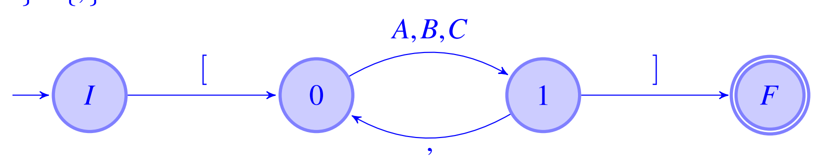
Théorie des Langages – Feuille bonus
EXERCICES DIVERS
CORRECTION

Exercice 1 - Soit U un ensemble fini d'éléments. Une liste de U est une séquence commençant par le symbole "[", se terminant par le symbole "]", et contenant des éléments de U séparés par des ",". Les mots "[A,B,C,C,A]" ou encore "[C]" sont des listes d'éléments de $U = \{A,B,C\}$. Tout au long de l'exercice, on fixera le fait qu'une liste contient toujours au moins un élément, i.e. le mot "[]" n'est pas une liste.

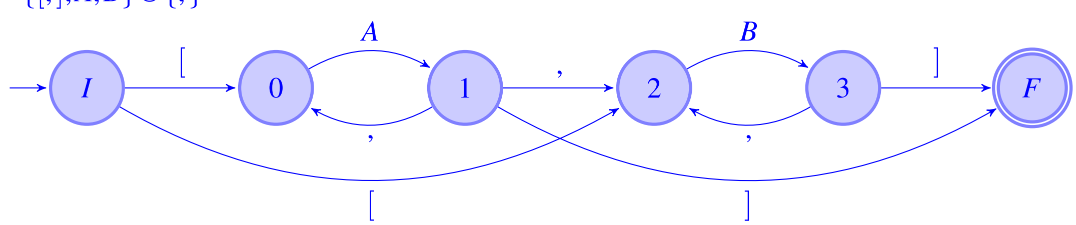
1. Est-il possible de reconnaître l'ensemble des listes d'éléments de $U = \{A,B,C\}$ avec un automate fini ? Si votre réponse est positive, donnez un automate fini reconnaissant ce langage, en indiquant précisément l'alphabet, l'état initial et le(s) état(s) finaux.

   $\Sigma = \{[,],A,B,C\} \cup \{,\}$

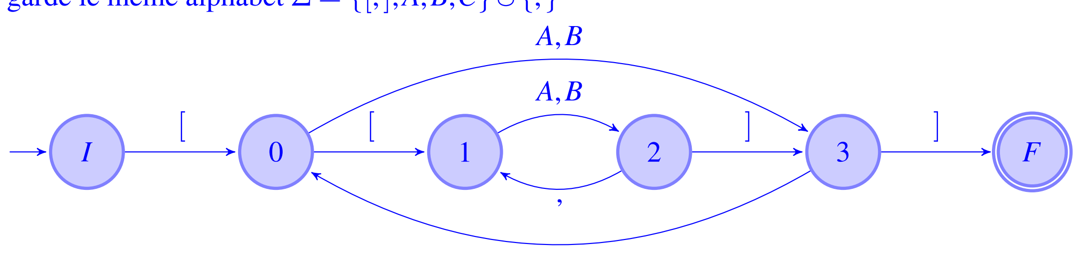
2. Dans le cas où U est totalement ordonné, on s'intéresse aux listes croissantes. Dans ces listes ordonnées, tout élément de la liste doit être supérieur ou égal à son prédécesseur. Peut-on reconnaître ces listes avec un automate fini ? Justifiez une réponse positive en représentant un tel automate dans le cas où $U = \{A,B\}$ où on dira que $A < B$. Sinon, expliquer pourquoi.

   $\Sigma = \{[,],A,B\} \cup \{,\}$

3. Jusqu'à présent, seules les listes simples était traitées. Cependant, il est possible d'étendre cette notion à des listes de listes comme par exemple $[A,[B,B],[C],[A,B,C]]$ ou encore $[[A],B,[C,C]]$. Est-il possible de reconnaître des listes de listes avec un automate fini ? Justifiez une réponse positive en représentant un tel automate dans le cas où $U = \{A,B\}$.

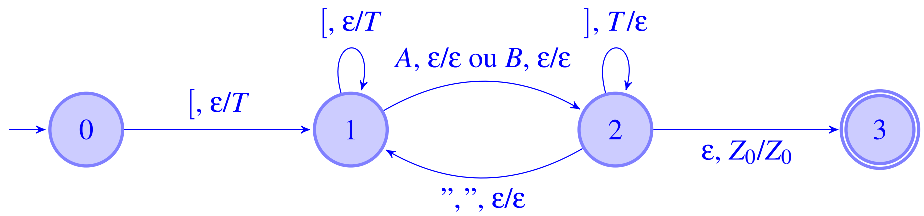
   On garde le même alphabet $\Sigma = \{[,],A,B,C\} \cup \{,\}$

4. Enfin on peut bien sûr étendre récursivement pour également avoir des listes de listes de listes, etc comme $[A,[[B,B],A],[C],[[A,B,C]]]$. Est-il possible de reconnaître ces listes avec un automate fini ? Justifiez une réponse positive en représentant un tel automate dans le cas où $U = \{A,B\}$.

Cette fois-ci, c'est impossible de représenter ces listes en utilisant un simple automate fini. On peut le faire pour un nombre fini d'imbriquation mais ici le nombre d'imbriquation est infini donc on a besoin d'un automate à pile pour pouvoir reconnaître toutes ces listes.

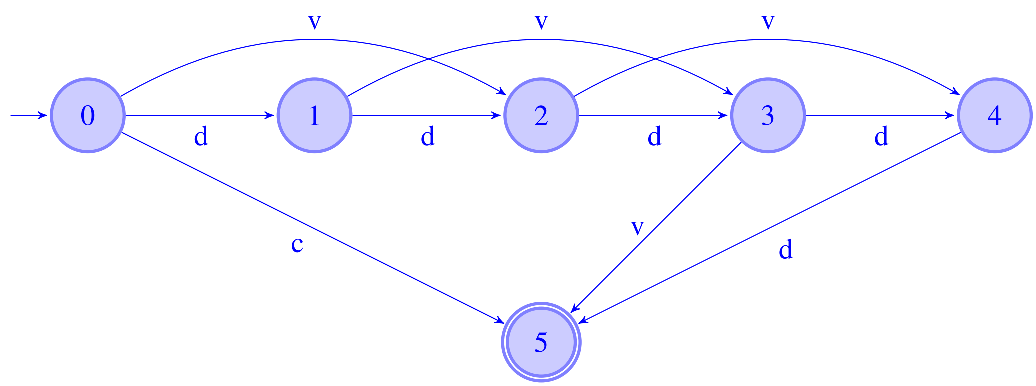
L'automate à pile suivant permet de reconnaître ces listes par acceptation par état final :



Exercice 2 - Le but de cet exercice est de modéliser un distributeur de boissons par un automate fini. L'alphabet d'entrée $\Sigma = \{d, v, c\}$ correspond aux pièces de 10, 20 et 50 centimes acceptées par le distributeur. Une boisson coûte 50 centimes.

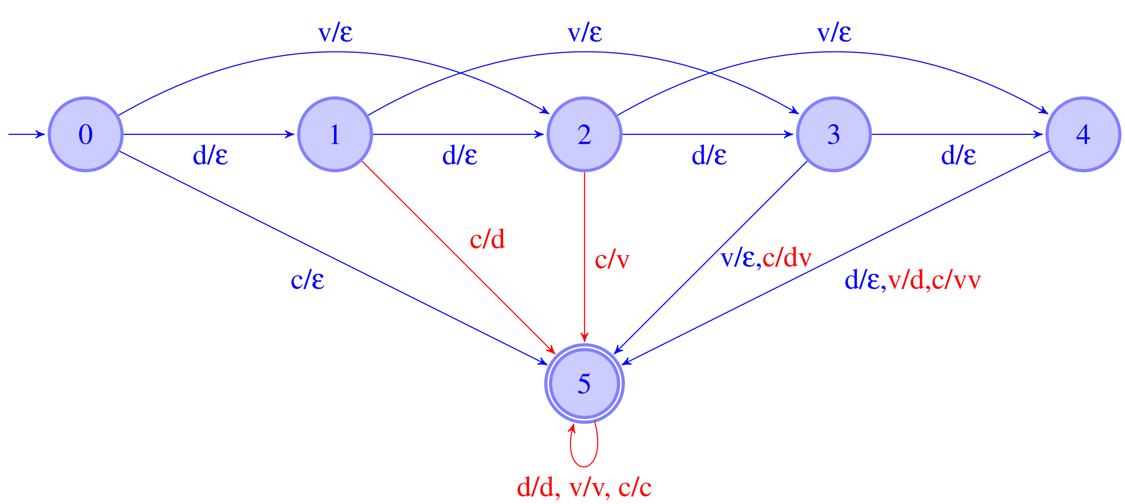
1. Construisez un automate déterministe reconnaissant toutes les séquences de pièces dont le total est 50 centimes.

L'idée est de construire un automate où chaque état représente une somme. Par exemple, l'état 0 sera l'état représentant 0 centime, l'état 1 représentera 10 centimes et ainsi de suite jusqu'à l'état 5 qui représentera 50 centimes.



2. Le but est maintenant de modéliser un distributeur qui rend la monnaie. Pour l'en rendre capable, nous allons compléter l'automate pour qu'il puisse non seulement "lire" (engranger) des pièces, mais également en "écrire" (en rendre). On parle alors de *transducteur fini*. Concrètement, chaque transition de votre nouvelle automate sera étiquetée par une paire entrée/sortie, notée *a/b* où $a \in \Sigma$ et $b \in \Sigma^*$. Emprunter une transition étiquetée *a/b* s'interprète comme : *lire un a et écrire un b*.
En partant de l'automate de la question 1), construire un transducteur fini simulant un distributeur qui rendrait la monnaie : pour toute séquence en entrée dont le total excède 50 centimes, votre transducteur doit produire en sortie une séquence dont la somme est la différence entre le montant entré et cinquante centimes.
Par exemple, la séquence de pièce (i.e. le mot) "dcv" doit être accepté et doit retourner, au choix, "dv", "ddd" ou "vd".

L'idée en fait est de rendre l'automate de la question complet en quelque sorte. On part bien sûr sur la base de l'automate de la Q1 auquel on ajoute la sortie vide ici car dans celui-ci il n'y a pas de pièce à rendre. Et on ajoute les transitions manquantes avec les pièces à rendre.



E.g. le mot "dcvv" (100 cent.) est bien accepté et la sortie est "dvv" (10+20+20 = 50 cent. rendus).

Exercice 3 [Jeu du barman aveugle] - Un barman et un client jouent au jeu suivant. Sur une table, entre les deux, se trouve un plateau où sont disposés quatre verres placés en carré. Ces verres peuvent être à l'envers ou à l'endroit. Le sens des verres (i.e. la configuration initiale) est choisi par le client et est inconnu du barman. Pour s'en assurer, celui-ci met un bandeau sur les yeux qui le rend aveugle et à interdiction de toucher les verres. Voici le déroulement de la partie :

— A chaque tour, le barman demande au client de retourner 1, 2 ou 3 verres de son choix (par exemple le barman dit au client de retourner le verre en haut à gauche et le verre en bas à gauche).

— Si tous les verres sont dans la même sens (i.e. tous à l'endroit ou tous à l'envers), le barman gagne.

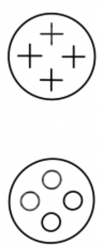
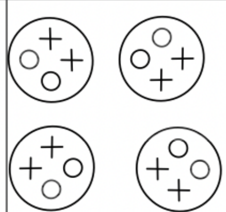
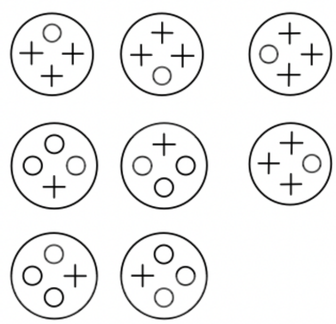
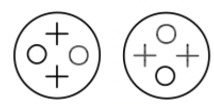
— Le barman possède 10 chances pour y parvenir, sinon le client gagne. A la fin de chaque tour, le client doit dire si le barman à gagné ou pas (il ne peut pas mentir !).

Le but est de modéliser ce jeu via un automate pour savoir s'il existe une stratégie permettant au barman de gagner quelque soit la configuration initiale.

1. Calculer le nombre total de configurations possibles et énumérer les.

   Il y a $2^4 = 16$ configurations possibles.

2. Justifier qu'il n'y a en fait que quatre dispositions réellement différentes (dont une où les verres sont tous dans le même sens).

4 verres dans le même sens	2 verres côte à côte dans le même sens	1 verre dans le même sens (ou 3 verres dans le même sens)	2 verres diagonalement opposés dans le même sens
			

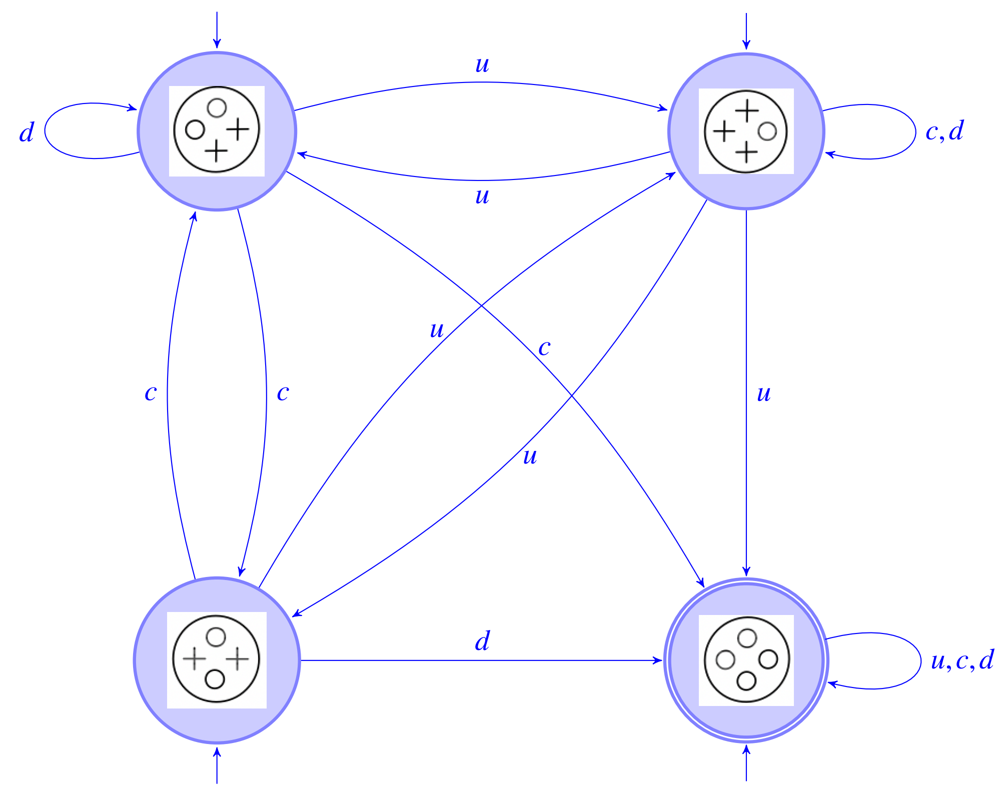
Ainsi, il y a seulement quatre états à distinguer, et seulement trois coups possibles à jouer, pour passer d'un état à un autre :

   u) retourner un verre (revient au même que d'en retourner 3),

   d) retourner deux verres en diagonales,

   c) retourner deux verres disposés sur le même côté du carré.

3. Construiser maintenant l'automate fini non-deterministe qui permet de représenté ce jeu. (Indications : les quatres dispositions précédemment trouvées sont les états de l'automate et les transitions sont les manipulations permettant de passer d'une configuration à une autre).

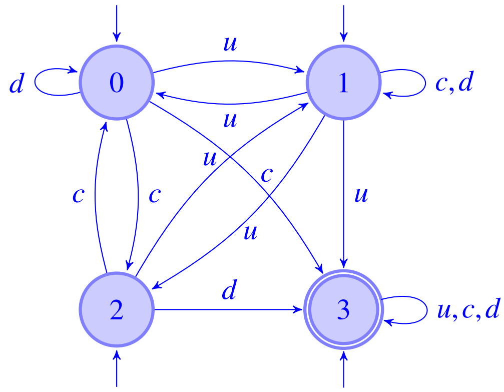


Les 4 états sont bien entendu initials car le jeu peut commencer par n'importe laquelle des configurations.
L'état final est celui où tous les verres sont dans le même sens. Une fois cette configuration atteinte, on considère que toute modification ne change rien car le barman a gagné.
On a donc trois transitions pour passer d'un état à un autre :

u) retourner un verre,
d) retourner deux verres en diagonales,
c) retourner deux verres disposés sur le même côté du carré.

4. Déterminiser l'automate.

Considérons l'automate suivant et son système d'équations associé :

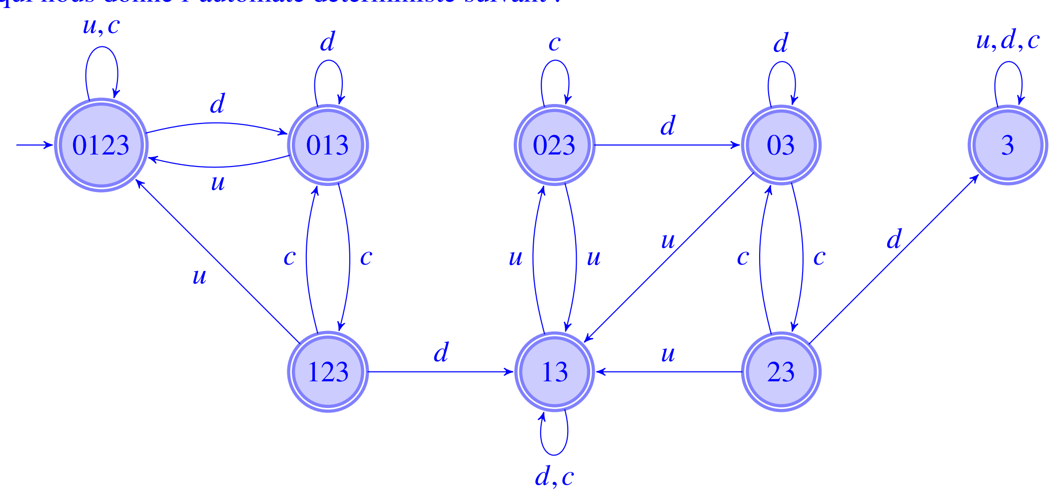


$$
\begin{array}{lcl}
L_0 & = & uL_1 + dL_0 + c(L_2 + L_3) \\
L_1 & = & u(L_0 + L_2 + L_3) + dL_1 + cL_1 \\
L_2 & = & uL_1 + dL_3 + cL_0 \\
L_3 & = & uL_3 + dL_3 + cL_3 + \varepsilon
\end{array}
$$

$\mathcal{L}(M) = L_0 + L_1 + L_2 + L_3$

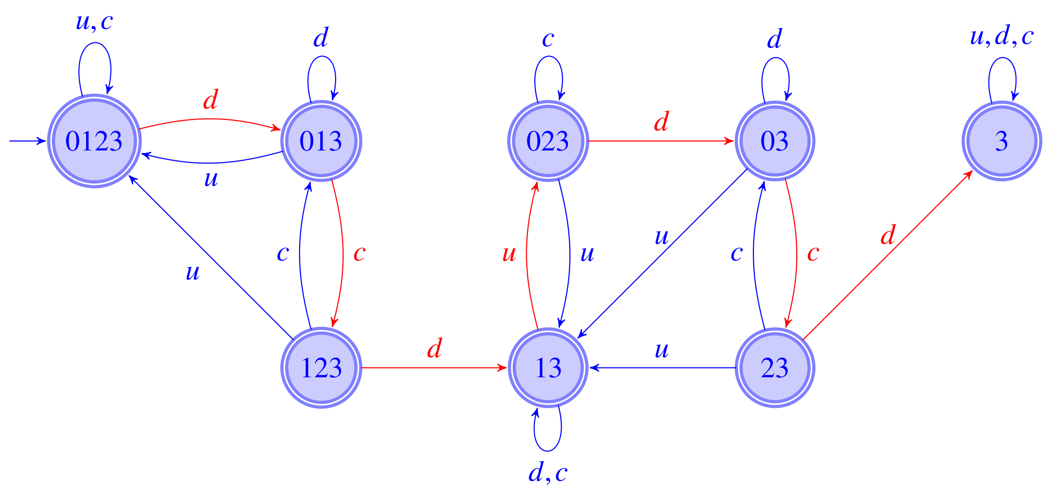
$$
\begin{array}{lcl}
L_0 + L_1 + L_2 + L_3 & = & u(L_0 + L_1 + L_2 + L_3) + d(L_0 + L_1 + L_3) + c(L_0 + L_1 + L_2 + L_3) + \varepsilon \\
L_0 + L_1 + L_3 & = & u(L_0 + L_1 + L_2 + L_3) + d(L_0 + L_1 + L_3) + c(L_1 + L_2 + L_3) + \varepsilon \\
L_1 + L_2 + L_3 & = & u(L_0 + L_1 + L_2 + L_3) + d(L_1 + L_3) + c(L_0 + L_1 + L_3) + \varepsilon \\
L_1 + L_3 & = & u(L_0 + L_2 + L_3) + d(L_1 + L_3) + c(L_1 + L_3) + \varepsilon \\
L_0 + L_2 + L_3 & = & u(L_1 + L_3) + d(L_0 + L_3) + c(L_0 + L_2 + L_3) + \varepsilon \\
L_0 + L_3 & = & u(L_1 + L_3) + d(L_0 + L_3) + c(L_2 + L_3) + \varepsilon \\
L_2 + L_3 & = & u(L_1 + L_3) + dL_3 + c(L_0 + L_3) + \varepsilon \\
L_3 & = & uL_3 + dL_3 + cL_3 + \varepsilon
\end{array}
$$

Ce qui nous donne l'automate déterministe suivant :



5. Grâce à cet automate, aider le barman à trouver une stratégie lui permettant de gagner quelque soit la configuration initiale.

Le but est ici de trouver le ou les chemins partant de l'état initial de l'AF déterministe pour aller dans l'état 3 seul (pas un multi état contenant 3).



Il y a bien entendu plusieurs possibilités. Mais on constate que la séquence de mouvement *dcdudcd* conduit nécessairement à une position gagnante pour le barman quelque soit la configuration initiale. Essayer chez vous pour constater que la stratégie est toujours gagnante (vous pouvez même laisser choisir votre adversaire en spécifiant s'il doit retourner 1 verre OU 2 verres en diagonale OU 2 verres du même côté).