

Handout 6

Le Lemme d'Arden, et l'Algorithme de Brzozowski et McCluskey

1 Le Lemme d'Arden et application aux automates

1.1 Le lemme d'Arden

1. La solution la plus petite de l'équation

$$L = A \cdot L \cup B$$

où A et B sont des langages et L une variable, est A^*B .

2. Si $\epsilon \notin A$ alors A^*B est même la seule solution de cette équation.

Quand tous les langages sont rationnels on se permet la syntaxe des expressions rationnelles (+ à la place de \cup etc.).

1.2 Application

Le lemme d'Arden est utile pour transformer un automate en expression rationnelle. Étant donné un AFD (S, Q, q_0, F, δ) , on écrit un système d'équations :

$$L_q = \sum_{a \in S} a L_{\delta(q,a)} \quad \text{si } q \in Q - F$$

$$L_q = \sum_{a \in S} a L_{\delta(q,a)} + \epsilon \quad \text{si } q \in F$$

Ce système décrit les langages $L_q = \{w \in \Sigma^* \mid \delta^*(q, w) \in F\}$. Pour trouver la solution la plus petite on transforme ce système en un système équivalent où l'équation pour L_{q_0} n'a plus de variables sur son côté droit. On se sert de

- remplacer la variable L partout par e quand on a une équation $L = e$ où L ne paraît pas dans e ;
- remplacer la variable L par sa solution selon Arden quand on a une équation $L = e$ où L paraît dans e ;
- Simplifier les expressions rationnelles tant qu'on peut.

2 L'algorithme de Brzozowski et McCluskey

Il s'agit d'un autre algorithme pour transformer un automate en expression rationnelle.

2.1 Les automates généralisés

Pendant les transformations on manipule un *automate généralisé*, c'est un automate qui a des expressions rationnelles sur les flèches. On peut donc représenter les transitions d'un tel automate généralisé par une fonction qui à chaque état q et chaque état p associe une expression rationnelle $E_{q,p}$. Quand une transition est étiquetée par \emptyset on ne la représente pas dans le dessin.

Un tel automate généralisé *accepte* un mot $w \in \Sigma^*$ quand il existe des états q_0, \dots, q_n et une *factorisation* du mot

$$w = w_0 \cdots w_{n-1}$$

tel que

- q_0 est un état initial ;
- q_n est un état acceptant ;
- pour tout i avec $0 \leq i < n$: $w_i \in \mathcal{L}(E_{q_i, q_{i+1}})$

2.2 L'algorithme de Brzozowski et McCluskey

Cet algorithme transforme un automate en expression rationnelle, par élimination des états. L'invariant maintenu par les transformation est

- il y a un seul état initial ;
- aucune transition entre dans l'état initial ;
- il y a un seul état acceptant ;
- aucune transition sort de l'état acceptant ;
- l'état initial n'est pas l'état acceptant.

Il faut d'abord assurer que l'automate de départ satisfait tous ces propriétés, si ce n'est pas le cas ajouter des nouveaux uniques états initial et acceptant avec les ϵ -transitions qu'il faut.

Pour éliminer un état p , on remplace pour tous états q, r l'expression $E_{q,r}$ par $E_{q,r} + E_{q,p}E_{p,p}^*E_{p,r}$. On peut à chaque étape choisir l'état qu'on veut éliminer, sauf l'état initial et sauf l'état acceptant. À la fin on obtient un automate généralisé avec seulement 2 états (l'état initial et l'état acceptant), l'expression sur la flèche de l'état initial vers l'état acceptant est l'expression cherchée.