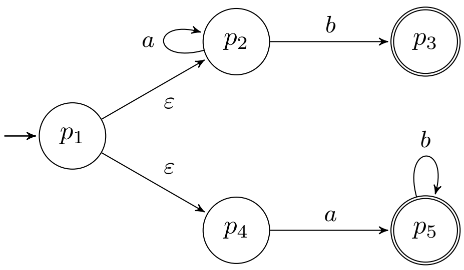
TD Théorie des langages 1 — Feuille 3

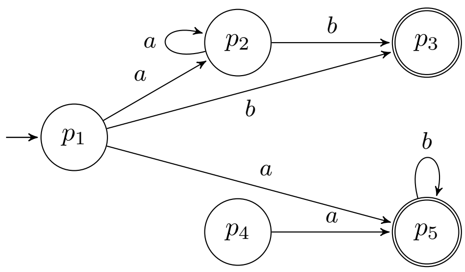
Langages réguliers – Déterminisation, Minimisation

Exercice 1 Déterminiser l'automate suivant :



Solution de l'Exercice 1. L'ensemble des états ε-accessibles depuis p_1 est $\{p_1, p_2, p_4\}$; les autres ensembles sont des singletons. On en déduit la relation de transition suivante pour l'automate obtenu après suppression des ε-transitions :

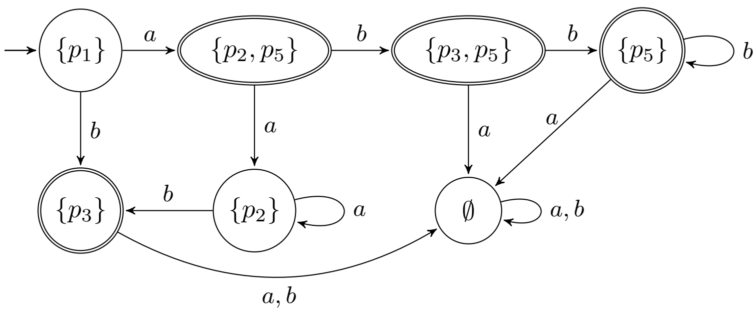
δ	a	b
p_1	p_2, p_5	p_3
p_2	p_2	p_3
p_3	-	-
p_4	p_5	-
p_5	-	p_5



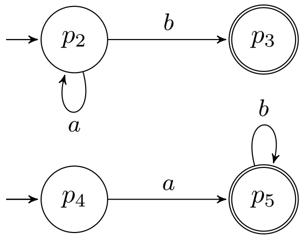
On remarque que p_4 n'est plus accessible et peut donc être supprimé, ce que la déterminisation va faire pour nous.

Déterminisation :

δ	a	b	I/F
p_1	p_2, p_5	p_3	I
p_2, p_5	p_2	p_3, p_5	F
p_3	$\emptyset$	$\emptyset$	F
p_2	p_2	p_3	
p_3, p_5	$\emptyset$	p_5	F
p_5	$\emptyset$	p_5	F
$\emptyset$	$\emptyset$	$\emptyset$	



Un des intérêts de l'exo : il est facile de décrire en français le langage reconnu sur l'automate d'origine (« autant de a qu'on veut puis un b, ou alors un a puis autant de b qu'on veut »), mais c'est plus dur sur l'AFD... Sur l'automate d'origine on voit bien l'union (avec les ε) ; on la verrait aussi bien avec



Exercice 2 On s'intéresse au langage L des mots binaires dont le dernier bit est un bit de parité. Plus précisément, un mot $wx \in \{0,1\}^*$ est dans L si le nombre de 1 dans w est impair et x vaut 1, ou si le nombre de 1 dans w est pair et x vaut 0. Autrement dit, on choisit x pour que le nombre de 1 dans wx soit pair.

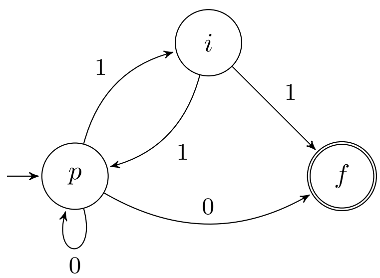
Exemples : 0, 011011, 1010 et 001111 sont dans L.

Contre-exemples : 11010, 110001 et ε ne sont pas dans L.

Construire un automate déterministe reconnaissant L.

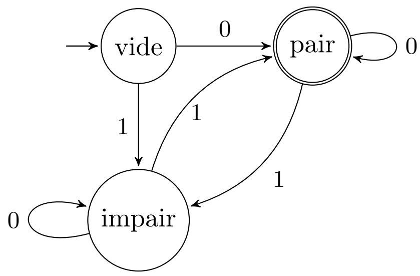
Solution de l'Exercice 2.

On commence par construire un automate qui compte la parité d'un mot w, puis on lui ajoute une transition vers un nouvel état pour le bit de parité. Ce dernier état est le seul état final.



Déterminisation :

δ	0	1	I/F
p	p,f	i	I
p,f	p,f	i	F
i	p_2	p,f	



Remarque : On peut aussi produire directement le résultat en reconnaissant les mots non vides avec un nombre pair de 1 (d'où les noms des états a posteriori dans l'AFD.)

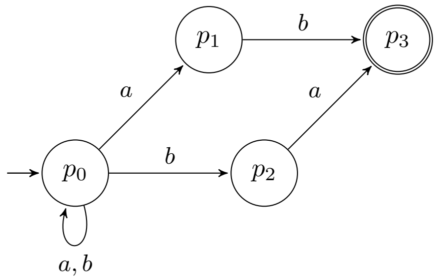
Exercice 3 Construire des automates déterministes reconnaissant les langages sur $\{a,b\}$ suivants :

1. L'ensemble des mots terminés par ab ou bien par ba.

2. L'ensemble des mots contenant au moins deux fois la séquence ab.
3. Le langage $\{aab\}^*\{b\}$.
4. Le langage $\{a\}^*\{aba\}^*$.

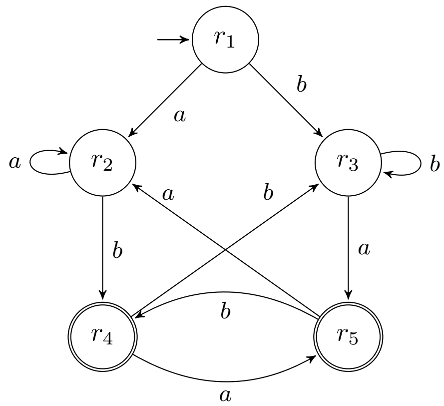
Solution de l'Exercice 3. Ici, selon leur provenance, certains ont tendance à chercher directement un automatate déterministe. Sur les questions 2 et 4 c'est facile de se tromper...

1. Automate non-déterministe :

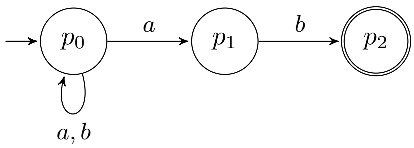


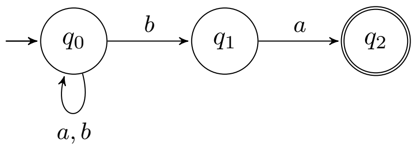
Déterminisation :

Nom	I/F	δ	a	b
r_1	I	p_0	p_0, p_1	p_0, p_2
r_2		p_0, p_1	p_0, p_1	p_0, p_2, p_3
r_3		p_0, p_2	p_0, p_1, p_3	p_0, p_2
r_4	F	p_0, p_2, p_3	p_0, p_1, p_3	p_0, p_2
r_5	F	p_0, p_1, p_3	p_0, p_1	p_0, p_2, p_3

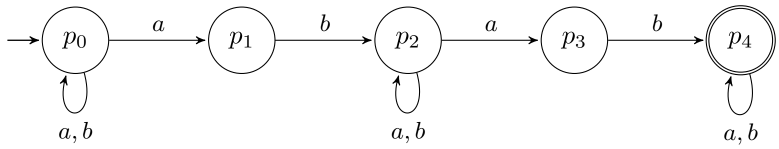


On peut aussi partir de



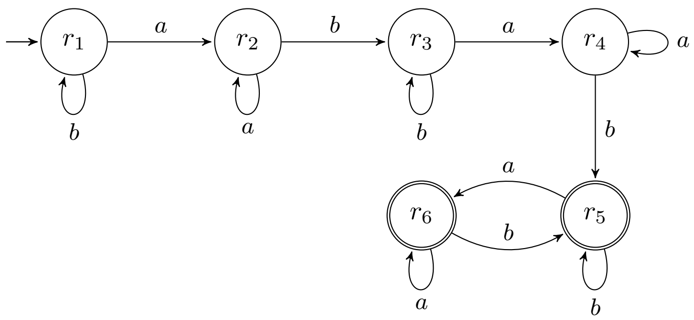


2. Automate non-déterministe :



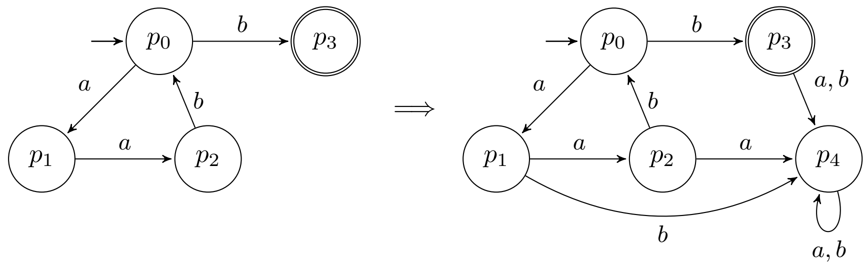
Déterminisation :

Nom	I/F	δ	a	b
r_1	I	p_0	p_0, p_1	p_0
r_2		p_0, p_1	p_0, p_1	p_0, p_2
r_3		p_0, p_2	p_0, p_1, p_2, p_3	p_0, p_2
r_4		p_0, p_1, p_2, p_3	p_0, p_1, p_2, p_3	p_0, p_2, p_4
r_5	F	p_0, p_2, p_4	p_0, p_1, p_2, p_3, p_4	p_0, p_2, p_4
r_6	F	p_0, p_1, p_2, p_3, p_4	p_0, p_1, p_2, p_3, p_4	p_0, p_2, p_4

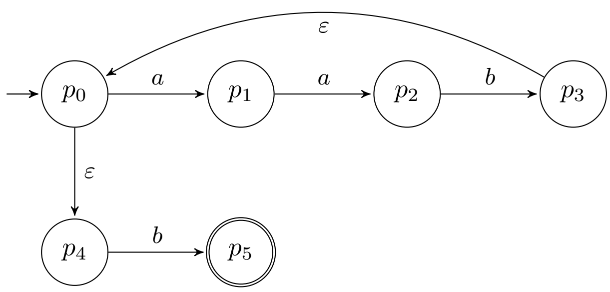


En question subsidiaire, on peut demander « exactement 2 fois la séquence ab »...

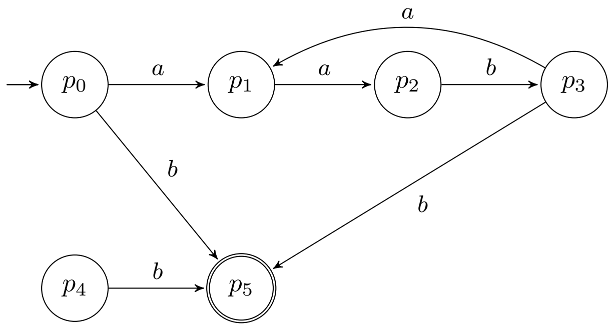
3. On peut directement construire un automate déterministe non complet, puis le compléter :



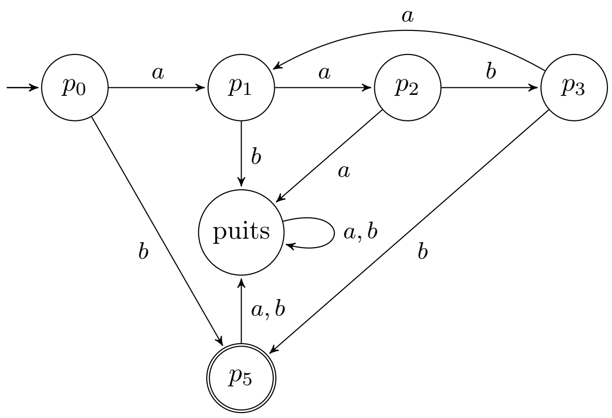
Ou bien passer par un automate non-déterministe, qu'on déterminise (et dont on supprime d'abord les ε-transitions au besoin) :



Après suppression des ε-transitions (non détaillée) :

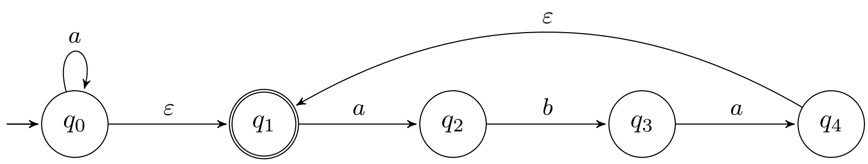


L'automate est déjà déterministe mais ni complet ni initialement connecté. On peut corriger cela en supprimant l'état p_4 pour devenir initialement connecté et en ajoutant un état puits pour devenir complet :



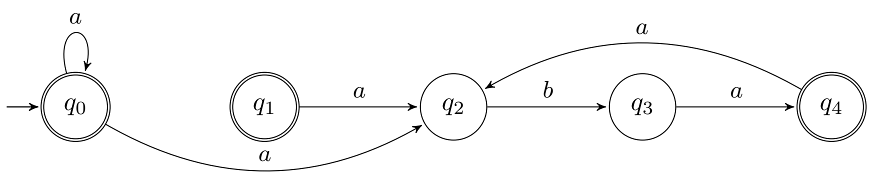
Remarquez que les automates déterministes obtenus par les deux méthodes n'ont pas le même nombre d'états mais sont équivalents.

4. Automate non déterministe :

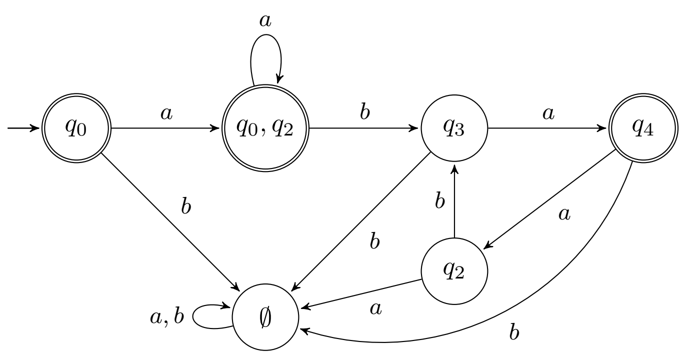


À comparer avec la question précédente : ici si on ne met pas d'ε-transition entre q_0 et q_1 ça ne marche pas... mais on peut faire directement aba en triangle (sans q_4 et son ε-transition vers q_1).

Automate non déterministe sans ε-transition :



Automate déterminisé :


a
q_0
a
q_0, q_2
b
q_3
a
q_4
b
b
b
a
q_2
a, b
$\emptyset$
a
b