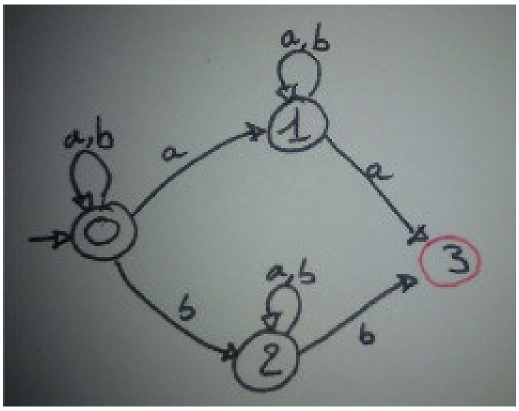
AF4

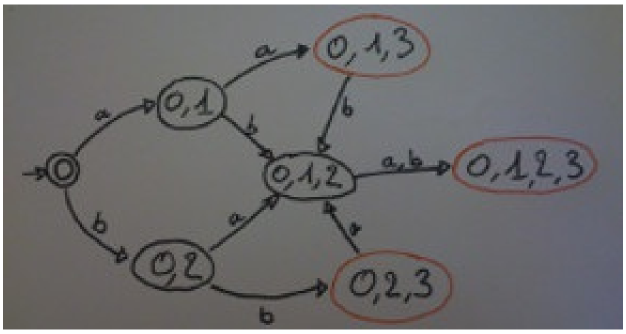
1) Determinisation :

L'état **Initial** est un état qui possède tous les numéros des états initiaux de l'automate

Les états **finaux** sont les états qui ont **au moins un** numéro d'un état final de l'automate

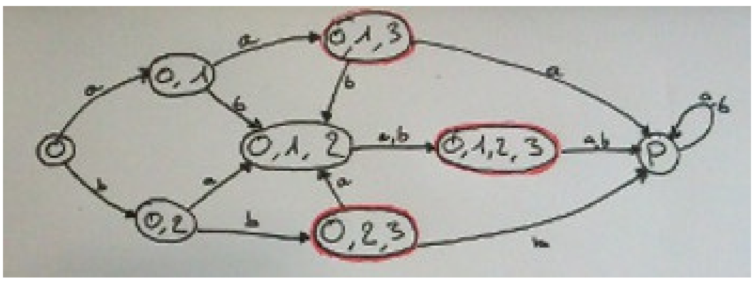
Etat	a	b
0	0,1	0,2
1	1,**3**	1
2	2	2,**3**
3	/	/





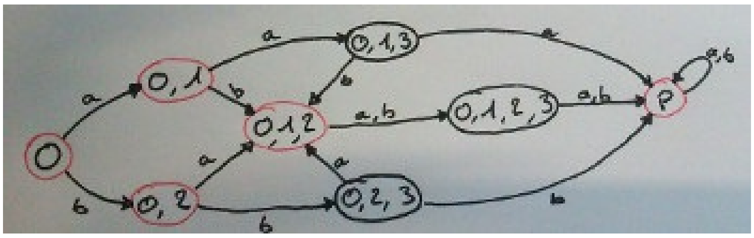
2) Automate Fini Complet Deterministe

Pour tansformer un AFD en un AFCD il faut ajouter un état "Poubelle" vers qui toutes les transitions manquante des autres états vont pointer



3) Complémentaire

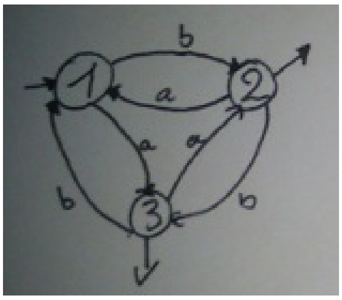
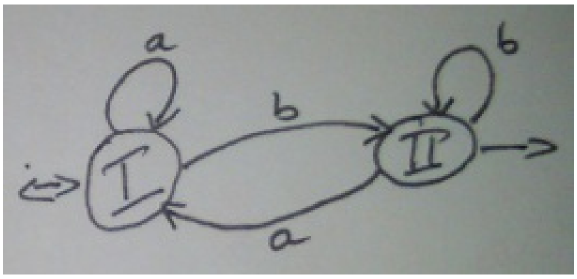
L'automate doit être complet. Tout les états finaux deviennent non-finaux et invrsement.



4) Mirroir

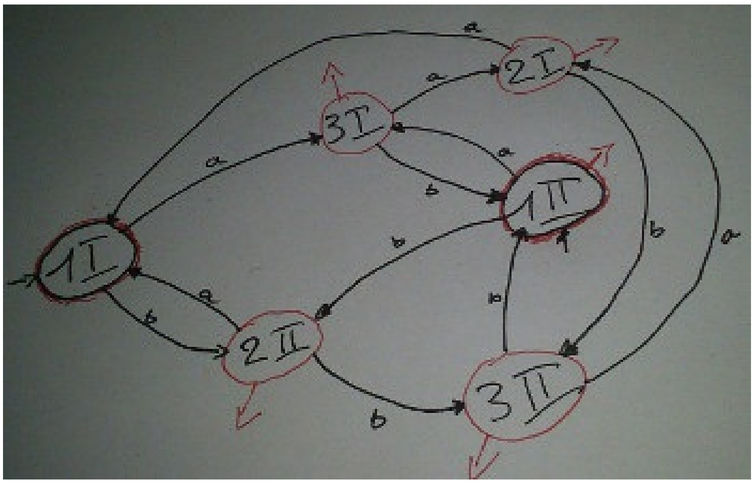
- On inverse les transitions
- États Initiaux => Terminaux
- États Terminaux => Initiaux

5) Union et Inter

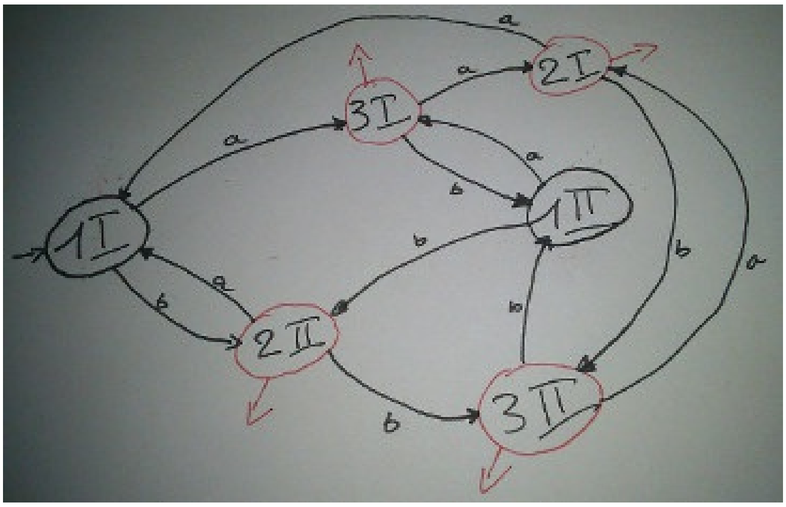
- Union

	I		II	
	a	b	a	b
1	**3 I**	**2 II**	**3 I**	**2 II**
2	**1 I**	**3 II**	**1 I**	**3 II**
3	**2 I**	**1 II**	**2 I**	**1 II**



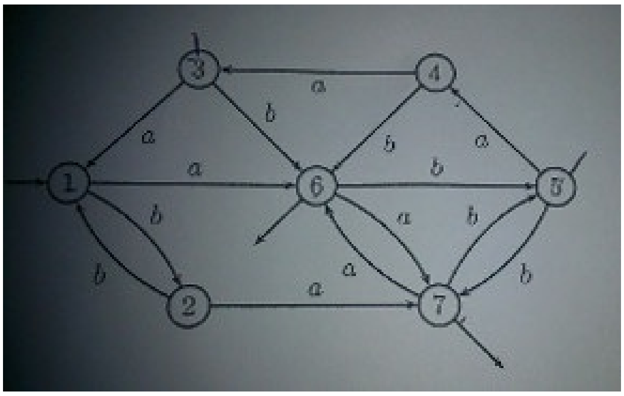
- Inter

	I		**II**	
	a	b	a	b
1	**3 I**	**2 II**	**3 I**	**2 II**
2	1 I	**3 II**	1 I	**3 II**
3	**2 I**	1 II	**2 I**	1 II



7) Moore

Minimisation d'un Automate

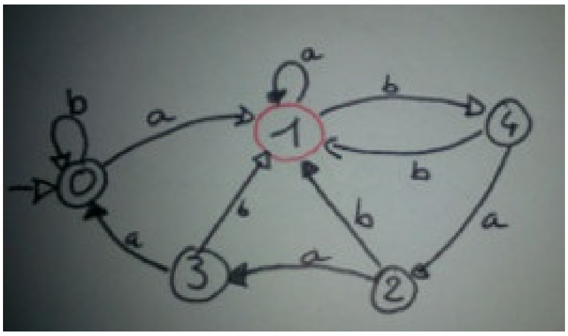


Etat	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Groupe	0	0	0	0	0	1	1
T	1 0	1 0	0 1	0 1	0 1	1 0	1 0

Groupe	**0**	**0**	**2**	**2**	**2**	**1**	**1**
T	1 0	1 0	0 1	**2** 1	**2** 1	1 **2**	1 **2**

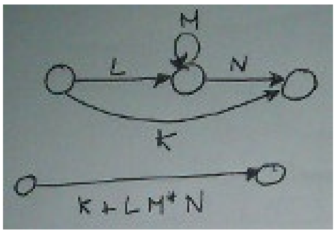
Groupe	0	0	3	2	2	1	1
T	1 0	1 0	0 1	**3** 1	2 1	1 2	1 2

Groupe	0	0	3	2	4	1	1
T	1 0	1 0	0 1	3 1	2 1	1 **4**	1 **4**



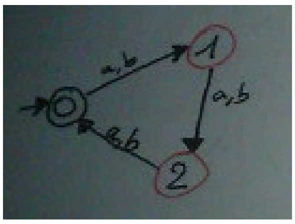
8) Mc Naughton et Yamada

Contruire une expression rationnelle à partir d'un automate

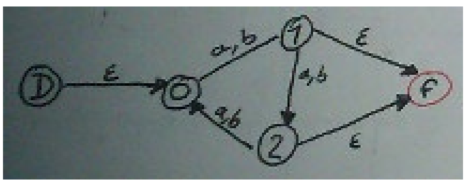


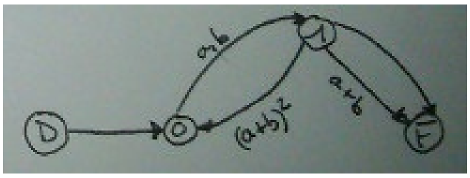
Règle de suppresion d'un état

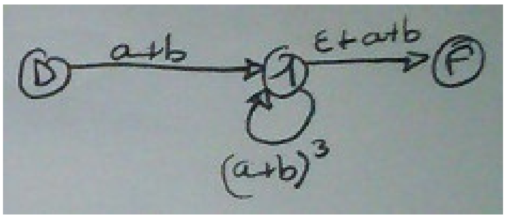
Soit `A` l'automate suivant:

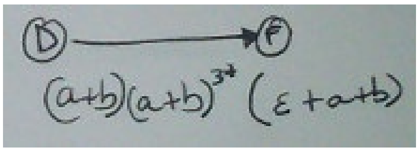


- On ajoute a l'automate un état `I` initial auquel on ajoute des epsilon transition vers les états initiaux et un état `F` Final ver qui vont pointer tout les états finaux avec des épsilon transitions



- Suppression de l'état `2` 

- Suppression de l'état `0` 

- Suppression de l'état `1` 

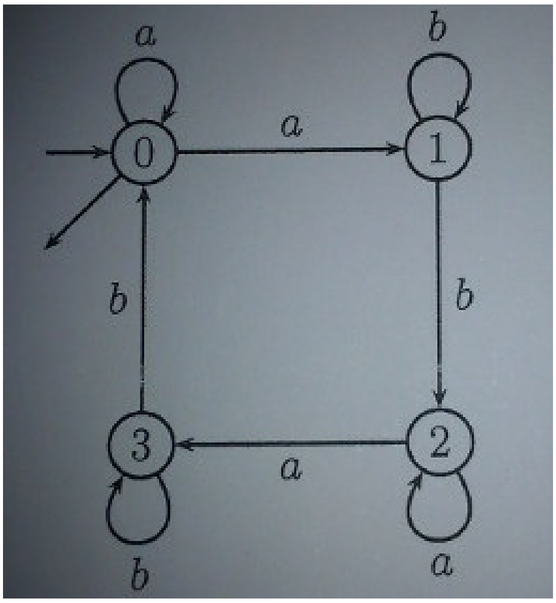
9) Équation linéaire droite

Contruire une expression rationnelle à partir d'un automate

Lemme d'Arden

L = X.L+Y => L = X*.Y

Lautomate doit être deterministe




```
L0 = a.L0 + a.L1 + ε
L1 = b.L1 + b.L2
L2 = a.L2 + a.L3
L3 = b.L3 + b.L0
```

On ne cherche que les états initiaux. S'il y a plusieurs états initiaux on fera l'union

```
L3 = b*.(b.L0)`  Lemme d'Arden
```

On remplace `L3` dans `L2` :

```
L2 = a.L2 + a.b*.(b.L0)
```

Il nous reste donc :

```
L0 = a.L0 + a.L1 + ε
L1 = b.L1 + b.L2
L2 = a.L2 + a.b*.(b.L0)
```

On applique le Lemme d'Arden sur `L2`

```
L2 = a*.a.b*.b.L0 => (Arden)
```

On remplace `L2` dans `L1`

```
L0 = a.L0 + a.L1 + ε
L1 = b.L1 + b.a*.a.b*.b.L0
```

On applique le Lemme d'Arden sur `L1`

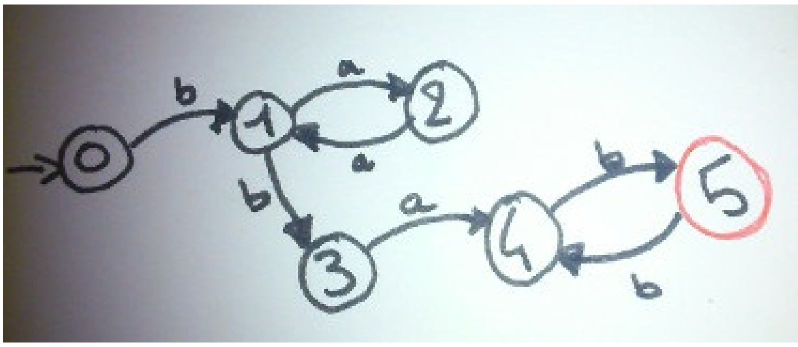
```
L1 = b*.b.a*.a.b*.b.L0 => (Arden)


L0 = a.L0 + a.b*.b.a*.a.b*.b.L0 + ε
L0 = (a.+ a.b*.b.a*.a.b*.b).L0 + ε


L0 = (a.+ a.b*.b.a*.a.b*.b)*  => (Arden)
```

10) Résiduel

`L = b.(a.a)*.b.a.(b.b)*.b`

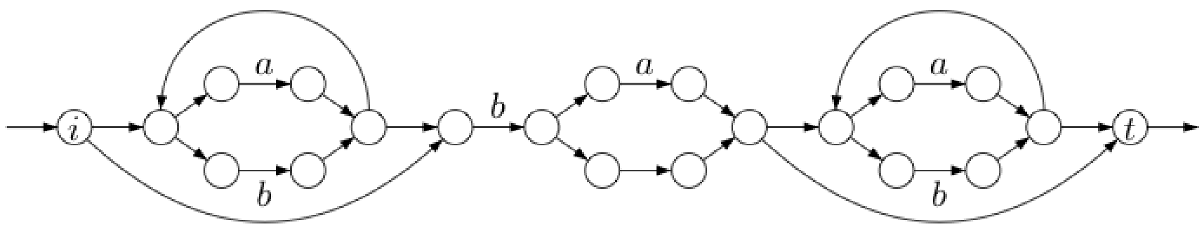
État		Expression
0	`L =`	`b.(a.a)*.b.a.(b.b)*.b`
-	`(a–¹)L =`	`Ø`
1	`(b–¹)L =`	`(a.a)*.b.a.(b.b)*.b`
2	`(ba–¹)L =`	`a.(a.a)*.b.a.(b.b)*.b`
3	`(bb–¹)L =`	`a.(b.b)*.b`

État		Expression
1	(baa–¹)L = (b–¹)L =	(a.a)*.b.a.(b.b)*.b
-	(bab–¹)L =	∅
4	(bba–¹)L =	(b.b)*.b
-	(bbb–¹)L =	∅
-	(bbaa–¹)L =	∅
5	(bbab–¹)L =	b.(b.b)*.b **+ ε**
-	(bbaba–¹)L =	∅
4	(bbabb–¹)L = (bba–¹)L =	(b.b)*.b



11) Thomson

$(a+b)^*b(a+\epsilon)(a+b)^*$



12) Glushkov

Contruire un AF à partir d'une éxpression rationnelle

ex : `L = ((a.(a.b)*.b)+(b + a.a))*`

- On remplace chaque lettre par un numéro : `L = (( 1.( 2. 3)*. 4)+( 5 +  6. 7))*`
- On rempli le tableau au ajoutant la ligne `0` qui sera l'état initial

Lettre	a	b
0	α1,α6	**α5**
α1	α2	**α4**
α2	-	α3
α3	α2	**α4**
α4	α1,α6	**α5**
α5	α1,α6	**α5**
α6	**α7**	-
α7	α1,α6	**α5**

- On dessine l'automate de la même maniere que pour la determinisation.

13) Monoïde de transition

ε	a = aaa	b = aab	aa	bb = bbb = abb = baa	ab	ba = baa = aba	bab
0	1	2	2	2	0	1	0
1	2	0	1	2	2	1	0
2	1	2	2	2	0	1	0

Soit $\{\varepsilon, a, b, aa, bb, ab, ba, bab\}$

14) Critère de cloture

15) Égalité de deux expressions rationnelles

2 methodes:

- On reduit les expressions rationnelles.
- On construit leurs Automates Minimal