Chapitre 5: Minimisation

Pourquoi?

- Moins un automate contient d'états, moins il prendra du temps à reconnaître un mot
- Il prendra moins d'espace en mémoire s'il est question de le sauvegarder.
- ✓ Il est donc logique de vouloir minimiser ce temps en essayant de réduire le nombre d'états. Si on peut trouver une multitude d'automates pour reconnaître le même langage, on ne peut trouver néanmoins qu'un seul automate minimal reconnaissant le même langage.
- ✓ Comment? La minimisation s'effectue en éliminant les états dits inaccessibles et en confondant(ou fusionnant) les états reconnaissant le même langage.

Définition d’un AEF minimal

- Un AEF déterministe est minimal (canonique) si et seulement si tout autre AEF déterministe reconnaissant le même langage a au moins le même nombre d’états.

Les états inaccessibles

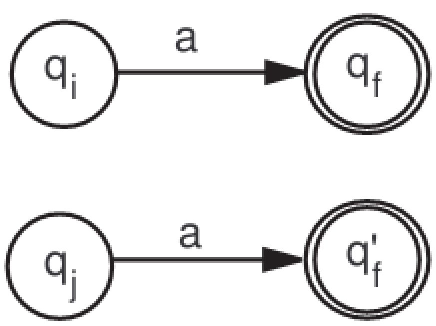
Un état est dit inaccessible s'il n'existe aucun chemin permettant de l'atteindre à partir de l'état initial.

les états inaccessibles sont improductifs, c'est-à-dire qu'ils ne participeront jamais à la reconnaissance d'un mot. Ainsi, la première étape de minimisation d'un AEF consiste à éliminer ces états.

Les états β-équivalents

- Deux états qi et qj sont dits β-équivalents s'ils permettent d'atteindre les états finaux en utilisant les mêmes mots. On écrit alors :qi β qj

- L'algorithme de minimisation consiste donc à fusionner simplement ces états pour n'en faire qu'un.



Remarque

- La relation β-équivalence est dite une relation de congruence.

- Le nombre de classes d'équivalence de la relation β-équivalence est égal au nombre des états de l'automate minimal car les état de chaque classe d'équivalence reconnaissent le même langage (ils seront fusionnés).

Minimiser un AEF

La méthode de réduction d'un AEF est la suivante :

1. Nettoyer l'automate en éliminant les états inaccessibles ;
2. Regrouper les états congruents (appartenant à la même classe d'équivalence).

Algorithme : Regroupement des états congruents

- **Remarque**: cet algorithme ne peut s'appliquer que si l'automate est **déterministe**.

1- Faire deux classes : A contenant les états finaux et B contenant les états non finaux ;

2- S'il existe un symbole a et deux états q_i et q_j d'une même classe tel que $\delta(q_i, a)$ et $\delta(q_j, a)$ n'appartiennent pas à la même classe, alors créer une nouvelle classe et séparer q_i et q_j. On laisse dans la même classe tous les états qui donnent un état d'arrivée dans la même classe ;

3- Recommencer l'étape 2 jusqu'à ce qu'il n'y ait plus de classes à séparer ;

Algorithme (suite)

Les paramètres de l'automate minimal sont, alors, les suivants :

– Chaque classe de congruence est un état de l'automate minimal ;

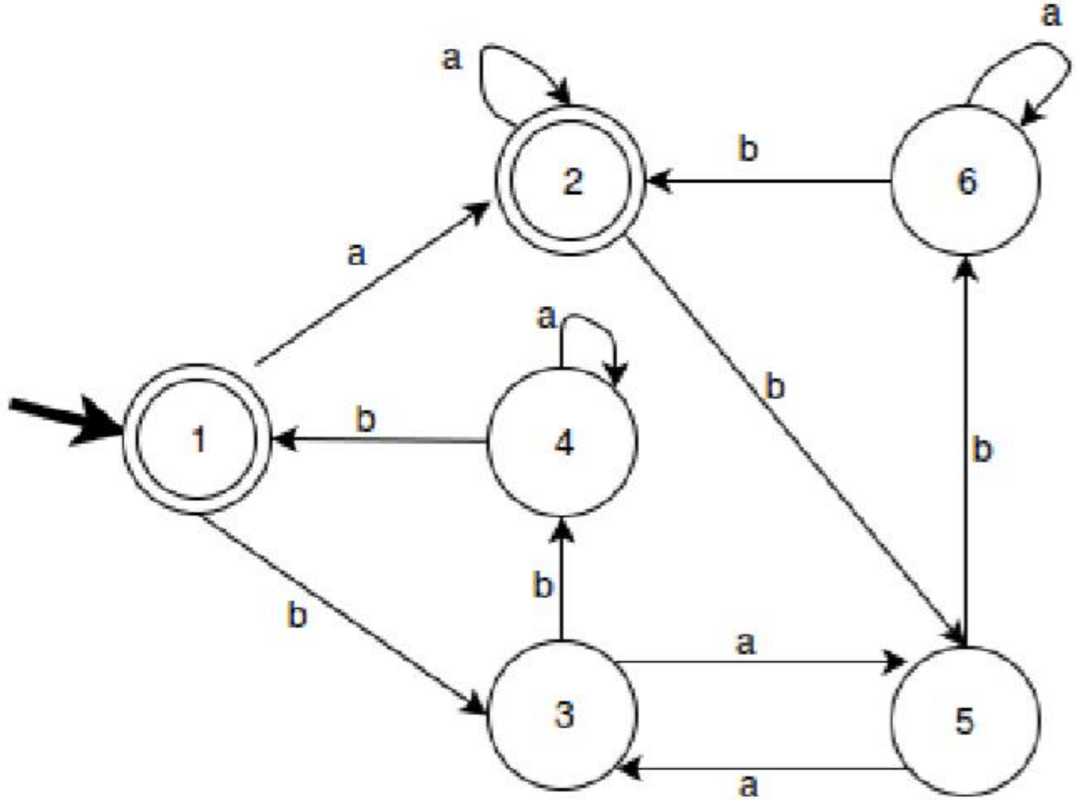
– La classe qui contient l'ancien état initial devient l'état initial de l'automate minimal ;

– Toute classe contenant un état final devient un état final ;

– La fonction de transition est définie comme suit : soient A une classe de congruence obtenue, a un symbole de l'alphabet et qi un état qi $\in$A tel que δ(qi,a) est définie. La transition δ(A,a)est égale à la classe B qui contient l'état qj tel que δ(qi,a) =qj.

Exemple

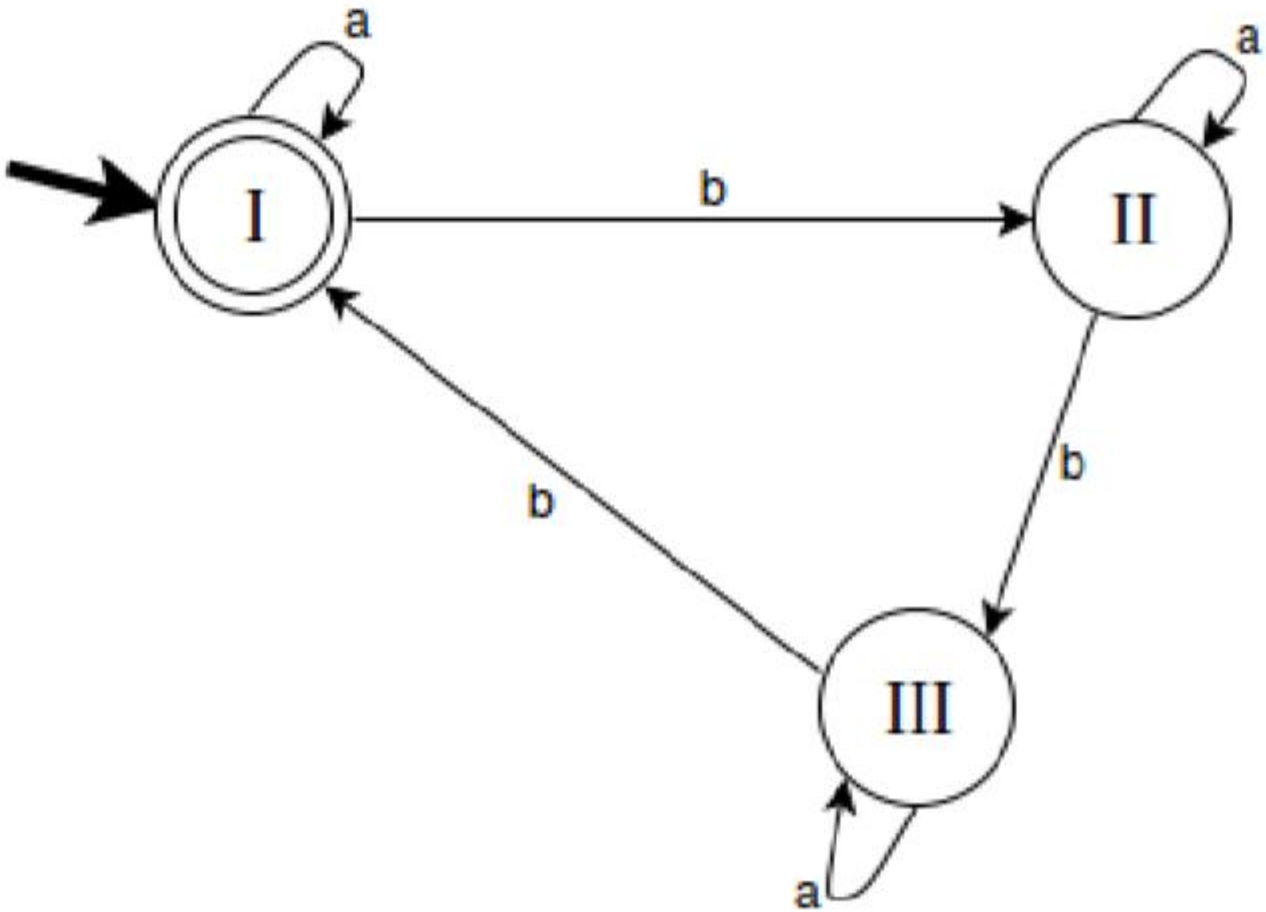
- Trouver l'AEF minimal de l'AEF représenté sur la figure suivante:



Construction de la relation β_i

	1	2	3	4	5	6
B_0	I	I	II	II	II	II
a	I	I	II	II	II	II
b	II	II	II	I	II	I
B_1	I	I	II	III	II	III
a	I	I	II	III	II	III
b	II	II	III	I	III	I
B_2	I	I	II	III	II	III

$$I = \{1,2\}\,, II = \{3,5\}\,, III = \{4,6\}$$


a
a
b
I
II
b
b
III
a

Exemple

Soit à minimiser l'automate suivant (les états finaux sont les états 1 et 2 tandis que l'état 1 est initial) :

État	a	b
1	2	5
2	2	4
3	3	2
4	5	3
5	4	6
6	6	1
7	5	7