

HA601I - Exercices de révisions

Benoît Huftier

2022

Construction d'un AFD à partir d'un AFN

Enoncé

- 1 Décrire l'expression régulière permettant de définir un nombre entier en base 10, correspondant au langage \mathcal{L}_{int} .
- 2 Faire de même pour le langage \mathcal{L}_{flot} , correspondant aux nombres flottants (sans exposant).
- 3 Peut-on écrire une expression régulière correspondant au langage if-then-else ? Et un AFD ?
- 4 Construire un AFD pour les langages \mathcal{L}_{int} et \mathcal{L}_{flot} . Combiner et minimiser les deux AFD.
- 5 Construire un AFD décrivant le langage à balise $\{\langle a \rangle, \langle /a \rangle\}$ avec l'algorithme de Thomson.
- 6 Est-ce intéressant de minimiser cet AFD dans le cadre d'une analyse lexicale ? Justifier la réponse.

Enoncé

- 1 Décrire l'expression régulière permettant de définir un nombre entier en base 10, correspondant au langage \mathcal{L}_{int} .

$$e = [0-9]^+$$

Enoncé

- 2 Faire de même pour le langage \mathcal{L}_{flot} , correspondant aux nombres flottants (sans exposant).

$$e = (. [0-9] | ([0-9] + .)) [0-9]^* {}^1$$

Il y a trois possibilités différentes pour définir un flottant :

- $xx.xx$: il y a une virgule entourée de deux nombres
- $xx.$: il y a une virgule qui n'est là que pour indiquer qu'il s'agit d'un flottant.
- $.xx$: il y a une virgule sans rien devant : synonyme de nombre inférieur à 0.

1. Le $.$ est considéré comme un caractère du vocabulaire à part entière et non pas comme n'importe quel caractère pour éviter que l'expression régulière ne devienne illisible.

Enoncé

- 3 Peut-on écrire une expression régulière correspondant au langage if-then-else ? Et un AFD ?

La grammaire des expressions régulières étant considérée comme une grammaire algébrique non régulière et selon le théorème d'Ogden il est possible d'écrire une telle expression régulière.²

En revanche, il est impossible de définir un AFD traduisant ce langage car il est impossible d'avoir un AFD (fini) traduisant un langage de type $i^n e^n$.

2. Ces diapos n'étant pas fait par un prof, je dénigre toute responsabilité quant à la réponse à cette question. C'est ce que j'ai compris du théorème (page 16 du cours), mais étant donné que c'est à partir des mots de taille $> k$ et $k \neq 0$, il est possible que je me trompe.

Enoncé

- 4 Construire un AFD pour les langages \mathcal{L}_{int} et \mathcal{L}_{flot} . Combiner les deux AFD.

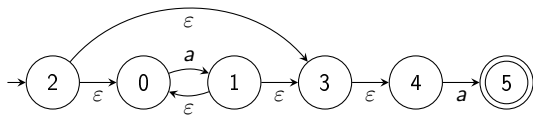
Ces AFD peuvent être écrit à la main en utilisant des neurones ou défini avec l'algorithme de Thompson (ce qui est fait ici). Pour ce faire, il faut traduire les expressions régulières données plus haut pour pouvoir utiliser l'algorithme du cours.

Note : pour plus de compréhension, on peut définir l'expression "[0-9]" comme un caractère du vocabulaire utilisé par l'AFD (on va le noter "a").

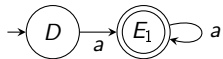
- $[0-9]^+ = aa^*$
- $(.[0-9]|([0-9]^+).)[0-9]^* = (.a|aa^*.)a^*$

$$e = aa^*$$

AFN

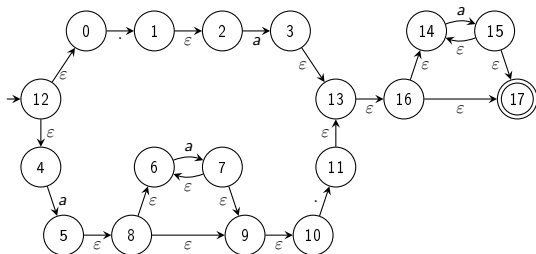


AFD

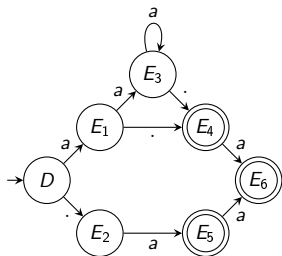


$$e = (.a|aa^*.)a^*$$

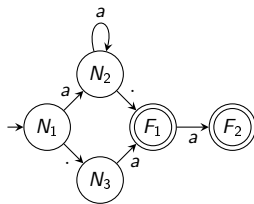
AFN



AFD

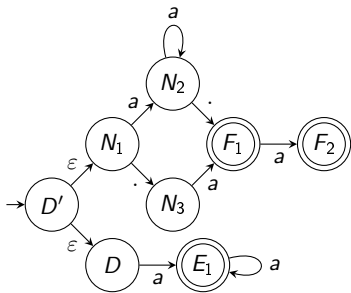


AFDM

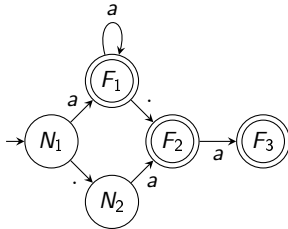


Combinaison des deux AFD

AFN



AFD

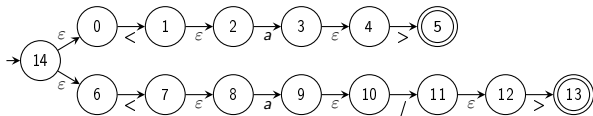


L'AFD est déjà minimisé.

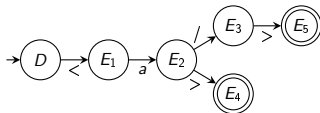
Enoncé

- 5 Construire un AFD décrivant le langage à balise $\{<a>, \}$ avec l'algorithme de Thomson.

AFN



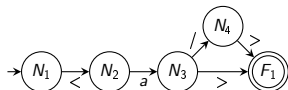
AFD



Enoncé

- 6 Est-ce intéressant de minimiser cet AFD dans le cadre d'une analyse lexicale ? Justifier la réponse.

Voici ce que donnerait la minimisation de l'AFD précédent :



Dans le cadre d'une analyse lexicale, minimiser cet AFD serait une erreur. En effet, la minimisation concatène des états finaux entre eux. Or, l'AFD d'une analyse lexicale utilise chaque état final pour renvoyer un jeton en particulier ! Si l'automate lit un flot et s'arrête sur l'état $F1$, comment savoir si l'on vient de lire une balise ouvrante ou une balise fermante ?

A retenir : minimiser un AFD est non seulement une perte de temps pour une analyse lexicale (car très coûteux pour un gain de performance faible), mais elle risque aussi d'entraîner des erreurs d'analyse.