

Chapitre V : Minimisation

Goulven GUILLOU

UBO – Département Informatique



1 / 10

Il est difficile de comparer deux expressions régulières :
représentent-elles le même langage ?

Il est (de même) difficile de comparer deux automates : sont-ils
équivalents ?

Dans ce chapitre on va montrer qu'il est possible de construire un
représentant canonique pour toute classe d'équivalence qu'on
appellera automate minimal.

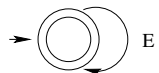
Par convention ce représentant sera déterministe et complet (un
afdc : on a vu que c'était toujours possible).

Il sera également connexe (ou accessible), c'est-à-dire sans état non
accessible.

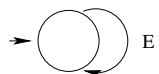
2 / 10

Le représentant canonique est minimal au sens du nombre d'états.

Ainsi :



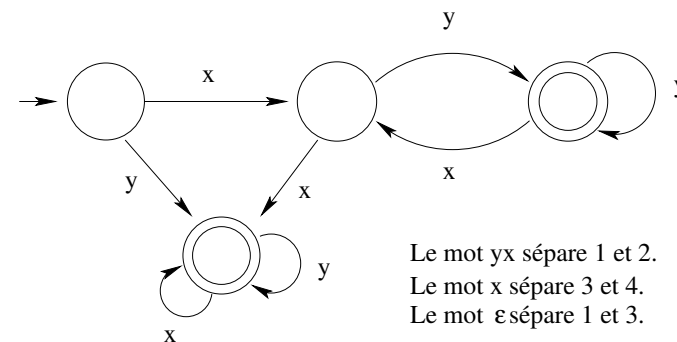
est le représentant canonique de E^* .



est le représentant canonique de \emptyset .

3 / 10

Définition Soit $\mathcal{A} = (E, Q, q_0, F, \delta)$ un afdc. Un mot $u \in E^*$ sépare
deux états s et t dans \mathcal{A} si $\delta(s, u)$ et $\delta(t, u)$ ne sont pas de même
nature vis-à-vis de l'acceptation.



Le mot yx sépare 1 et 2.
Le mot x sépare 3 et 4.
Le mot ε sépare 1 et 3.

4 / 10

Définition Soit $\mathcal{A} = (E, Q, q_0, F, \delta)$ un afdc. Deux états s et t sont dits équivalents ($s \sim t$), si aucun mot $u \in E^*$ ne les sépare.

Définition $\bar{\mathcal{A}} = (E, \bar{Q}, \bar{q}_0, \bar{F}, \bar{\delta})$ est l'automate quotient de \mathcal{A} avec :

- \bar{q} est la classe d'équivalence de q .
- \bar{Q} classes d'équivalence de Q modulo \sim .
- $\bar{\delta}$ définie par :

$(\bar{q}, x, \bar{q}') \in \bar{\delta} \Leftrightarrow \exists q \in \bar{q}, \exists q' \in \bar{q}' \text{ tels que } (q, x, q') \in \delta.$

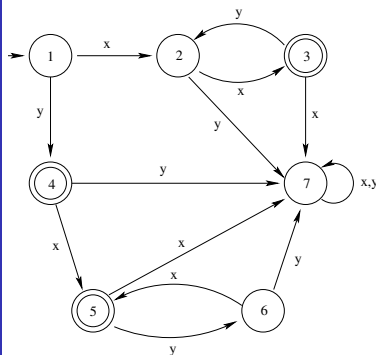
Proposition 1 L'automate \mathcal{A} est un afdc qui reconnaît le même langage que $\bar{\mathcal{A}}$.

Proposition 2 Soit \mathcal{A} un afdc, on peut toujours construire l'automate quotient $\bar{\mathcal{A}}$.

Définition Pour $i \in \mathbb{N}$, on définit la relation \equiv_i sur Q par :

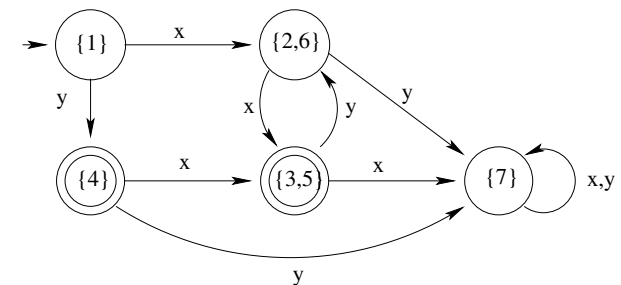
$$s \equiv_i t \iff \forall u \in \bigcup_{0 \leq j \leq i} E^j (\delta(s, u) \in F \text{ et } \delta(t, u) \in F) \text{ ou } (\delta(s, u) \notin F \text{ et } \delta(t, u) \notin F)$$

Lemme Si $\equiv_{i+1} = \equiv_i$ alors $\forall k \in \mathbb{N}, \equiv_{i+k} = \equiv_i$ et donc $\sim = \equiv_i$.



| ε | x | y | xx | xy | yx | yy |
|---------------|---|---|----|----|----|----|
| 1 | 2 | 4 | | | | |
| 2 | 3 | 7 | 7 | 2 | | |
| 6 | 5 | 7 | 7 | 6 | | |
| 7 | 7 | 7 | | | | |
| 3 | 7 | 2 | | | 3 | 7 |
| 4 | 5 | | | | | |
| 5 | 7 | 6 | | | 5 | 7 |

Avant d'appliquer l'algorithme de Moore il faut avoir un automate fini déterministe et complet (afdc).



Définition Le résiduel du langage L par rapport à u est le langage :

$$u^{-1}L = \{v \in E^* \mid uv \in L\}$$

Proposition 3 Si L est un langage rationnel, il possède un nombre fini de résiduels.

Proposition 4 Si un langage possède un nombre fini de résiduels alors il est rationnel.

Proposition 5 Soit L un langage rationnel. Il existe un afdc unique, à isomorphisme près, qui reconnaît L et possède un nombre minimum d'états, et on peut le construire à partir d'un automate fini quelconque reconnaissant L .

Corollaire 1 Etant donnés deux automates finis quelconques, on peut décider s'ils reconnaissent le même langage.

Corollaire 2 Etant données deux expressions rationnelles, on peut décider si ce sont deux expressions du même langage.