TD : Théorème de Kleene

Exercice 1

En langage C une constante littérale entière de type `int` peut être écrite en base 2 en la préfixant par `0b`, voici quelques exemples :

```
int n = 0b1001; // a même sens que int n = 9
int m = 0B1001; // est aussi possible
int k = -0b111; // ainsi que les negatifs
```

autrement dit une valeur commence par un signe optionnel, un préfixe `0b` ou `0B` puis une suite finie non vide de caractères `0` ou `1`.

1. Proposer une expression régulière dénotant le langage des constates entières littérales binaires en C sur l'alphabet $\Sigma = \{+, -, 0, 1, b, B\}$
2. En déduire un automate fini déterministe reconnaissant ce langage.

Exercice 2

Soit $\Sigma = \{a, b\}$.

1. Donner une expression régulière et un automate fini déterministe simple pour les mots contenant un nombre pair de `a`.
2. Donner une expression régulière et un automate fini déterministe simple pour les mots contenant un nombre impair de `b`.
3. En déduire une expression régulière dénotant les mots contenant un nombre pair de `a` et un nombre impair de `b`.

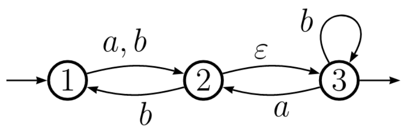
Exercice 3

Soit $\Sigma = \{a, b\}$.

1. Construire un automate à la Thomson pour le langage L dénoté par l'expression régulière $a * (ab) *$.
2. En déduire un automate reconnaissant $\overline{L}$.

Exercice 4

À l'aide de l'algorithme par élimination d'états, déterminer une expression régulière dénotant le langage reconnu par l'automate :



★Exercice 5

Quotients et dérivée de Brzozowski (Centrale, 2022)

Soit L un langage sur Σ et u un mot. On appelle quotient à gauche par u de L le langage :

$$u^{-1}L = \{v \in \Sigma^*, uv \in L\}$$

1. Soit L_1 l'ensemble des mots sur $\Sigma = \{a, b\}$ contenant autant de a que de b, déterminer $a^{-1}L_1$, $b^{-1}L_1$ et $(ab)^{-1}L_1$.
2. Soit L_2 le langage dénoté par $e_2 = a * b*$, déterminer $a^{-1}L_2$, $b^{-1}L_2$ et $(ab)^{-1}L_2$.
3. Montrer que pour toute paire de mots (u, v) et tout langage L.

$$(uv)^{-1}L = v^{-1}u^{-1}L$$

Soit e une expressions régulière et $a \in \Sigma$ une lettre. On appelle dérivée de Brozozwski de e, notée $\partial_a e$, obtenue en *dérivant* l'expression par rapport à a, selon les règles suivantes :

- $\partial_a a = \varepsilon$
- $\partial_a x = \varnothing$ pour toute lettre $x \neq a$
- $\partial_a \varepsilon = \partial_a \varnothing = \varnothing$

- $\partial_a(e_1 \mid e_2) = \partial_a e_1 \mid \partial_a e_2$
- $\partial_a(e_1.e_2) = (\partial_a e_1).e_2 \mid c(e_1).(\partial_a e_2)$
- $\partial_a(e*) = (\partial_a e).e*$

avec $c(f) = \varepsilon$ si l'expression régulière f dénote un langage contenant ε et $c(f) = \varnothing$ sinon.

4. Calculer $\partial_a e_2$ et $\partial_b e_1$, en essayant de simplifier les résultats.
5. Calculer $\partial_a[(a*b)*a*]$, en essayant de simplifier les résultats.
6. Montrer comment calculer $c(f)$ par induction.
7. Soit L un langage dénoté par e et a une lettre, démontrer que $\partial_a e$ dénote le langage $a^{-1}L$.

En 1964, Brzozowski démontre que l'ensemble des expressions régulières qu'on peut obtenir par dérivées successives d'une expression régulière est fini (modulo certaines simplifications). Cette remarque est à la base de l'algorithme d'Antimirov permettant de construire un automate équivalent à une expression régulière.