Automates à états finis

Damien Nouvel

Langages et automates

- Du langage ...
  - Hiérarchie (**réguliers**, hors-contexte, contextuels)

  ⇒ Mécanisme pour **accepter** / **reconnaître** un langage ?
- ...aux automates
  - Machine de Turing
  - États, transitions

⇒ **Automates** ...à **états finis** (langages réguliers)

- Représentations

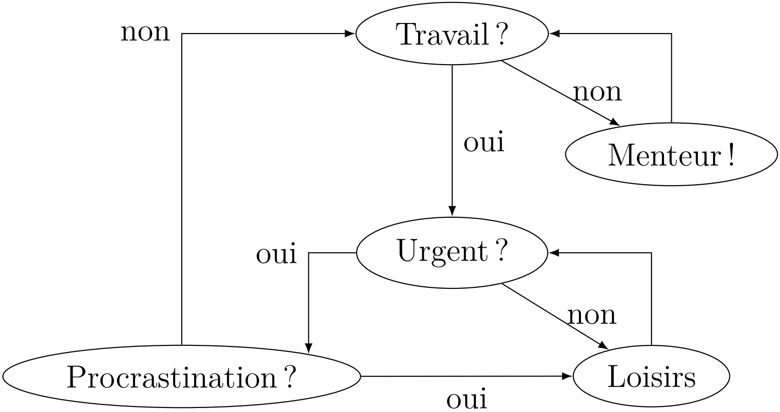
  ⇒ Diagrammes de transition (dessin)
  - **Graphe** : **nœuds**, **arcs**

  ⇒ Tables de transition

Plan

Diagrammes

- Inspiré des diagrammes de flux / organigrammes

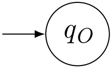


États

⇒ Indique où en est l'analyse d'un mot

- États : **nœuds**
  - **Cercle**
  - **Label** : q_i avec i un entier

- État **initial**
  - Ajout d'une **flèche** devant
  - Souvent q_0 (mais pas obligatoire)

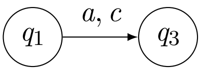


- État **final**
  - **Double** cercle

Transitions

⇒ Indique quelles prochains symboles sont acceptés

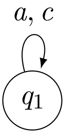
- Transitions : **arcs**
    - **Arc orienté** (flèche) qui relie deux états
    - **Label** : liste (ensemble) de symboles de Σ



⇒ Reconnaît le langage $\{a, c\}$ ou $\{a\} \cup \{c\}$ (mais pas $\{a.c\}$!)

⇒ Si depuis q_1 le prochain symbole est a ou c aller en q_3

- Transition d'un état vers lui-même

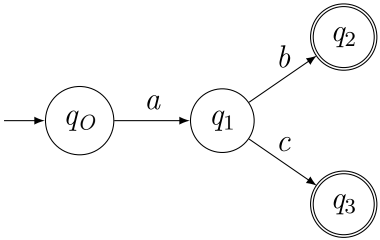


    - **Boucle** au dessus d'un état

⇒ Correspond à l'étoile de Kleene

Reconnaissance d’un mot

- Chemin suivi au travers d’un automate
  - L’automate **consomme** les symboles
  - Une liste d’état « visités » est établie
  - Arrivée en fin de mot dans l’état final
- Exemple : mots *ab* ou *ac*

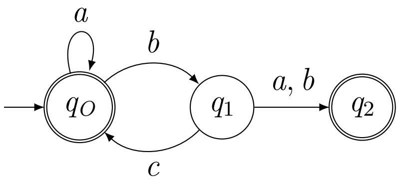


Automate à états finis déterministe

- Contraintes
  - **Un seul** état initial
  - **Déterministe** : par nœud / symbole, max **une** transition
- Remarques
  - Autant d'états finaux que nécessaires
  - L'état initial peut-être final
  - Des transitions peuvent partir de l'état final

$\Rightarrow$ Boucles possible sur un état ou par cycles

- Exemple : expression régulière `(a|bc)*(b(a|b))?`



Plan

Fonction de transition

- Définition formelle de l'automate : $A = (Q, \Sigma, \delta, q_0, F)$
  - États $Q = \{q_0, q_1, q_2, \ldots\}$
  - Alphabet $\Sigma = \{a, b, c, \ldots\}$
  - Fonction de transition $\delta : Q \times \Sigma \rightarrow Q \cup \{\emptyset\}$
  - État initial q_0
  - Ensemble d'états finaux F (avec $F \subset Q$)
- Importance de la **fonction de transition**
  - Cœur de l'automate (entrée : état et symbole / sortie : état)
  - Exemples : $\delta(q_0, a) = q_1$ ou $\delta(q_1, b) = q_0$
  - Valeurs à définir : $|Q| * |\Sigma|$
  - Une transition qui n'existe pas peut être notée $\emptyset$

Fonction de transition étendue

- La fonction de transition n'effectue qu'**une** transition
  - $\Rightarrow$ Comment savoir quel état est atteint avec n transitions ?
  - Pour abc, il faut calculer $\delta(\delta(\delta(q_0, a), b), c)$
  - $\Rightarrow$ Impossible de savoir combien de fois appliquer δ ...
- Fonction récursive $\delta^* : Q \times \Sigma^* \rightarrow Q \cup \{\varnothing\}$
  - Forme générale : $\delta^*(q, w)$ avec $w \in \Sigma^*$
    - Si $w = x.a$ avec $x \in \Sigma^*,\ a \in \Sigma$
    - $\Rightarrow$ Retourner $\delta(\delta^*(q, x), a)$
    - Si $w = a$ avec $a \in \Sigma$
    - $\Rightarrow$ Retourner $\delta(q, a)$
  - $\Rightarrow$ Permet de déterminer le **chemin**
  - $\Rightarrow$ Quel état est atteint pour un état et un mot donné

Langage reconnu

- Un automate reconnaît un langage

$\Rightarrow$ Concaténation de symboles telles que les transitions conduisent, depuis l'état initial, à un état final

$\Rightarrow$ $L = \{w | \delta^*(q_0, w) \in F\}$

Plan

Table de transition

- Représentation équivalente aux diagrammes
  - **Lignes** : états
  - **Colonnes** : symboles
  - **Cases** : résultat de la fonction de transition
- Notations
  - État initial : flèche
  - État final : étoile
- Exemple (cf diagramme)

Q	a	b	c
$\rightarrow *q_0$	q_0	q_1	$\emptyset$
q_1	q_2	q_2	q_0
$*q_2$	$\emptyset$	$\emptyset$	$\emptyset$

Exercices

- Modélisez par diagramme et faites la table de transision des automates qui reconnaissent
  - Un nombre entre 10 et 39
  - Une dizaine (10, 20, 30...90)
  - Un nombre entre 0 et 20
  - Un nombre avec ou sans virgule
  - Un nombre pair entre 10 et 999
  - Une heure au format `HH:MM`
  - Une date au format `JJ/MM/AAAA`
  - Une adresse postale

Plan

Types d'automates

- Terminologie
  - **FSA** : Finite State Automata
  - **DFA** : Deterministic Finite state Automata
  - **NFA** : Non-deterministic Finite state Automata
  - **ϵ-NFA** : NFA avec transitions ϵ

NFA

- Fonction de transition renvoit un ensemble
  $\delta : Q \times \Sigma \rightarrow Q^*$
- Dans les représentations
  - Diagramme : plusieurs transitions pour un état / symbole
  - Table : une case peut contenir plusieurs états

$\Rightarrow$ Comment reconnaître tous les mots se terminant par ab ?

- Fonction de transition étendue $\delta^* : Q \times \Sigma^* \rightarrow Q^*$
  - Forme générale : $\delta^*(q, w)$ avec $w \in \Sigma^*$
    - Si $w = x.a$ avec $a \in \Sigma$

    $\Rightarrow$ Retourner $\cup_{r \in \delta*(q,x)} \delta(r, a)$

    $\Rightarrow$ Union des états atteints selon les états précédents
    - Si $w = a$ avec $a \in \Sigma$

    $\Rightarrow$ Retourner $\delta(q, a)$
- Langage reconnu : $L = \{w | \delta^*(q_0, w) \cap F \neq \emptyset\}$

Exercice

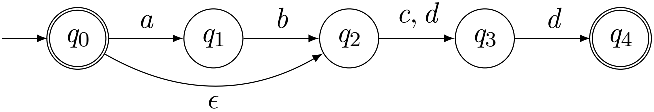
- Dessinez un automate qui reconnaît des codes couleurs
  - Avec 3 chiffres héxadécimaux (par ex. `AA0318`)
  - Avec 3 chiffres de 0 à 256 en décimal (par ex. `138027250`)
- Exécutez l'automate pour indiquer les états atteints pour
  - `FFFFFF`
  - `524317`
  - `005132075`
  - `213A5B`

Exercice

- Dessinez les DFA et NFA sur $\Sigma : \{a, b, c, d\}$ qui reconnaissent
  - Les mots contenant ad suivi d'au moins un c suivi de da
  - Les mots dont l'avant dernière lettre est un b ou un d

ϵ-NFA

- Fonction de transition peut renvoyer ϵ
  $\delta : Q \times \Sigma \cup \{\epsilon\} \rightarrow Q^*$
- Extension des NFA
  - Facilité de transition : ϵ
  - Exemple : reconnaître le langage $\{abcde, abdde, cd, dd\}$



- Calcul de la fermeture transitive d'un état
  - Initialisation : $\forall q_i \in Q,\ q_i \in fermeture(q_i)$
  - Itération : $\forall q_i \in Q$, si $\delta(q_i, \epsilon) = R$ alors $R \subset fermeture(q_i)$

  $\Rightarrow$ $fermeture(q_i)$ contient tous les états atteints avec ϵ
- Fonction de transition étendue sur
  $\delta^*(q, x.a) = \cup_{r \in \delta^*(q,x)} \cup_{s \in \delta(r,a)} fermeture(s)$

Exercice

- Dessinez un automate qui reconnaît des dates
  - Des années seules (par ex. `2010`)
  - Des mois / années (par ex. `11/2020`)
  - Des jours / mois / années (par ex. `21/05/2010`)
  - Les jours et les mois peuvent être sans 0
- Exécutez l'automate pour indiquer les états atteints pour
  - `2017`
  - `1193`
  - `05/1980`
  - `12/1995`
  - `31/10/1979`
  - `3/5/2015`
  - `10/10/2010`

Plan

Propriétés des automates et langages réguliers

- Pour un NFA, il existe un DFA équivalent

⇒ États comme sous-ensembles d'états

- Le **complémentaire** d'un langage régulier est régulier

⇒ Inversion états initial / finaux

- La **fermeture** d'un langage régulier est régulière

⇒ Transitions ϵ finaux vers initial

- L'union de deux langages réguliers est régulière
- Le produit de deux langages réguliers est régulier
- L'intersection de deux langages réguliers est régulier
- Par **homomorphisme**, un langage reste régulier

⇒ Homorphisme : $f(x * y) = f(x) \star f(y)$

Compléments

- **Dérivation** comme transition entre configurations
  - **Configuration** : état et mot à lire (q, w)
  - **Dérivation** : $(q, w) \rightarrow (q', w')$

  $\Rightarrow$ Transition avec $w = a.w'$ par $\delta(q, x) = q'$
- Calcul du **langage généré** à partir d'un état
  - Fonction récursive
    $L(q) = \cup_{a \in \Sigma} a.L(\delta(q, a)) \cup f(q)$
    avec $f(q) = \epsilon$ si $q \in F$

  $\Rightarrow$ L'ensemble peut être de cardinalité infinie

TP Unitex

- Unitex
  - Disponible en ligne sur le site de l'IGM
  - Logiciel à base d'automates (dictionnaires, grammaires, etc.)
- Travail sur un projet à définir par promotion
  - 2015/2016 : textes de loi (codes : civil / pénal / travail)
  - 2016/2017 : textes littéraires historiques
  - 2017/2018 : arguments, avis, opinions
  - ...