Informatique Théorique

Auguste Gardette

14 Octobre 2022

1 Exercice 1.

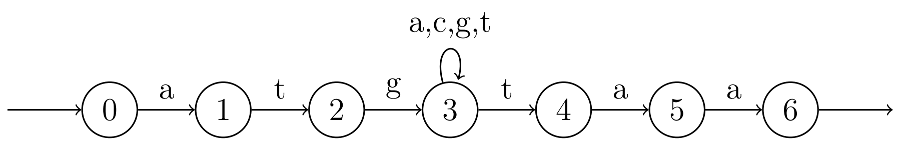
Soit L le langage des mots sur l'alphabet a, c, g, t qui commencent par atg et finissent par taa.

1. Donnez une expression rationnelle de L.

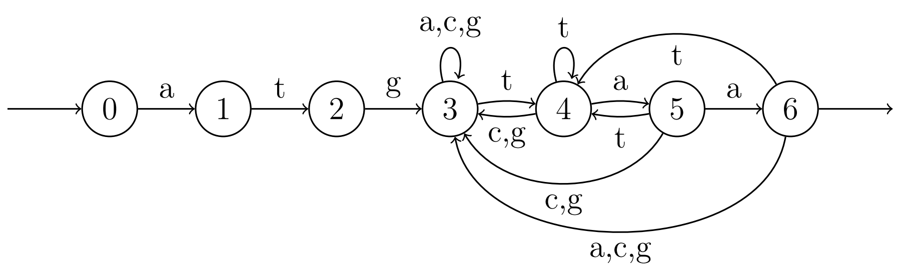
atg (a | c | g | t)* taa

2. On dit qu'un automate A =< A, Q, q_0, F, δ > est déterministe s'il satisfait les deux conditions suivantes :
— Il ne contient pas d'ϵ-transition.
— Pour tout état q $\in$ Q et toute lettre a $\in$ A, il existe au maximum une transition

(a) Construisez un automate fini non déterministe à sept états pour L.



(b) Construisez un automate fini déterministe à sept états pour L



2 Exercice 2.

Soit la grammaire G suivante :

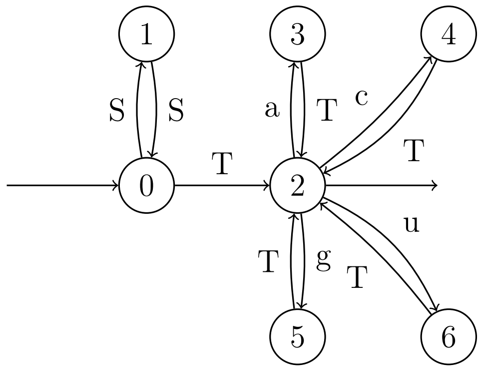
$$\boxed{\begin{array}{l} S \rightarrow aSu \mid uSa \mid cSg \mid gSc \mid SS \mid T \\ T \rightarrow aT \mid cT \mid gT \mid uT \mid \epsilon \end{array}}$$

1. Montrez que L(G) = $\{a, c, g, u\}^*$

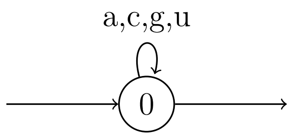
Simplifions la grammaire G comme suit :

$$\boxed{\begin{array}{l} S \rightarrow SS \mid T \\ T \rightarrow aT \mid cT \mid gT \mid uT \mid \epsilon \end{array}}$$

On peut construire un pseudo automate correspondant :



En simplifiant on obtient :

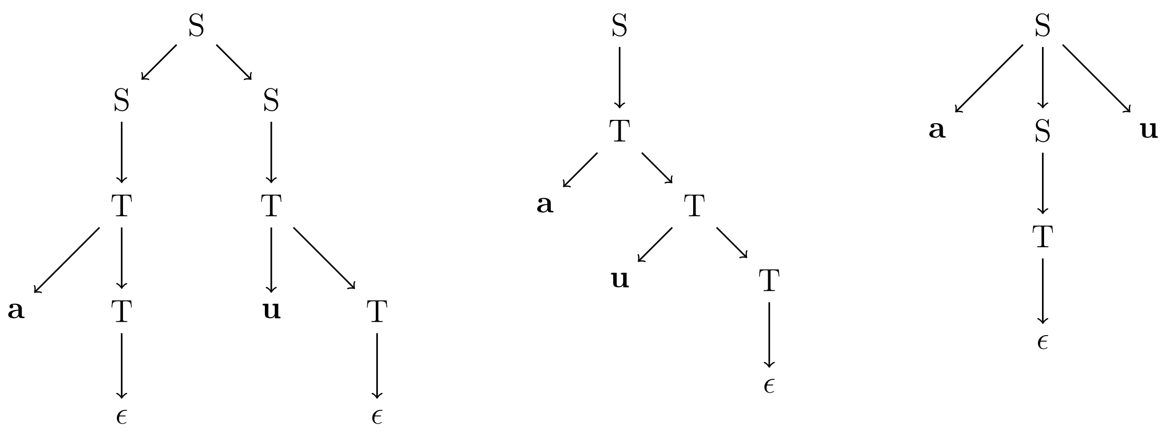


Grossièrement on pouurait même simplifier la grammaire en ne gardant que :

$T \rightarrow aT \mid cT \mid gT \mid uT \mid \epsilon$. Avec cette grammaire on peut commencer avec n'importe quelle lettre, lui incrémenter n'importe quelle lettre et finir par n'importe quelle lettre.

2. On dit qu'une grammaire G est ambigüe s'il existe un mot $w \in L(G)$ qui peut être généré par au moins deux arbres de dérivation dont la racine est l'axiome. Montrez que G est ambiguë, en considérant les arbres de dérivation qui génèrent le mot *au*.

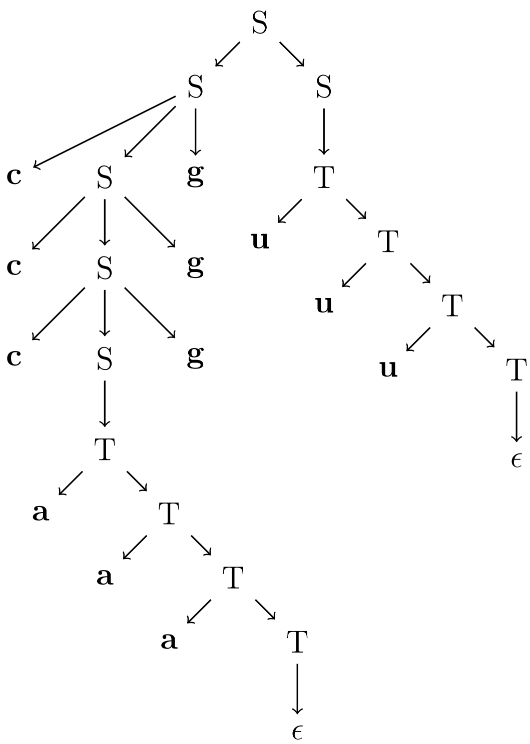
Voici quelques exemples d'arbres pour former le mot *au* :



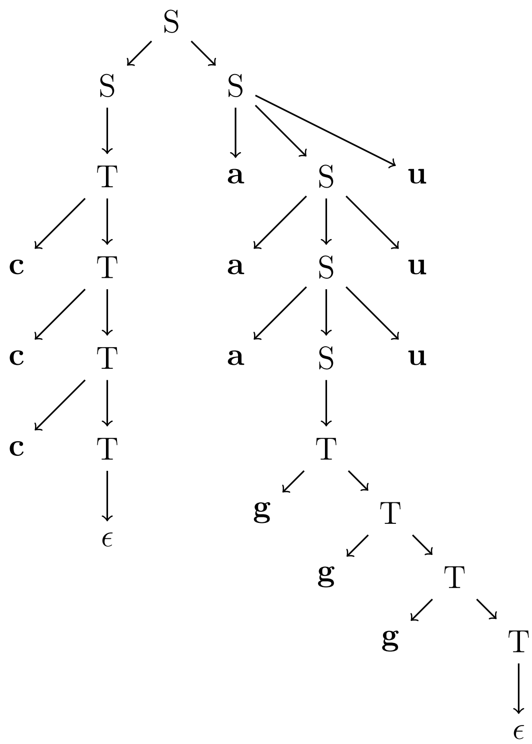
En réalité, avec la possibilité de faire S → SS, on peut générer une infinité d'arbres pour former le mot *au*, voir n'importe quel mot du langage $L(G) = \{a, c, g, u\}^*$. On peut raisonnablement dire que la grammaire G est ambigüe.

3. Soit le mot *cccaaaggguuu*. Montrez que, parmi tous les arbres de dérivation menant à ce mot depuis l'axiome, il y en a exactement deux qui utilisent trois fois une règle du type S → xSy et qui utilisent une fois la règle S → SS. Dessinez ces deux arbres.

Arbre 1 :



Arbre 2 :

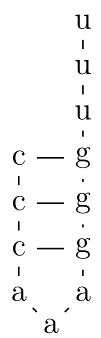


Si on veut respecter la règle S → SS, l'uinque solution est de la placer au début puisqu'aucune règle de S n'est de la forme cSu (nécessaire pour le mot *cccaaaggguuu*) donc rien ne peut substituer SS pour le premier coup. De mme, comme SS est obligatoirement en premier, cela empche son utilisation par la suite

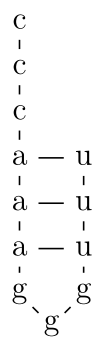
Puis, si l'on souhait respecter la règle 'trois fois S →' xSy, nous somme obligé d'utiliser les règles cSg (pour le premier S) ou aSu (pour le second S) trois fois d'affilés pour réprondre à cette contrainte. Il n'y a donc bien que 2 solutions d'arbre possible.

4. Chacun de ces arbres de dérivation correspond à une structure secondaire qui peut *ê*tre adoptée par la séequence d'ARN*cccaaaggguuu*. Dessinez les deux structures secondaires correspondant respectivement à chacun des deux arbres.

Arbre 1 - Structure secondaire :



Arbre 2 - Structure secondaire :



Représentation de la structure ARN par séquence et système de parenthèses :

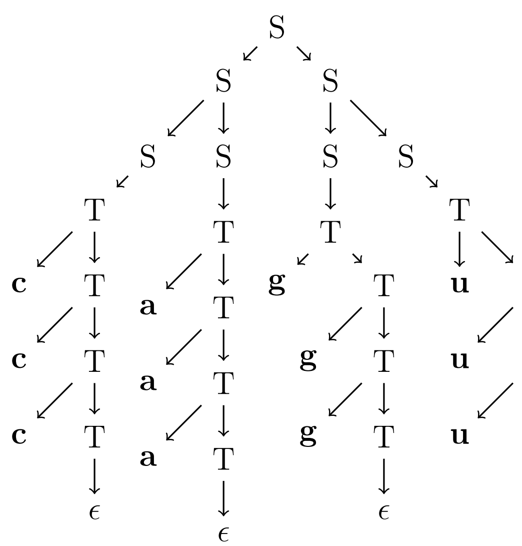
Arbre 1 - Structure secondaire bis :

(((. . .))) . . .
c c c a a a g g g u u u

Arbre 2 - Structure secondaire bis :

. . . (((. . .)))
c c c a a a g g g u u u

5. Donnez un autre arbre de dérivation pour la même séequence et dessinez la structure secondaire correspondante.



Structure secondaire correspondante :

. . . .
c-c-c-a-a-a-g-g-g-u-u-u

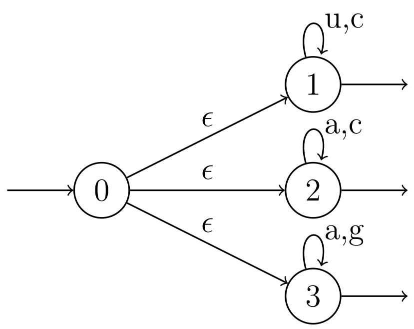
6. On considère pour cette question que les appariements possibles dans une structure secondaire d'ARN sont les appariements canoniques a-u, u-a, c-g et g-c, ainsi que les appariements dits "wobble" g-u et u-g.

(a) Soit L_0 le plus grand langage sur l'alphabet {a, c, g, u} qui possède la propriété suivante : quel que soit $w \in L_0$, il n'y a aucun appariement possible dans w. Donnez une expression rationnelle et un automate fini pour L_0.

Expression rationnelle de L_0 :

(a | c)* | (a | g)* | (u | c)*

Automate fini de L_0 :



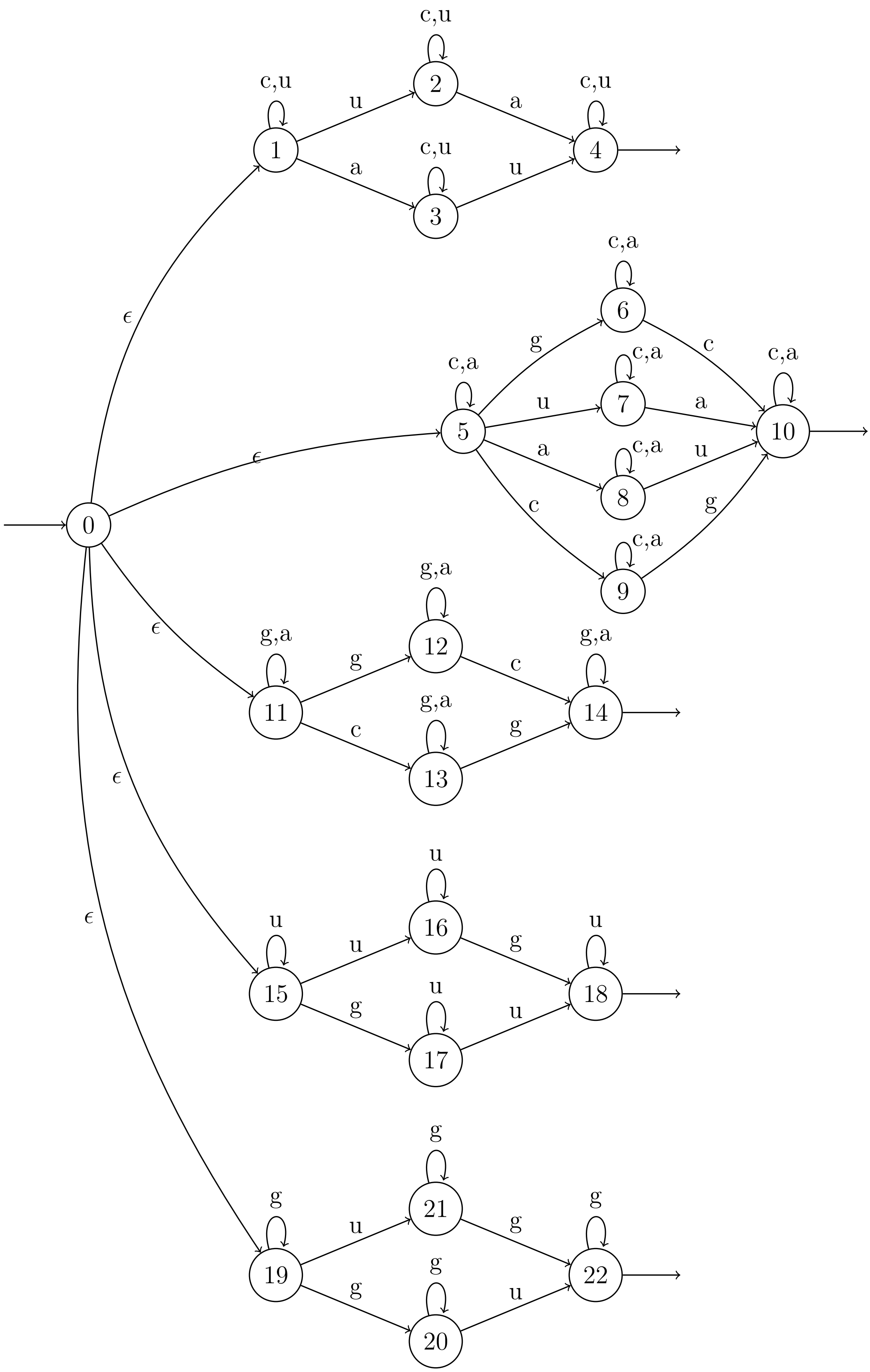
(b) Soit L_1 le plus grand langage sur l'alphabet {a, c, g, u} qui possède la propriété suivante : quel que soit $w \in L_1$, il y a exactement un appariement possible dans w . Donnez une expression rationnelle et un automate fini pour L_1 .

On interprètre la question comme suit :Il peut y avoir plusieurs appariements dans la séquence, mais pas simultanément. Ainsi AGU et AAAAU sont dans L1. Mais AAAAUU ne l'est pas.

Expression rationnelle de L_1 :

(c | a)* a (c | a)* u (c | a)* | (c | u)* a (c | u)* u (c | u)* |
(c | a)* u (c | a)* a (c | a)* | (c | u)* u (c | u)* a (c | u)* |
(c | a)* g (c | a)* c (c | a)* | (g | a)* g (g | a)* c (g | a)* |
(c | a)* c (c | a)* g (c | a)* | (g | a)* c (g | a)* g (g | a)* |
(u)*g(u)*u(u)* | (u)*u(u)*g(u)* | (g)*g(g)*u(g)* | (g)*u(g)*g(g)*

Automate fini de L_1 :



0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
ϵ
c,u
u
a
c,a
g
c
g,a