Automates - CM5

Clément Agret
clement.agret@cyu.fr

CY Cergy Paris Université

Récapitulatif

- Un automate est une machine informatique mathématique idéalisée.
- Un langage est un ensemble de chaînes, une chaîne est une séquence (finie) de caractères, et un caractère est un élément d'un alphabet.
- Objectif : Déterminer dans quels cas nous pouvons construire des automates pour des langages particuliers.

Plan du cours

Construction d'automates pour un langage donné

Transformation d'automates

Passage d'un AEF (automate états finis) vers grammaire

Si $\mathcal{L}$ est un langage régulier alors il existe un automate d'états finis $\mathcal{A}$ acceptant $\mathcal{L}$.

$\mathcal{A} = (\mathcal{Q}, \Sigma, \delta, q_0, F)$ et $G = (T, N, S, P)$

$T = \Sigma$ $\quad N = \mathcal{Q}$ $\quad S = q_0$

Exemple :

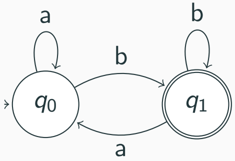

Figure 1: Automate A *déterministe* sans $\epsilon-$transitions

On définit la grammaire $G = (T, N, S, P)$ associée à cet automate 1.

$T = \{a, b\}$, $N = \{q_0, q_1\} = \{S, A\}$, $S = q_0$

D'après l'AEF, on a : $q_0 \rightarrow$ a q_0 / b q_1, $q_1 \rightarrow$ b q_1 / a q_0 , $q_1 \rightarrow \epsilon$

Donc, on aura $P = S \rightarrow aS/bA, A \rightarrow bA/aS/\epsilon$

Passage de grammaire vers un AEF (automate états finis)

Soit une grammaire régulière $G = (T, N, S, P)$, on veut construire l'automate $\mathcal{A} = (\mathcal{Q}, \Sigma, \delta, q_0, F)$, associé à G.

$\Sigma = T, \qquad \mathcal{Q} = N \cup \{f\} \qquad F = \{f\}, \qquad q_0 = S$

δ:

si B$\rightarrow$ wC alors δ (B,w) = C

si B$\rightarrow$ w alors δ (B,w) = f

Exemple

Soit G une grammaire définie par :
P =S→aB/aC, B→bS/a, C→bS/aC/b

L'automate associé à G sera donc $\mathcal{A} = (\mathcal{Q}, \Sigma, \delta, q_0, F)$ tel que :
$\Sigma = T = \{a, b\}$ $\quad \mathcal{Q} = \{S, B, C, f\}$ $\quad F = \{f\}$

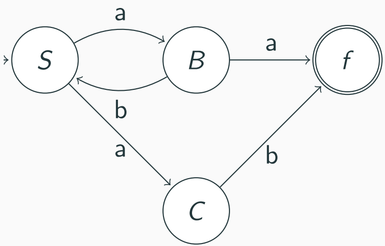

Figure 2: Automate A *non-déterministe* avec ϵ−transitions

Plan du cours

Construction d'automates pour un langage donné

Transformation d'automates

Cette construction pour transformer un NFA (Automate non deterministe) en un DFA (automate déterministe) est appelé la construction du sous-ensemble.

- Chaque état du DFA est associé à un ensemble d'états du NFA.
- L'état de début dans le DFA correspond à l'état de début du NFA, plus tous les États accessibles via ϵ-transitions.
- Si un état q dans le DFA correspond à un ensemble d'états S dans le NFA, alors la transition de l'état q sur un caractère a suit un ensemble de règles.

Ensemble des règles :
– Soit S' l'ensemble des états de la NFA qui peuvent être atteints en suivant une transition étiquetée a à partir de l'un des états de S. (Cet ensemble peut être vide.)
– Soit S'' l'ensemble des états de la NFA accessibles à partir d'un état en S' par zéro ou plusieurs transitions epsilon.
– L'état q dans les transitions DFA sur a vers un état DFA correspondant à l'ensemble d'états S"

Lors de la conversion d'un NFA en DFA, les états DFA correspondent à des ensembles d'états du NFA.
Propriété : $|\phi(S)| = 2^{|S|}$ pour n'importe quel finie ensemble S.

Dans le pire des cas, la construction peut aboutir à un DFA exponentiel plus grand que le NFA d'origine.

Défi intéressant : Trouver une langue pour lesquels ce comportement dans le pire des cas se produit (il y en a infiniment!)

Étant donné un automate $\mathcal{A} = (\mathcal{Q}, \Sigma, \Delta, 0, F)$ non déterministe avec ϵ-transition, construire un automate déterministe $\mathcal{A}'$, complet et sans *epsilon*, acceptant le même langage.

Exemple introductif

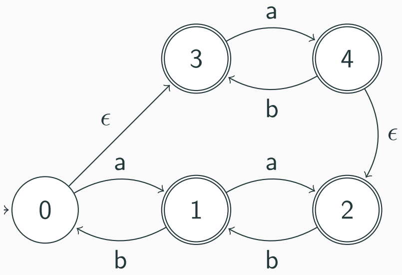

Figure 3: Automate A *non-déterministe* avec $\epsilon-$transitions