* \mid m \text{ a au moins un } 1\}$$
$$(0+1)^*001(0+1)^* = \{m \in \{0,1\}^* \mid m \text{ contient la sous} - \text{chaine } 001\}$$
$$\big((0+1)(0+1)\big)^* = \{m \in \{0,1\}^* \mid m \text{ est paire}\}$$