**1. Lexème**

Un **lexème** est l'unité de base du lexique d'une langue. C'est une abstraction qui regroupe toutes les formes fléchies ou dérivées d'un mot, représentant son sens fondamental.

* Exemple : le lexème *manger* inclut les formes *mange*, *manges*, *mangeons*, etc.  
  En informatique, dans le contexte de l'analyse lexicale, un lexème représente une séquence de caractères correspondant à un motif (comme un mot-clé ou un identifiant) reconnu par un programme (exemple : int dans un code source C).

**2. Unité lexicale**

Une **unité lexicale** (ou **token**, en analyse lexicale informatique) est un élément concret de la langue utilisé dans un contexte donné.

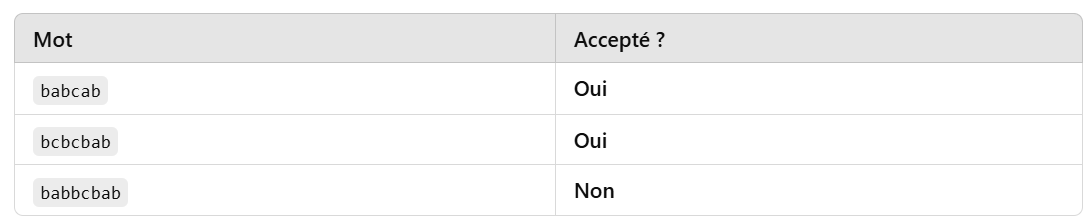
* En linguistique, c'est une occurrence d'un mot ou d'une expression dans un texte ou un discours.
* En informatique, il désigne une structure syntaxique extraite d'une chaîne de caractères lors de l'analyse lexicale. Par exemple, pour int x = 10;, les unités lexicales sont :
  + int (mot-clé),
  + x (identifiant),
  + = (opérateur),
  + 10 (constante entière).

**3. Langage**

Un **langage** est un système de communication structuré, composé de symboles et de règles qui définissent leur usage.

* **Langage naturel** : utilisé par les humains (comme le français ou l’anglais), caractérisé par des règles syntaxiques, sémantiques et pragmatiques.
* **Langage formel** : utilisé en mathématiques et en informatique, il se définit par un alphabet, une grammaire, et un ensemble de règles permettant de générer des chaînes valides (exemple : langages de programmation comme Python ou Java).

**L=(b+c)∗bc(b+c)∗(bab(b+c)∗)∗**

**~~~~**

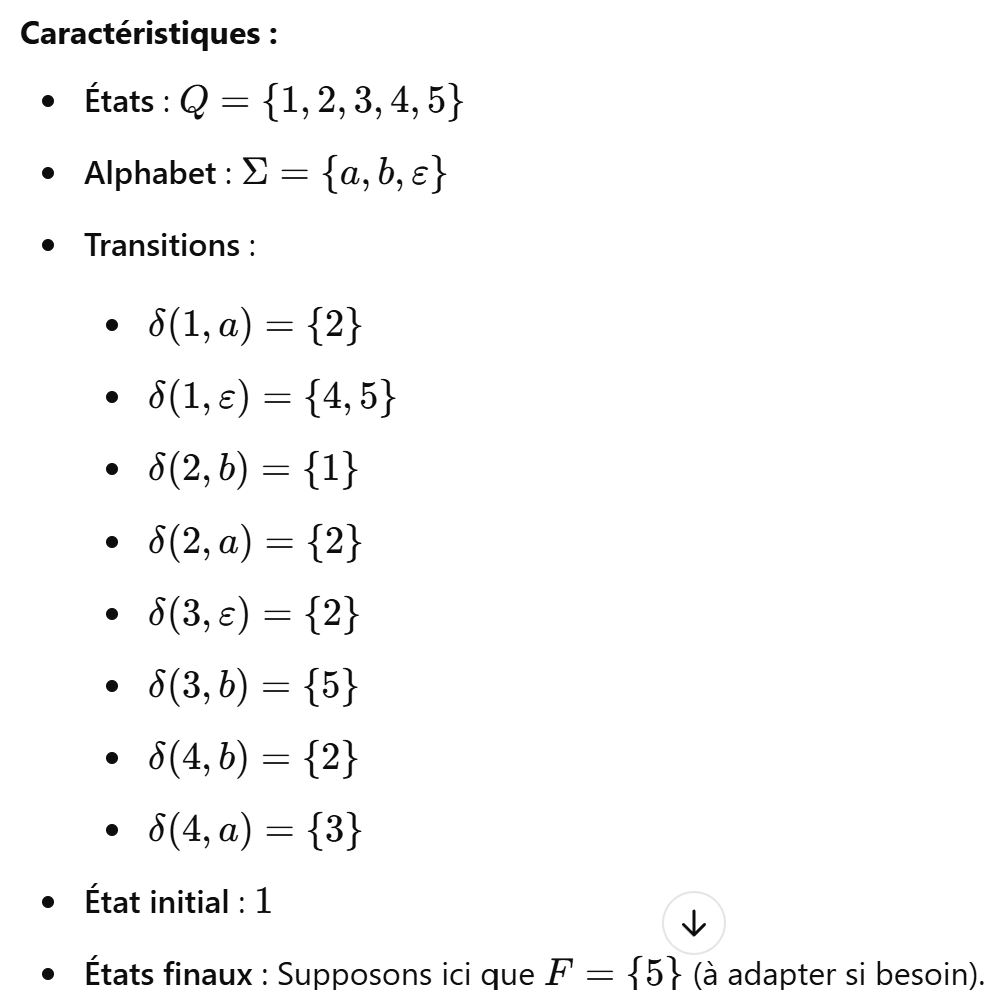
**Exercice 2 :**

On donne l’Automate fini M=< E={1,2,3,4,5}, Σ ={a,b}, q0=1, F={5}> avec les transitions suivantes :

δ(1,a)=2, δ (1,ε)=4, δ(1,ε)=5, δ(2,b)=1, δ(2,a)=2, δ(3,ε)=2, δ(3,b)=5, δ(4,b)=2, δ (4,a)=3.

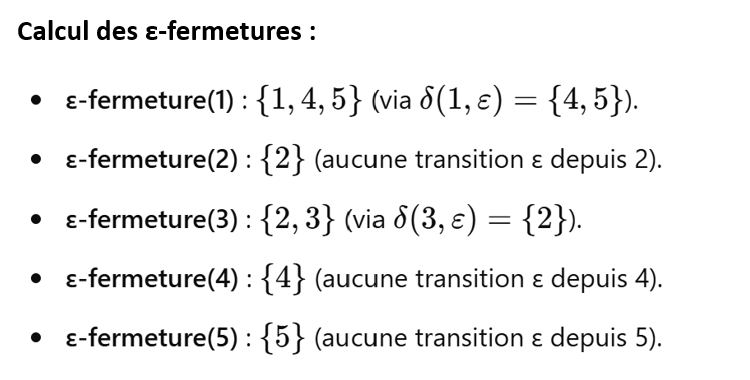
a) Construire l’automate M, puis le rendre déterministe.

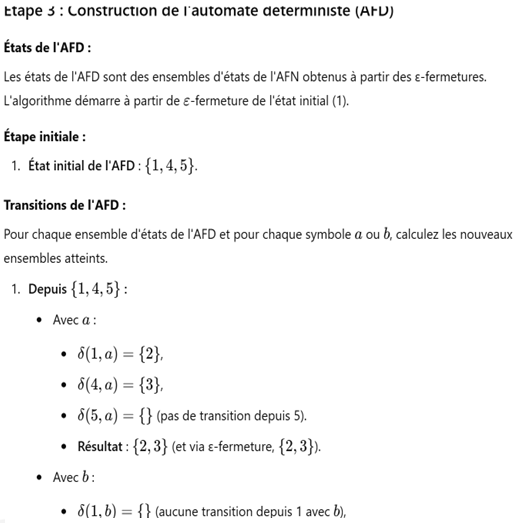
b) Donner deux mots acceptés, respectivement de longueur 2 et 4, par l’automate obtenu en a).

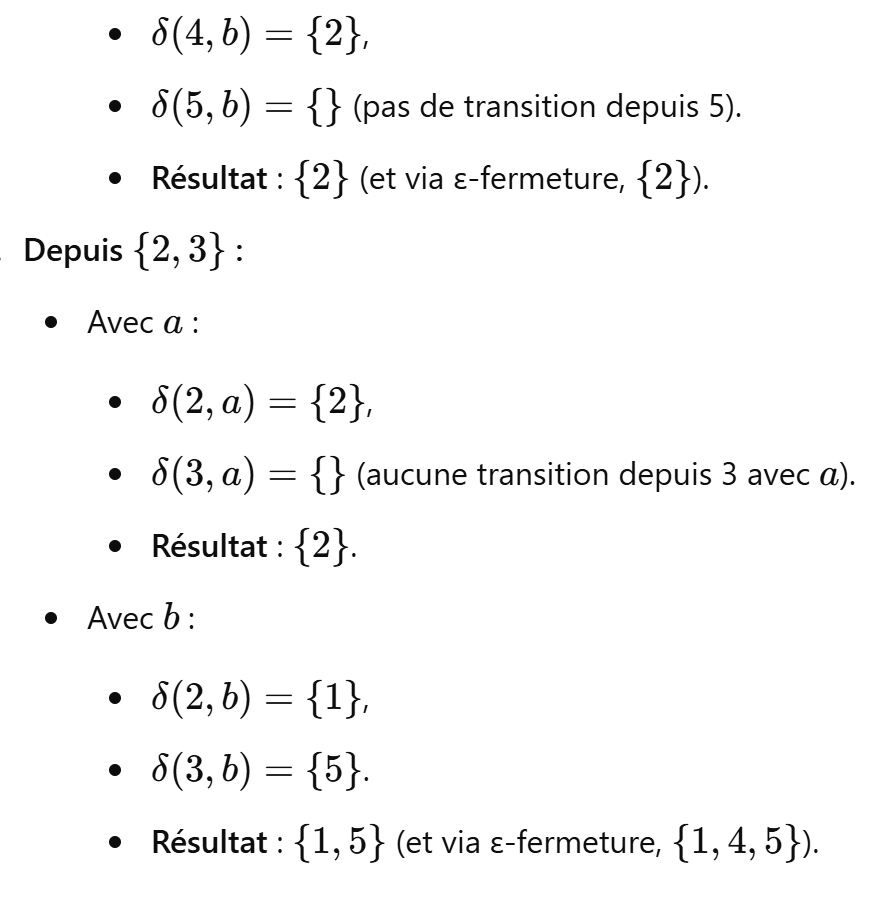


**Étape 2 : Élimination des transitions ε**

Pour convertir cet AFN en AFD, il faut d'abord gérer les transitions ε (ε) en calculant les **ε-fermetures**.







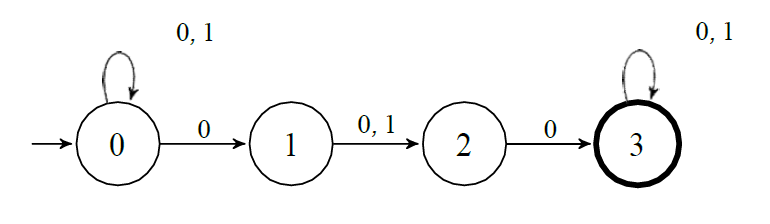
|  |  |  |
| --- | --- | --- |
| **Nouvelle état** | **Transition 0** | **Transition1** |
| **{1,4,5}** | **{2,3}** | **{2}** |
| **{2,3}** | **{2}** | **{1,4,5}** |
| **{2}** | **{2}** | **{1,4,5}** |

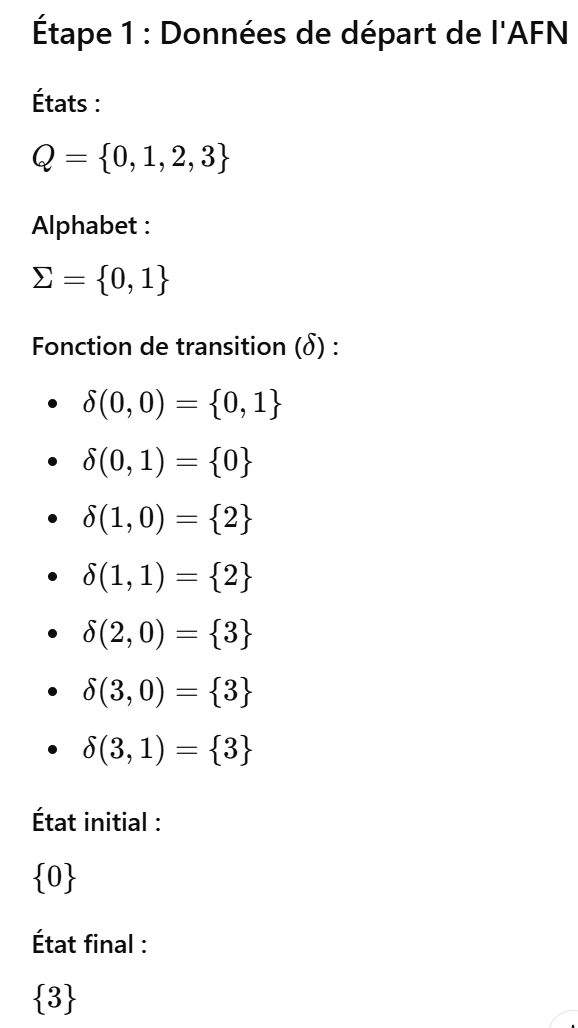
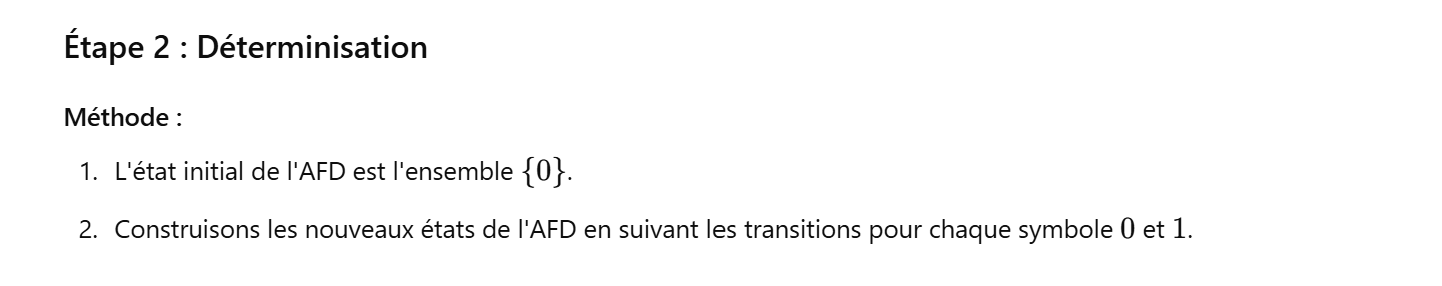
|  |  |  |  |
| --- | --- | --- | --- |
| **Nouvelle état** | **Changement variable état** | **Transition 0** | **Transition1** |
| **{1,4,5}** | **A** | **B** | **C** |
| **{2,3}** | **B** | **C** | **A** |
| **{2}** | **C** | **C** | **A** |

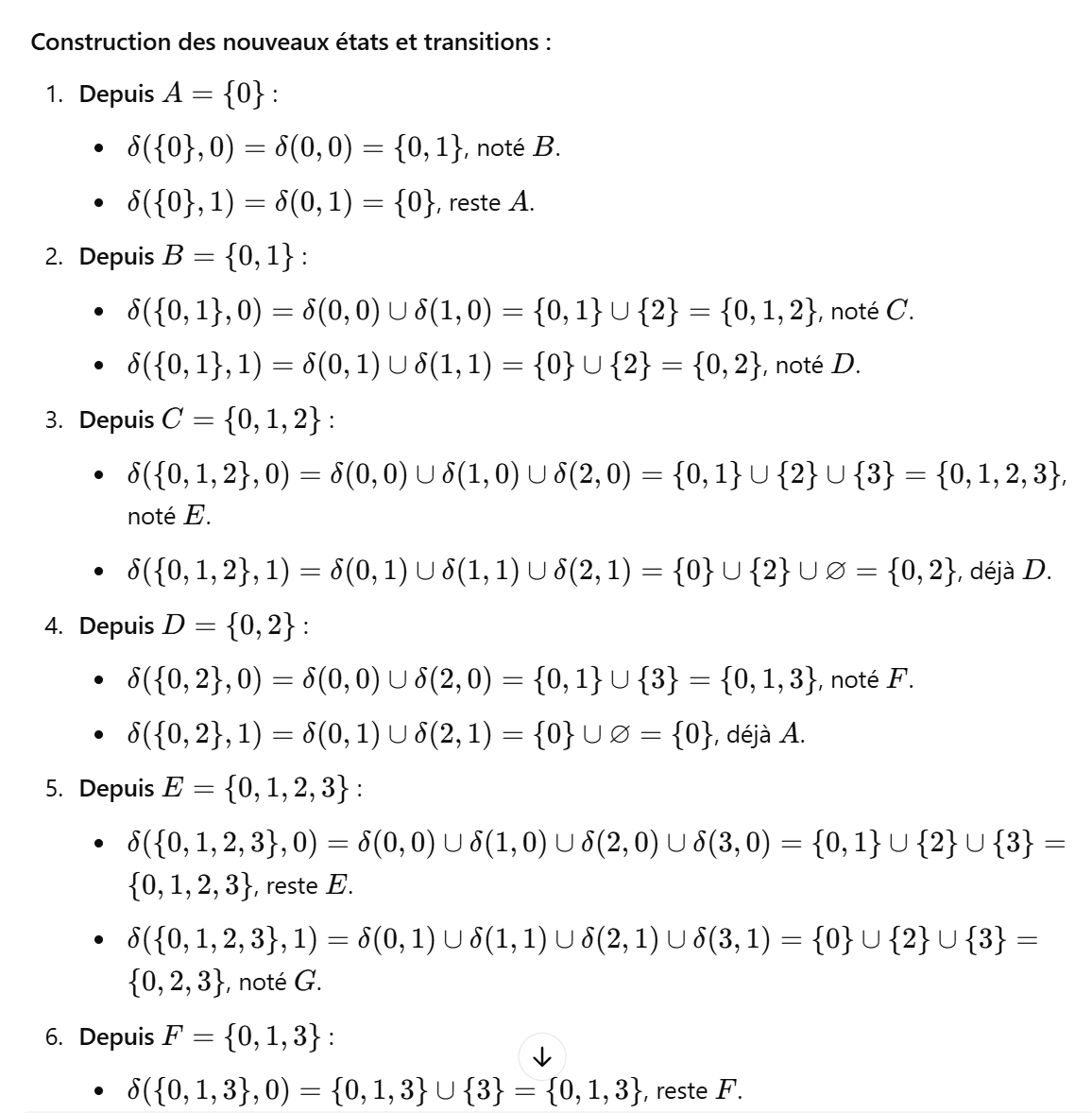
* Chaine de Longueur 2 validé par l’automate ab
* Chaine de Longueur 4 validé par l’automate abab

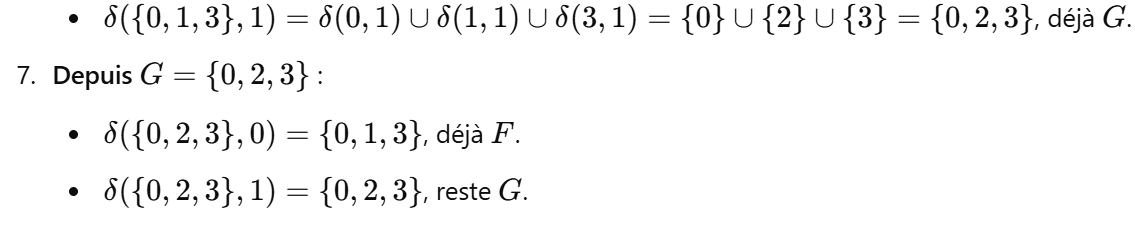
**Exercice 3**

δ(0,0)=0, δ(0,1)=0, δ(0,0)=1, δ(1,0)=2, δ(1,1)=2, δ(2,0)=3, δ(3,0)=3, δ(3,1)=3,

****

**** ****

****

****

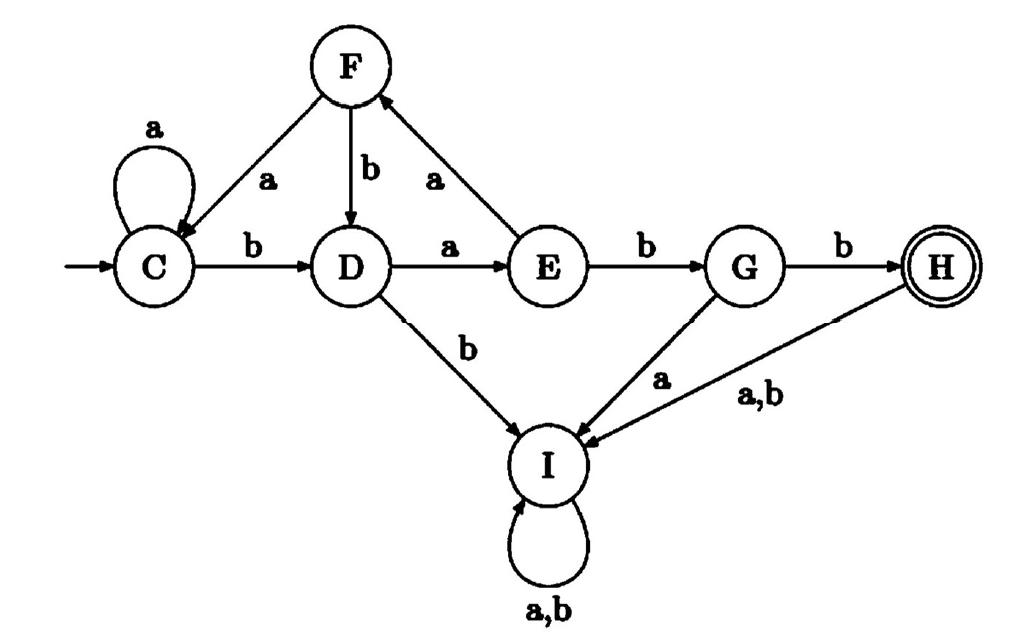
|  |  |  |
| --- | --- | --- |
| **Nouvelle état** | **Transition 0** | **Transition1** |
| **{0}** | **{0,1}** | **{0}** |
| **{0,1}** | **{0,1,2}** | **{0,2}** |
| **{0,1,2}** | **{0,1,2,3}** | **{0,2}** |
| **{0,2}** | **{0,1,3}** | **{0}** |
| **{0,1,2,3}** | **{0,1,2,3}** | **{0,2,3}** |
| **{0,1,3}** | **{0,1,3}** | **{0,2,3}** |
| **{0,2,3}** | **{0,1,3}** | **{0,2,3}** |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nouvelle état** | **Changement variable état** | **Transition 0** | **Transition1** |
| **{0}** | **A** | **B** | **A** |
| **{0,1}** | **B** | **C** | **D** |
| **{0,1,2}** | **C** | **E** | **D** |
| **{0,2}** | **D** | **F** | **A** |
| **{0,1,2,3}** | **E** | **E** | **G** |
| **{0,1,3}** | **F** | **F** | **F** |
| **{0,2,3}** | **G** | **F** | **F** |

**Avec E,F et G état terminal on les regroupe en E**

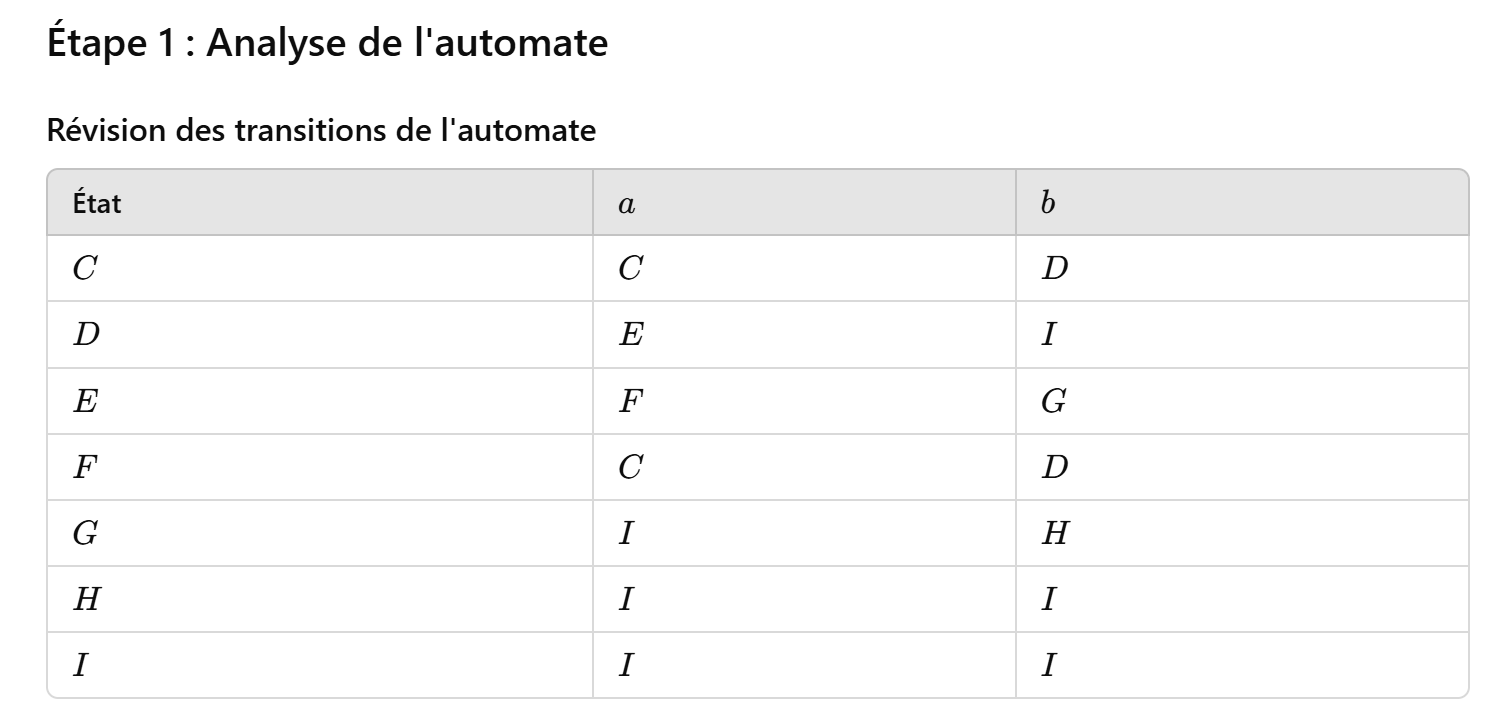
|  |  |  |  |
| --- | --- | --- | --- |
| **Nouvelle état** | **Changement variable état** | **Transition 0** | **Transition1** |
| **{0}** | **A** | **B** | **A** |
| **{0,1}** | **B** | **C** | **D** |
| **{0,1,2}** | **C** | **E** | **D** |
| **{0,2}** | **D** | **E** | **A** |
| **{0,1,2,3}** | **E** | **E** | **E** |

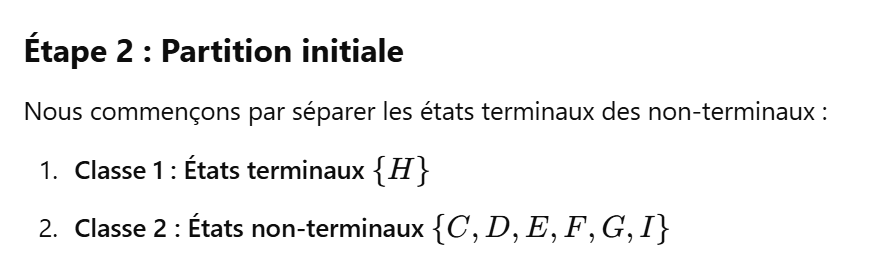
**Exercice 4**

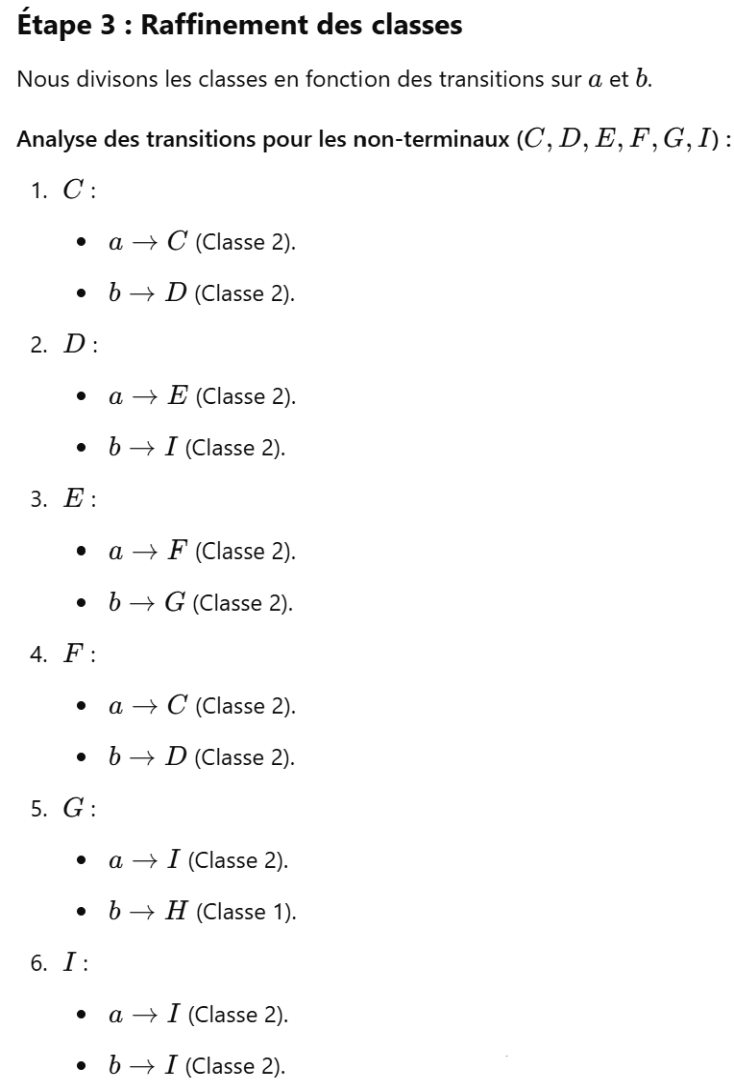


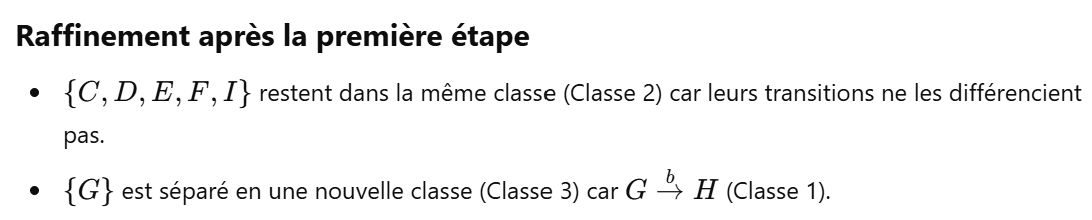
δ(C,a)=C, δ(C,b)=D, δ(D,a)=E, δ(D,b)=I, δ(E,b)=G, δ(I,a)=I, δ(I,b)=I, δ(G,b)=H, δ(G,a)=I, δ(H,a)=I, δ(H,b)=I

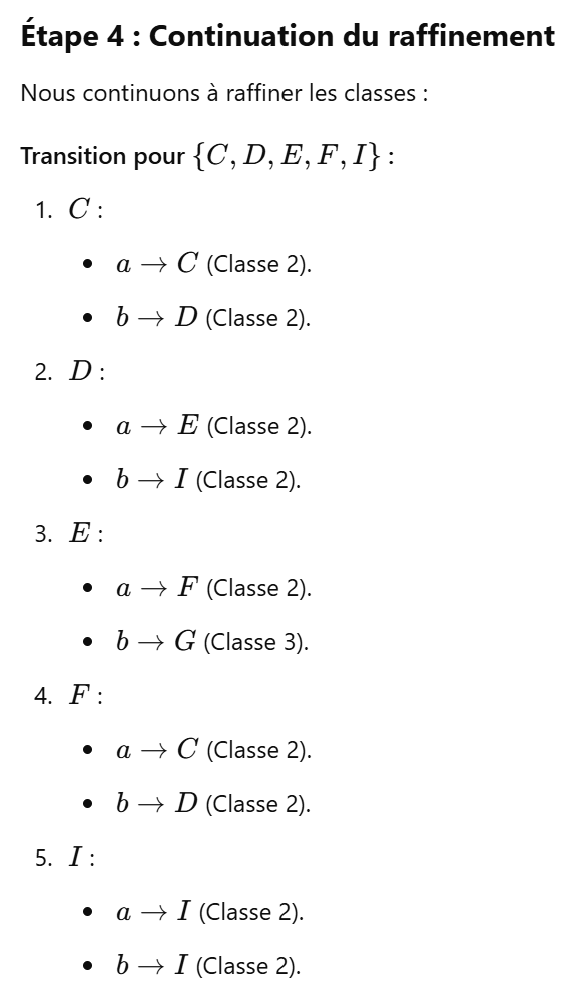
, δ(E,a)=F, δ(F,b)=D et , δ(F,a)=C

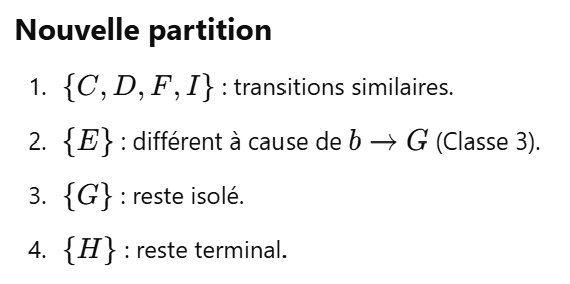


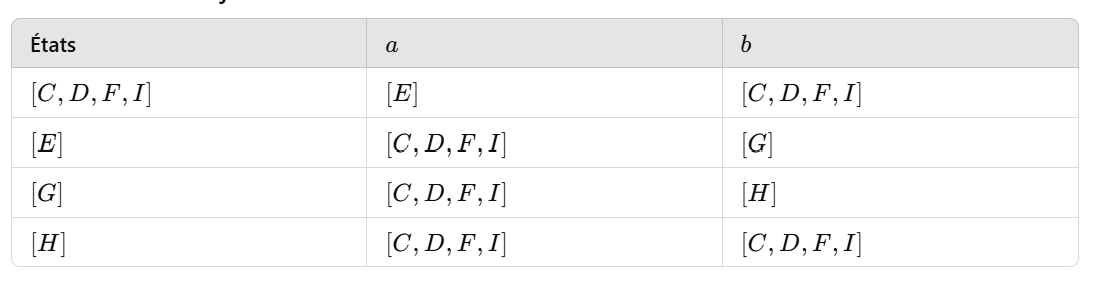






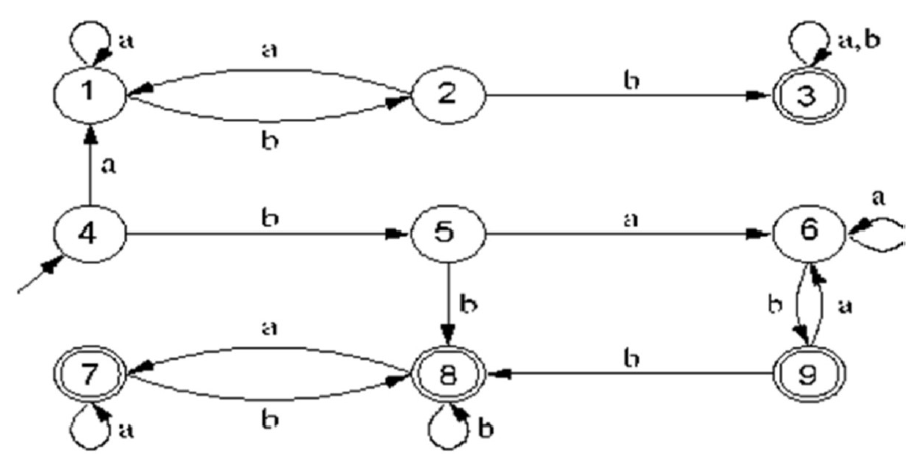






|  |  |  |  |
| --- | --- | --- | --- |
| Etat | Changement variable état | Transition a | Transition b |
| [C,D,F,I] | W | X | W |
| [E] | X | W | Y |
| [G] | Y | W | Z |
| [H] | Z | W | W |

**Exercice5**



δ(4,a)=1, δ(4,b)=5, δ(1,a)=1, δ(1,b)=2, δ(2,b)=3, δ(3,a)=3, δ(3,b)=3, δ(5,a)=6, δ(5,b)=8, δ(6,a)=6, δ(6,b)=9, δ(9,a)=6, δ(9,b)=8, δ(8,a)=7 , δ(8,b)=8, δ(7,a)=7, δ(7,b)=8

+

**Caractéristiques :**

* **État initial** : 4
* **États terminaux** : 3,7,8,9

**Minimisation par regroupement des états équivalents**

**Déterminer l’équivalence initiale des états :**

1. **États terminaux** A={3,7,8,9} sont équivalents au départ.
2. **États non terminaux** B={1,2,4,5,6} sont également regroupés ensemble.

|  |  |  |  |
| --- | --- | --- | --- |
| Etat groupé | Etat | a | b |
| {1,2,4,5,6} | 4 | 1 | 5 |
| {1,2,4,5,6} | 1 | 1 | 2 |
| {1,2,4,5,6} | 2 | - | 3 |
| {3,7,8,9} | 3 | 3 | 3 |
| {1,2,4,5,6} | 5 | 6 | 8 |
| {1,2,4,5,6} | 6 | 6 | 9 |
| {3,7,8,9} | 9 | 6 | 8 |
| {3,7,8,9} | 8 | 7 | 8 |
| {3,7,8,9} | 7 | 7 | 8 |

|  |  |  |  |
| --- | --- | --- | --- |
| Etat groupé | Changement variable état | a | b |
| {1,2,4,5,6} | A | A | B |
| {3,7,8,9} | B | B | B |