chapitre 5

Langages rationnels

I Expressions régulières

I.1 Définitions

Définition 1

Les expressions régulières sur un alphabet Σ sont
- ▷ *soit $\varnothing$;*
- ▷ *soit ε ;*
- ▷ *soit un caractère de Σ ;*
- ▷ *soit la concaténation de deux expressions régulières (que l'on note $e.f$ ou ef) ;*
- ▷ *soit le choix de deux expressions régulières (que l'on note $e|f$).*
- ▷ *soit l'étoile d'une expression régulière (que l'on note $e\star$).*

Ainsi, les expressions régulières sont définies de manière récursive à partir des caractères et des variables et de trois constructeurs (un unaire et deux binaires).

Remarque 2

Il y a a priori des parenthèses dans l'écriture d'une expression régulière. Toutefois, en définissant les règles de priorité étoile puis concaténation puis choix, on peut les éviter. Par exemple, $ab\star|b$ représente $(a.(b\star))|b$.
La taille d'une expression régulière est le nombre de caractères de son écriture sans parenthèse.

I.2 Interprétation

Définition 3

L'opération d'interprétation L associe à une expression rationnelle sur l'alphabet Σ un langage sur le même alphabet de manière récursive avec les règles suivantes
- ▷ $L(\varnothing) = \varnothing$,
- ▷ $L(\varepsilon) = \{\varepsilon\}$,
- ▷ *pour tout* $x \in \Sigma$, $L(x) = \{x\}$,
- ▷ *pour toutes expressions régulières* e *et* f, $L(e|f) = L(e) \cup L(f)$,
- ▷ *pour toutes expressions régulières* e *et* f, $L(e.f) = L(e)L(f)$,
- ▷ *pour toute expression régulière* e, $L(e\star) = L(e)^*$.

Exemple 4

L'expression régulière $ab\star|b$ s'interprète en $ab^* \cup b$.

Remarque 5

L'interprétation ne définit pas une application injective. Par exemple, les expressions rationnelles $a|bb\star a$ et $b\star a$ s'interprètent toutes les deux en le langage b^*a car

$$a \cup bb^*a = a \cup b^+a = (\{\varepsilon\} \cup b^+)a = b^*a.$$

Définition 6

Deux expressions régulières e et f sont (sémantiquement) équivalentes si elles s'interprètent en le même langage, c'est-à-dire $L(e) = L(f)$.

Définition 7

Un langage $L \subset \Sigma^*$ est rationnel s'il est l'interprétation d'une expression régulière sur l'alphabet Σ.

Proposition 8

L'ensemble des langages rationnels sur Σ est la plus petite partie de Σ^* contenant $\varnothing$, $\{\varepsilon\}$, $\{x\}$ pour tout $x \in \Sigma$ et stable par réunion, concaténation et passage à l'étoile.

I.3 Implémentation

```
type expreg=
    | vide
    | const of string
    | concatenation of expreg * expreg
    | choix of expreg * expreg
    | etoile of expreg
;;
```

L'expression régulière $e = a|bb\star a$ est alors codée par

```
let e = choix (const "a", concatenation (concatenation (const "b",
        etoile (const "b")), const "a"));;
```

II Théorème de Kleene (sens direct)

Théorème 9 (*Kleene*)

Un langage rationnel est reconnaissable.

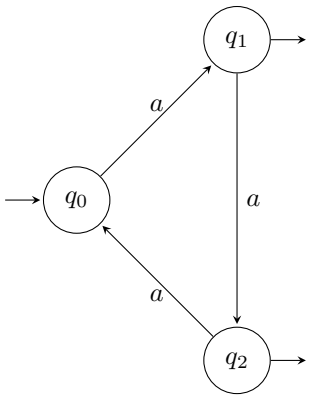
II.1 Algorithme de Thompson

Définition 10

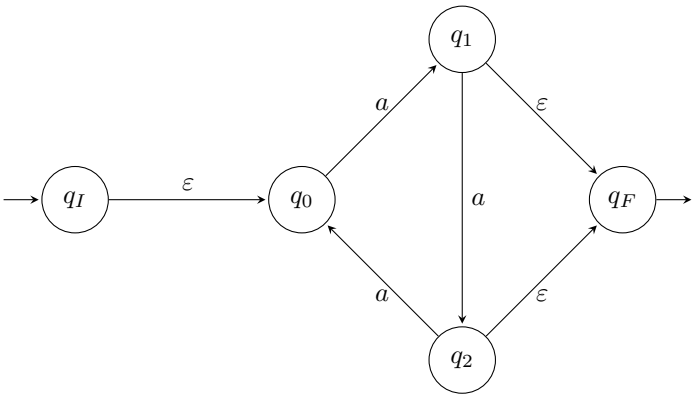
Un automate fini est standard s'il satisfait les conditions suivantes
- ▷ *il n'y a qu'un état initial et aucune transition n'y aboutit ;*
- ▷ *il n'y a qu'un état acceptant et aucune transition n'en part.*

Exemple 11

L'automate suivant n'est pas standard

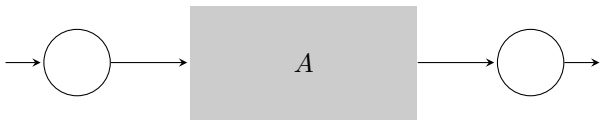


En revanche, en ajoutant un état initial et un état acceptant et des transitions instantanées depuis ou vers ces états, on obtient



Notation 12

On notera pour simplifier les preuves les automates standards sous la forme d'une « boite noire » avec un état initial et un état acceptant :



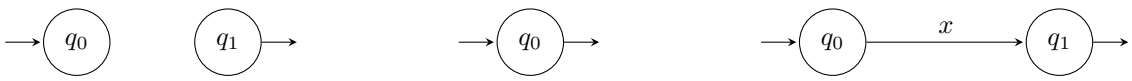
Proposition 13

Tout langage rationnel est reconnaissable par un automate standard.

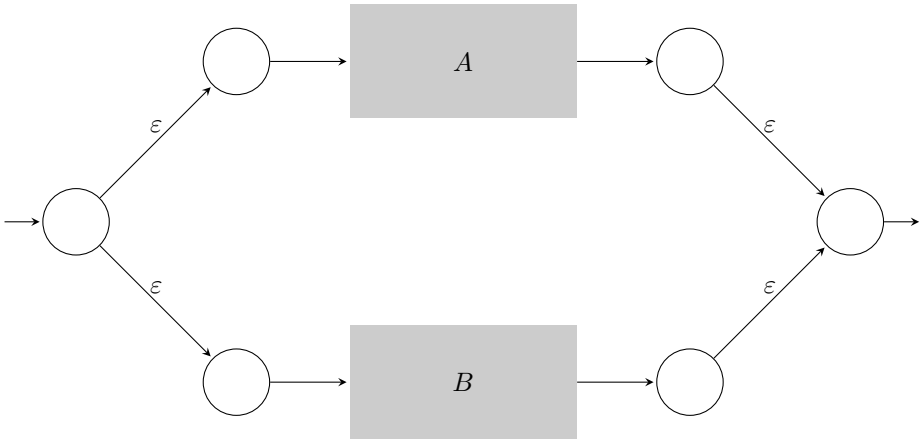
Preuve

On raisonne par induction structurelle.

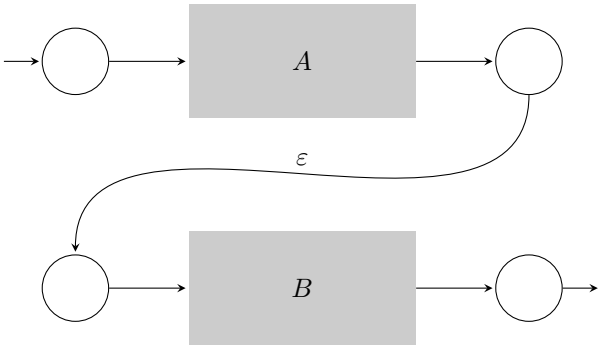
▷ Les langages $\varnothing$, $\{\varepsilon\}$ et $\{x\}$ pour $x \in \Sigma$ sont reconnaissables par les automates standards suivants



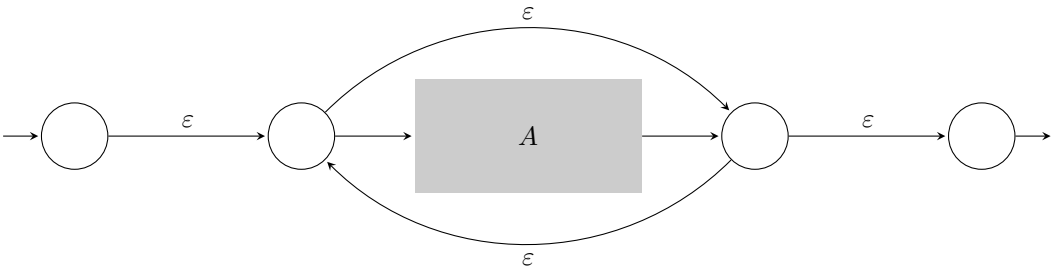
▷ Si deux expressions régulières e et f s'interprètent en des langages reconnus par les automates standards A et B, alors le langage interprétant $e|f$ est reconnu par l'automate standard suivant



▷ Si deux expressions régulières e et f s'interprètent en des langages reconnus par les automates standards A et B, alors le langage interprétant $e.f$ est reconnu par l'automate standard suivant



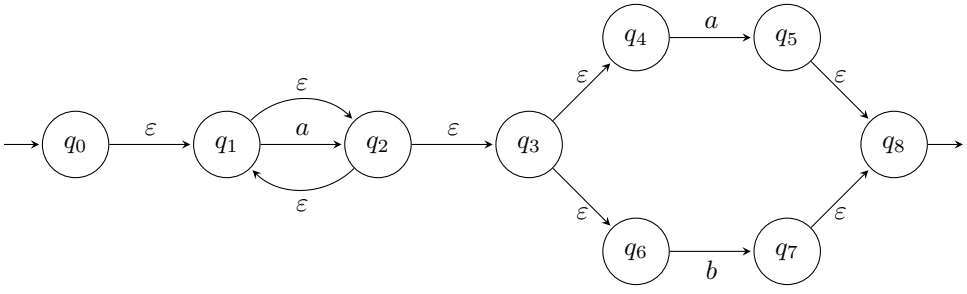
▷ Si une expression régulière e s'interprète en une langage reconnu par l'automate standard A, alors le langage interprétant $e\star$ est reconnu par l'automate standard suivant



■

Exemple 14

L'automate de Thompson associé à l'expression régulière $a\star(a|b)$ est



II.2 Algorithme de Glushkov

Lemme 15

Tout langage rationnel non vide est l'interprétation d'une expression régulière n'utilisant pas le symbole $\varnothing$.

Preuve
Soit L un langage rationnel non vide associée à une expression régulière e. Procédons par induction sur l'expression e.
► Si $e = \varepsilon$ ou $e \in \Sigma$, alors e n'utilise pas le symbole $\varnothing$ et le résultat est établi.
►
▷ Si $e = e_1|e_2$, alors $L = L(e_1) \cup L(e_2)$. Si $L(e_1)$ est vide, alors $L(e_2)$ est non vide ; par hypothèse d'induction, il existe une expression f_2 n'utilisant pas le symbole $\varnothing$ telle que $L = L(e_2) = L(f_2)$. Si les deux langages $L(e_1)$ et $L(e_2)$ sont non vides, il existe des expressions f_1 et f_2 n'utilisant par le symbole $\varnothing$ telles que $L(e_1) = L(f_1)$ et $L(e_2) = L(f_2)$; par conséquent, L est l'interprétation de l'expression régulière $f_1|f_2$ qui n'utilise pas le symbole $\varnothing$.
▷ Si $e = e_1e_2$, alors $L = L(e_1)L(e_2)$. Si l'un des langages $L(e_1)$ ou $L(e_2)$ était vide, alors $L(e)$ serait vide. Ainsi, les deux langages $L(e_1)$ et $L(e_2)$ sont non vides donc, hypothèse d'induction, il existe des expressions f_1 et f_2 n'utilisant par le symbole $\varnothing$ telles que $L(e_1) = L(f_1)$ et $L(e_2) = L(f_2)$; par conséquent, L est l'interprétation de l'expression régulière f_1f_2 qui n'utilise pas le symbole $\varnothing$.
▷ Si $e = e_1\star$, alors $L = L(e_1)^*$. Si $L(e_1)$ est vide, alors $L = L(\varepsilon)$; sinon, il existe une expression f_1 n'utilisant pas le symbole $\varnothing$ telle que $L(e_1) = L(f_1)$ et donc L est l'interprétation de l'expression régulière $f_1\star$ qui n'utilise pas le symbole $\varnothing$.

■

Lemme 16

Soit e une expression régulière qui n'utilise pas le symbole $\varnothing$. Alors il existe une expression régulière f qui n'utilise ni $\varnothing$, ni ε telle que e est équivalente à f ou à $\varepsilon|f$.

Preuve
Procédons encore une fois par induction.
► Si $e \in \Sigma$, alors le résultat est établi.
►
▷ Si $e = e_1|e_2$, alors, on applique l'hypothèse d'induction à e_1 et e_2 et on calcule des expressions équivalentes de la forme voulue pour e. Plus précisément, si f_1 et f_2 sont des expressions régulières qui n'utilisent ni $\varnothing$, ni ε, alors e est équivalente à

$e_1\backslash e_2$	f_2	$\varepsilon\|f_2$
f_1	$f_1\|f_2$	$\varepsilon\|(f_1\|f_2)$
$\varepsilon\|f_1$	$\varepsilon\|(f_1\|f_2)$	$\varepsilon\|(f_1\|f_2)$

▷ Si $e = e_1e_2$, alors on procède à l'identique et on obtient les formes équivalentes suivantes

$e_1\backslash e_2$	f_2	$\varepsilon\|f_2$
f_1	f_1f_2	$f_1\|f_1f_2$
$\varepsilon\|f_1$	$f_2\|f_1f_2$	$\varepsilon\|(f_1\|f_2\|f_1f_2)$

▷ Si $e = e_1\star$, alors il existe une expression régulière f_1 qui n'utilise ni $\varnothing$, ni ε telle que e_1 est équivalente à f_1 ou à $\varepsilon|f_1$. Dans les deux cas, e est équivalente à $f_1\star$.

■

Passons à l'algorithme de Glushkov.

Proposition 17

Il existe un automate qui reconnaît le langage rationnel associé à une expression régulière sans $\varnothing$, ni ε.

Preuve

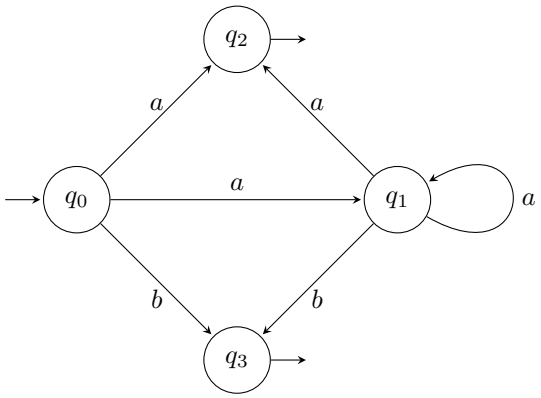
Soit e une expression régulière sur l'alphabet Σ sans $\varnothing$, ni ε et $x_1,\dots,x_n$ les lettres de Σ dans l'ordre d'apparition lors de la lecture de e de gauche à droite (par exemple, pour $e = a.(a|b)\star$, $x_1 = x_2 = a$, $x_3 = b$). Considérons l'automate $(Q = \{q_0,\dots,q_n\}, \{q_0\}, F, \delta)$ défini par

- ▷ F est l'ensemble des états q_i tel que le caractère x_i termine un mot du langage associé à e.
- ▷ pour tous $i,j \in [\![1,n]\!]$, $\delta(q_i,x_j) = q_j$ dès que le facteur x_ix_j peut apparaître dans un mot du langage associé à e.
- ▷ pour tout $i \in [\![1,n]\!]$, $\delta(q_0,x_i) = q_i$ dès que le caractère x_i commence un mot du langage associé à e.

■

Exemple 18

L'automate de Glushkov associé à l'expression régulière $a\star(a|b)$ est



Remarque 19

L'algorithme de Glushkov a un avantage important sur l'algorithme de Thompson ; le nombre d'états est fixé par avance : $n+1$ où n est le nombre de caractères apparaissant dans l'expression. En revanche, la construction de l'automate est plus délicate.

III Théorème de Kleene (sens retour)

Théorème 20 (*Kleene*)

Un langage est rationnel si, et seulement si, il est reconnaissable.

Remarque 21

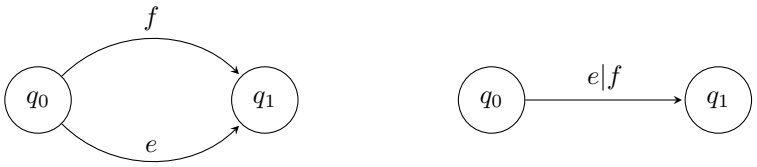
En fait, nous avons déjà vu comment obtenir le langage reconnu par un automate.

- ▷ *On écrit les équations reliant les langages $L_q = \{m \in \Sigma^*,\ \delta^*(q,m) \in F\}$ des mots acceptants depuis l'état q.*
- ▷ *On applique le lemme d'Arden pour résoudre ce système d'équation.*

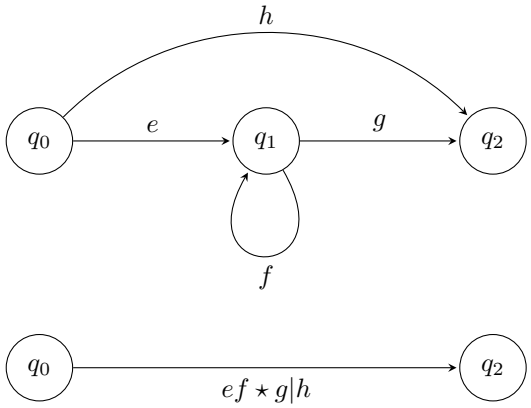
III.1 Algorithme d'élimination de Brzozowski-McCluskey

On décrit ici un algorithme qui permet de déterminer un expression régulière correspondant au langage reconnu par un automate standard.

▷ On standardise l'automate en ajoutant au besoin un état initial et final (voir l'algorithme de Thompson pour un exemple).
▷ On considère chaque transition étiquetée avec une expression régulière (et donc pas seulement avec des caractères ou ε) puis on fusionne les transitions entre deux mêmes états.



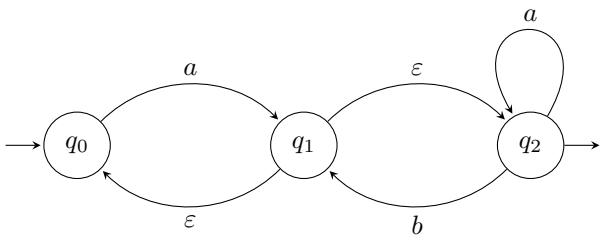
▷ On élimine chaque état autre que l'état initial et l'état final. Pour éliminer un état q_1, on considère tout couple (q_0, q_2) tel qu'il existe une transition de q_0 vers q_1 (étiquetée avec l'expression e) et une transition de q_1 vers q_2 (étiquetée avec l'expression g) ; notons f l'expression de l'éventuelle transition de q_1 vers q_1 et h l'expression de l'éventuelle transition de q_0 vers q_2. On crée alors lors de l'élimination de q_1 (et des transitions associées) une transition de q_0 vers q_2 étiquetée $ef \star g|h$.



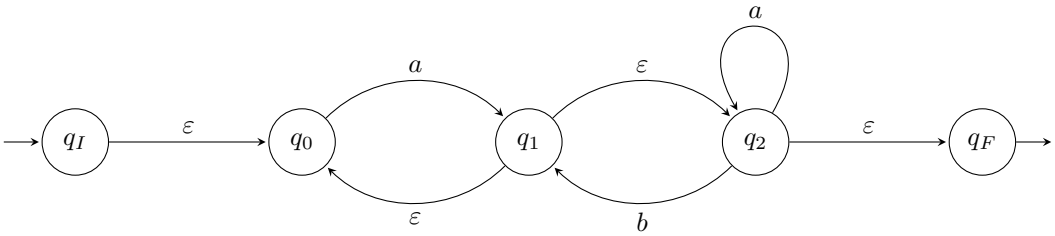
▷ L'expression régulière étiquetant la transition entre l'état initial et l'état final s'interprète en le langage reconnu par l'automate initial.

Exemple 22

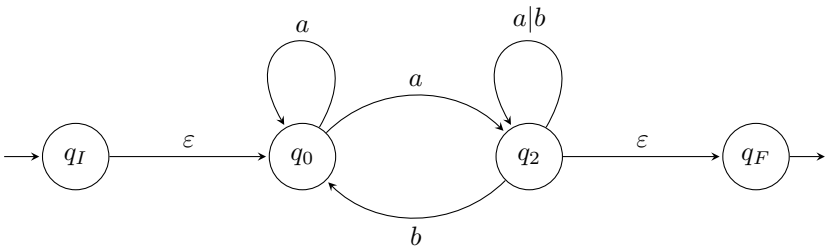
Appliquons cet algorithme à l'automate suivant.



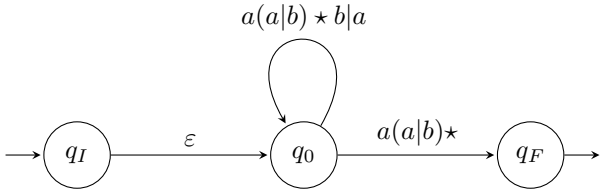
On standardise



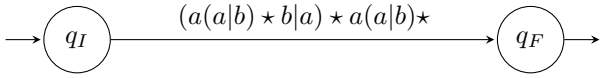
On supprime q_1 en considérant les couples d'états (q_0, q_0), (q_0, q_2), (q_2, q_2) et (q_2, q_0).



On supprime q_2 en considérant les couples d'états (q_0, q_0) et (q_0, q_F)



On supprime q_0 en considérant le couple d'états (q_I, q_F)



On a donc obtenu l'expression régulière $(a(a|b) \star b|a) \star a(a|b)\star$.
Si on avait éliminé les états dans l'ordre q_2, q_0 puis q_1, on aurait trouvé l'expression (équivalente) $a(a \star b|a) \star a\star$.
Remarquons que ces deux expressions restent très complexes puisque le langage reconnu est simplement $a\Sigma^*$ donc est décrit par l'expression régulière $a(a|b)\star$.

IV Exercices

Exercice 50

La hauteur d'étoile d'une expression e est l'entier $h(e)$ défini récursivement à partir des règles suivantes

- ▷ $h(\varnothing) = 0$
- ▷ $h(\varepsilon) = 0$
- ▷ pour tout $x \in \Sigma$, $h(x) = 0$
- ▷ pour toutes expressions régulières e et f, $h(e|f) = \max\{h(e), h(f)\}$
- ▷ pour toutes expressions régulières e et f, $h(e.f) = \max\{h(e), h(f)\}$
- ▷ pour toute expression régulière e, $h(e\star) = h(e) + 1$

Écrire une fonction Caml `hauteur` qui renvoie la hauteur d'étoile d'une expression régulière (de type précisé à la section I.3).

Exercice 51

Montrer que les deux expressions régulières suivantes s'interprètent en le même langage :

$$(a|b)\star \qquad (a\star b)\star a\star$$

Exercice 52

Déterminer l'automate de Thompson et de Glushkov associé à l'expression rationnelle $a(a|b)\star a$.

Exercice 53

Appliquer l'algorithme d'élimination des états pour déterminer une expression régulière correspondant au langage reconnu par l'automate suivant