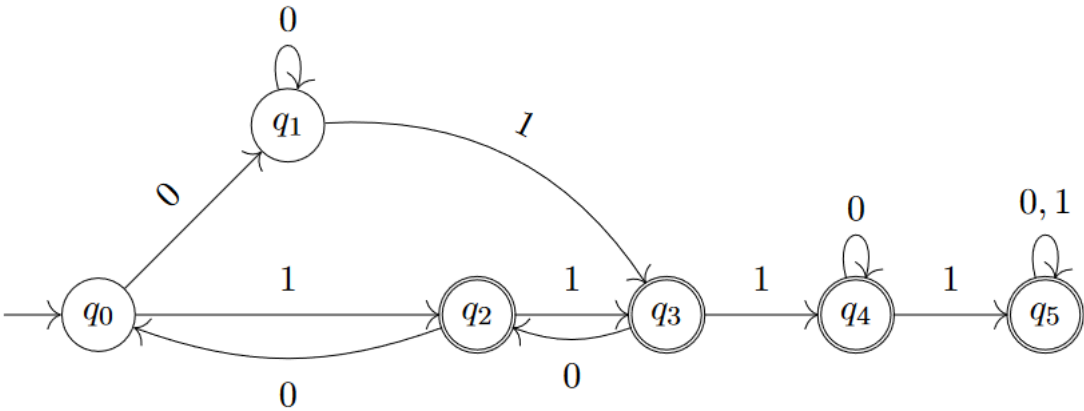


Commencé le	vendredi 3 mars 2023, 14:00
État	Terminé
Terminé le	lundi 10 avril 2023, 11:06
Temps mis	37 jours 20 heures
Points	9,00/9,00
Note	10,00 sur 10,00 (100%)

On considère l'AFD A suivant sur l'alphabet $\{0, 1\}$:



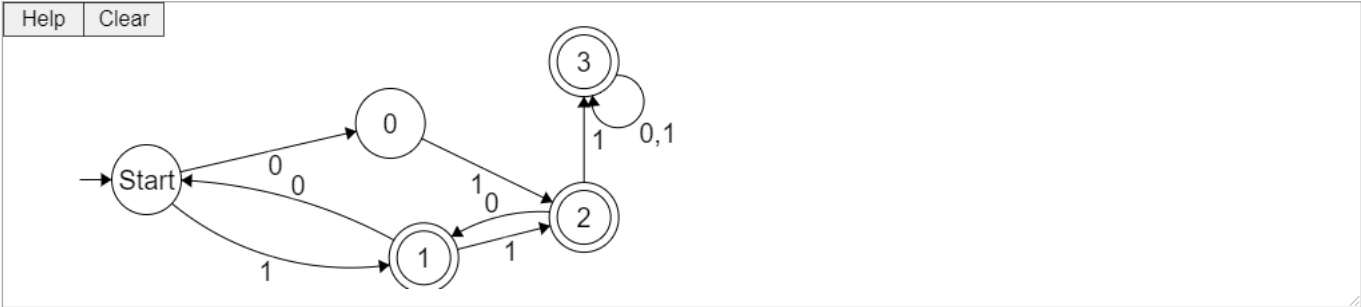
1. Minimiser cet AFD.

Par exemple:

Test	Résultat
<code>nb_states(dfa_stud)== nb_states(dfa_OK)</code>	True
<code>nb_accept_states(student_answer) == nb_accept_states(correct_answer)</code>	True
<code>accept(dfa_stud,"0")</code>	False
<code>accept(dfa_stud,"1")</code>	True
<code>accept(dfa_stud,"11")</code>	True
<code>accept(dfa_stud,"011")</code>	True
<code>accept(dfa_stud,"01")</code>	True

Réponse : (régime de pénalités : 0 %)

Réinitialiser la réponse



	Test	Résultat attendu	Résultat obtenu	
✓	<code>nb_states(dfa_stud)== nb_states(dfa_OK)</code>	True	True	✓
✓	<code>nb_accept_states(student_answer) == nb_accept_states(correct_answer)</code>	True	True	✓
✓	<code>accept(dfa_stud,"0")</code>	False	False	✓
✓	<code>accept(dfa_stud,"1")</code>	True	True	✓
✓	<code>accept(dfa_stud,"01")</code>	True	True	✓
✓	<code>accept(dfa_stud,"11")</code>	True	True	✓
✓	<code>accept(dfa_stud,"011")</code>	True	True	✓

	Test	Résultat attendu	Résultat obtenu	
✓	accept(dfa_stud,"01")	True	True	✓
✓	accept(dfa_stud,"111") == accept(dfa_stud,"1111")	True	True	✓
✓	accept(dfa_stud,"111")	True	True	✓
✓	accept(dfa_stud,"01110")	True	True	✓

Tous les tests ont été réussis ! ✓

Correct

Note pour cet envoi : 1,00/1,00.

Question 2

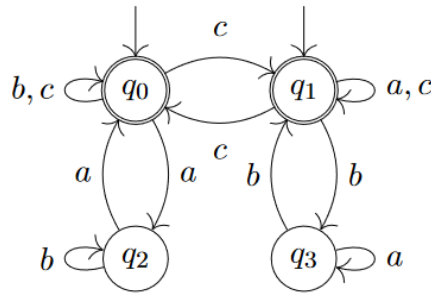
Correct

Note de 1,00 sur 1,00

Exercice 2.

Complexité des AFI

On considère l'AFI A suivant sur l'alphabet $\{a, b, c\}$:



Donner l'ensemble des transitions de A .

Réponse : (régime de pénalités : 0 %)

Réinitialiser la réponse

```

1  #remplir les ensembles d'états atteints à chaque transition
2  #séparer les états par ',' et vous pouvez mettre q ou Q ou rien, tous les q sont effacés
3  (0, 'a'): {2}
4  (0, 'b'): {0}
5  (0, 'c'): {0,1}
6  (1, 'a'): {1}
7  (1, 'b'): {3}
8  (1, 'c'): {0,1}
9  (2, 'a'): {0}
10 (2, 'b'): {2}
11 (2, 'c'): {}
12 (3, 'a'): {3}
13 (3, 'b'): {1}
14 (3, 'c'): {}

```

Tous les tests ont été réussis ! ✓

Correct

Note pour cet envoi : 1,00/1,00.

Question 3

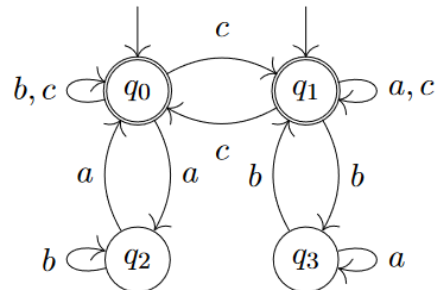
Correct

Note de 1,00 sur 1,00

Exercice 2.

Complexité des AFI

On considère l'AFI A suivant sur l'alphabet $\{a, b, c\}$:



Donner la suite des ensembles d'états atteints par lecture de $bbcab$

Réponse : (régime de pénalités : 0 %)

Réinitialiser la réponse

```

1 #remplir les ensembles d'états atteints à chaque lecture d'une lettre de "bbcab" !
2   : {0, 1}
3 b : {0, 3}
4 b : {0, 1}
5 c : {0, 1}
6 a : {2, 1}
7 b : {2, 3}
  
```

Tous les tests ont été réussis ! ✓

Correct

Note pour cet envoi : 1,00/1,00.

Question 4

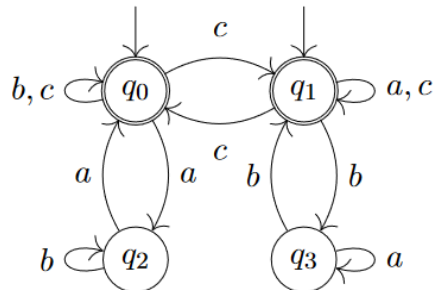
Correct

Note de 1,00 sur 1,00

Exercice 2.

Complexité des AFI

On considère l'AFI A suivant sur l'alphabet $\{a, b, c\}$:



Donner la suite des ensembles d'états atteints par lecture de $acbab$

Réponse : (régime de pénalités : 0 %)

Réinitialiser la réponse

```

1 #remplir les ensembles d'états atteints à chaque lecture d'une lettre de "acbab" !
2   : {0, 1}
3 a : {1, 2}
4 c : {0, 1}
5 b : {0, 3}
6 a : {2, 3}
7 b : {1, 2}

```

Tous les tests ont été réussis ! ✓

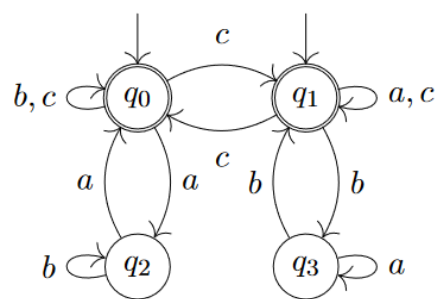
Correct

Note pour cet envoi : 1,00/1,00.

Exercice 2.

Complexité des AFI

On considère l'AFI *A* suivant sur l'alphabet $\{a, b, c\}$:



Dans votre langage de programmation préféré, c'est-à-dire Python, écrire une fonction récursive booléenne qui prend en entrée un mot *w* (une chaîne de caractère) et retourne vrai si et seulement si l'AFI *A* accepte le mot *w*

Par exemple:

Test	Résultat
rec_ifa_accept("aba")	True
rec_ifa_accept("")	True

Réponse : (régime de pénalités : 0 %)

Réinitialiser la réponse

```
1 def rec_ifa_accept(word):
2     if word=="":
3         return True
4     if word[0]=="a":
5         return rec_ifa_accept(word[1:]) or rec_ifa_acceptQ2(word[1:])
6     if word[0]=="b":
7         return rec_ifa_accept(word[1:]) and rec_ifa_acceptQ3(word[1:])
8     if word[0]=="c":
9         return rec_ifa_accept(word[1:])
10
11 def rec_ifa_acceptQ3(word):
12     if word=="":
13         return True
14     if word[0]=="a":
15         return rec_ifa_acceptQ3(word[1:])
16     if word[0]=="b":
17         return rec_ifa_accept(word[1:])
18     if word[0]=="c":
19         return False
20
21 def rec_ifa_acceptQ2(word):
22     if word=="":
23         return True
24     if word[0]=="a":
25         return rec_ifa_accept(word[1:])
26     if word[0]=="b":
27         return rec_ifa_acceptQ2(word[1:])
28     if word[0]=="c":
29         return False
30
```

	Test	Résultat attendu	Résultat obtenu	
✓	rec_ifa_accept("bbcab")	False	False	✓
✓	rec_ifa_accept("acbab")	True	True	✓
✓	rec_ifa_accept("")	True	True	✓

Tous les tests ont été réussis ! ✓

Correct

Note pour cet envoi : 1,00/1,00.

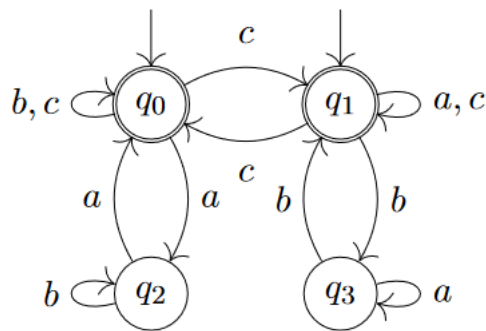
Question 6

Correct

Note de 1,00 sur 1,00

Ex2.6

On considère l'AFI A suivant sur l'alphabet $\{a, b, c\}$:



Détermiser l'AFI A en un DFA B .

Par exemple:

Test	Résultat
alphabet(fa_stud)	['a', 'b', 'c']
is_det(fa_stud)	True
accept(fa_stud,"a")	True
accept(fa_stud,"ababaa")	True

Réponse : (régime de pénalités : 0 %)

Réinitialiser la réponse

Help Clear

	Test	Résultat attendu	Résultat obtenu	
✓	alphabet(fa_stud)	['a', 'b', 'c']	['a', 'b', 'c']	✓
✓	is_det(fa_stud)	True	True	✓
✓	init_state(fa_stud)	0,1	0,1	✓
✓	accept_states_lab(fa_stud)	['0,1', '0,3', '1,2']	['0,1', '0,3', '1,2']	✓
✓	accept(fa_stud,"a")	True	True	✓
✓	accept(fa_stud,"ababaa")	True	True	✓
✓	accept(fa_stud,"abbc")	True	True	✓

	Test	Résultat attendu	Résultat obtenu	
✓	accept(fa_stud,"aacaaccba")	False	False	✓
✓	accept(fa_stud,"ababacab")	False	False	✓
✓	accept(fa_stud,"abbcabacabbcb")	True	True	✓

Tous les tests ont été réussis ! ✓

Correct

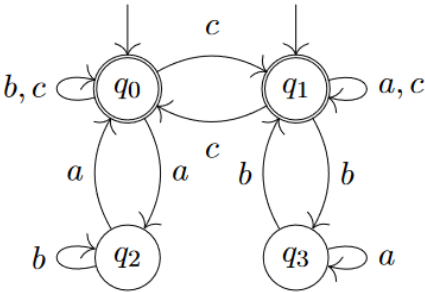
Note pour cet envoi : 1,00/1,00.

Ex2.7

Exercice 2.

Complexité des AFI

On considère l’AFI A suivant sur l’alphabet $\{a, b, c\}$:



Dans votre langage de programmation préféré, c'est-à-dire Python, écrire une fonction booléenne qui prend en entrée un mot w (une chaîne de caractère) et retourne vrai si et seulement si l'AFD B (et donc l'AFI A) accepte le mot w .

Par exemple:

Test	Résultat
accept("aba")	True
accept("")	True

Réponse : (régime de pénalités : 0 %)

Réinitialiser la réponse

```
1 def accept(word):
2     if word=="":
3         return True
4     if word[0]=="a":
5         return accept(word[1:]) or acceptQ2(word[1:])
6     if word[0]=="b":
7         return accept(word[1:]) and acceptQ3(word[1:])
8     if word[0]=="c":
9         return accept(word[1:])
10
11 def acceptQ3(word):
12     if word=="":
13         return True
14     if word[0]=="a":
15         return acceptQ3(word[1:])
16     if word[0]=="b":
17         return accept(word[1:])
18     if word[0]=="c":
19         return False
20
21 def acceptQ2(word):
22     if word=="":
23         return True
24     if word[0]=="a":
25         return accept(word[1:])
26     if word[0]=="b":
27         return acceptQ2(word[1:])
28     if word[0]=="c":
29         return False
30
```

	Test	Résultat attendu	Résultat obtenu	
✓	accept("bbcab")	False	False	✓
✓	accept("acbab")	True	True	✓

	Test	Résultat attendu	Résultat obtenu	
✓	accept("")	True	True	✓

Tous les tests ont été réussis ! ✓

Correct

Note pour cet envoi : 1,00/1,00.

Question 8

Correct

Note de 1,00 sur 1,00

Ex3.1

Écrire un AFI A en 4 états qui reconnaît le langage suivant

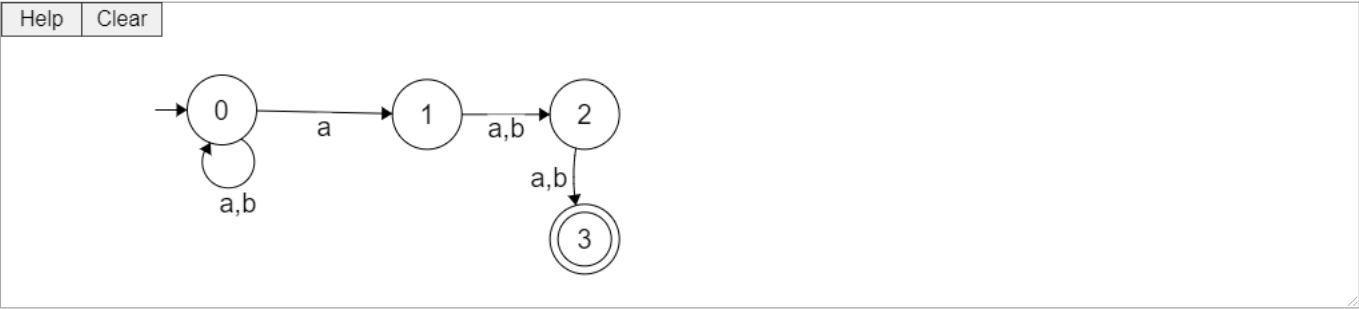
$$L = \{w = w_1w_2 \dots w_n \in \{a, b\}^* \mid |w| \geq 3 \text{ et } w_{n-2} = a\}.$$

Par exemple:

Test	Résultat
alphabet(fa_stud)	['a', 'b']
is_det(fa_stud)	False
accept(fa_stud,"a")	False
accept(fa_stud,"abb")	True

Réponse : (régime de pénalités : 0 %)

Réinitialiser la réponse



	Test	Résultat attendu	Résultat obtenu	
✓	alphabet(fa_stud)	['a', 'b']	['a', 'b']	✓
✓	is_det(fa_stud)	False	False	✓
✓	accept(fa_stud,"a")	False	False	✓
✓	accept(fa_stud,"ababaa")	False	False	✓
✓	accept(fa_stud,"abb")	True	True	✓
✓	accept(fa_stud,"aaaaba")	False	False	✓
✓	accept(fa_stud,"ababaab")	False	False	✓
✓	accept(fa_stud,"bbbabb")	True	True	✓

Tous les tests ont été réussis ! ✓

Correct

Note pour cet envoi : 1,00/1,00.

Question 9

Correct

Note de 1,00 sur 1,00

Ex3.2

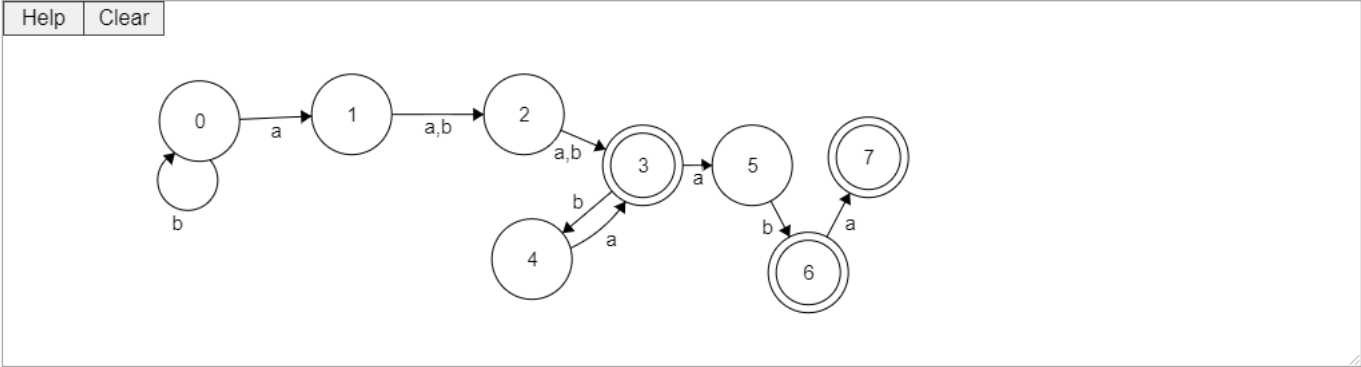
2. Déterminez votre AFI.

Par exemple:

Test	Résultat
alphabet(fa_stud)	['a', 'b']
is_det(fa_stud)	True
is_complete(fa_stud)	True
accept(fa_stud,"abb")	True

Réponse : (régime de pénalités : 0 %)

Réinitialiser la réponse



	Test	Résultat attendu	Résultat obtenu	
✓	alphabet(fa_stud)	['a', 'b']	['a', 'b']	✓
✓	is_det(fa_stud)	True	True	✓
✓	is_complete(fa_stud)	True	True	✓
✓	accept(fa_stud,"ababaa")	False	False	✓
✓	accept(fa_stud,"abb")	True	True	✓
✓	accept(fa_stud,"aaaaba")	True	True	✓
✓	accept(fa_stud,"ababaab")	True	True	✓
✓	accept(fa_stud,"bbbabb")	True	True	✓

Tous les tests ont été réussis ! ✓

Correct

Note pour cet envoi : 1,00/1,00.