Automates (2)

option informatique

De l'automate à l'expression rationnelle

Détermination du langage associé à un automate

Soit $\mathcal{A} = (\Sigma, Q, \{q_0\}, F, \delta)$ un automate fini.

- On définit le **langage** L_{q_i} :

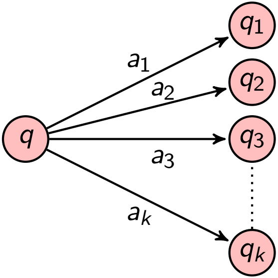
$$L_{q_i} = \left\{ w \in \Sigma^* \mid q_i \xrightarrow{w} q_j \text{ avec } q_j \in F \right\}$$

  C'est l'ensemble des mots w de A^* tels que le calcul de $\mathcal{A}$ sur w à partir de q_i soit acceptant.

- L_{q_0} est le **langage reconnu par** $\mathcal{A}$.
- On désigne par $X_q, X_{q_1}, \ldots, X_{q_k}$ les **expressions rationnelles** qui dénotent les langages $L_q, L_{q_1}, \ldots, L_{q_k}$.

Détermination du langage associé à un automate

- Soit q un état de Q et l'ensemble $\{q_1, \ldots, q_k\}$ des états accessibles depuis cet état



où chaque q_i est un état de Q et chaque a_j est un symbole de l'alphabet Σ. Alors :

  - si $q \notin F$: $X_q = a_1 X_{q_1} + \cdots + a_k X_{q_k}$
  - si $q \in F$: $X_q = a_1 X_{q_1} + \cdots + a_k X_{q_k} + \varepsilon$

Détermination du langage associé à un automate

- En écrivant les équations associées à chaque état $q_i \in Q$, on obtient un **système d'équations associé à l'automate $\mathcal{A}$**.
- Sa résolution à l'aide du **lemme d'Arden** permet d'obtenir une **expression rationnelle qui dénote $L(\mathcal{A})$**.

Lemme d'Arden

Théorème 1 (lemme d'Arden)

Soit Σ un alphabet et A, B deux langages sur Σ. Le langage $X = A^*B$ est le plus petit langage (pour l'inclusion ensembliste) solution de l'équation $X = AX + B$. Si de plus A ne contient pas le mot vide ε, cette solution est unique.

Lemme d'Arden

Démonstration

- Vérifions que $X = A^*B$ est une solution de $X = AX + B$. On a :

$$A(A^*B) + B = A^+B + B = (A^+ + \varepsilon)B = A^*B$$

- Montrons que c'est la plus petite solution pour l'inclusion ensembliste. Si Y est une solution de $Y = AY + B$ différente de A^*B, pour tout entier naturel n, par récurrence, on a :

$$Y = A^{n+1}Y + \sum_{i=0}^{n} A^iB \tag{E}$$

  Par conséquent, pour tout entier naturel n, Y contient tous les A^nB. Ce qui entraîne que toute solution Y autre que A^*B contient ce dernier langage. A^*B est donc la plus petite solution au sens de l'inclusion ensembliste.
- Si A ne contient pas le mot vide ε, montrons que la solution $X = A^*B$ est unique. Soit Y une autre solution. D'après les résultats précédents, $A^*B \subset Y$ et pour tout entier naturel n, Y vérifie l'équation (E).
  Soit w un mot de Y de longueur m. Puisque $w \in Y$, alors $w \in A^{m+1}X + \sum_{i=0}^{m} A^iB$.
  Mais $w \notin A^{m+1}X$ car les mots de $A^{m+1}X$ sont de longueur supérieure ou égale à $(m+1)$. Donc, $w \in \sum_{i=0}^{m} A^iB \subset A^*B$. Ainsi, $Y \subset A^*B$.
  Par conséquent, $Y = A^*B$. Ce qui établit l'unicité de la solution A^*B si $\varepsilon \notin A$.

☛ **Pour s'entraîner :** exercices 1, 2 (automates (2))

Langage local

Généralités

Les **langages locaux**, sous-ensemble des langages rationnels, présentent un intérêt tout particulier lors de la reconnaissance efficace d'une expression rationnelle par un automate.

Ensembles P, S, F

Soit A un alphabet et L un langage sur A. On définit les **trois ensembles fondamentaux** suivants.

- $P(L) = \{a \in A, aA^* \cap L \neq \varnothing\}$
  C'est l'ensemble des premières lettres des mots de L.
- $S(L) = \{a \in A, A^* a \cap L \neq \varnothing\}$
  C'est l'ensemble des dernières lettres des mots de L.
- $F(L) = \{u \in A^2, A^* uA^* \cap L \neq \varnothing\}$
  C'est l'ensemble des facteurs de longueur 2 des mots de L.

Ensembles N et A

Soient à présent les deux deux ensembles suivants.

- $N(L) = A^2 \setminus F(L)$
  C'est l'ensemble des mots de A^2 qui ne sont pas dans $F(L)$.
- $A^* N(L) A^*$
  C'est l'ensemble des mots de A^* qui comportent au moins un facteur dans $N(L)$.

Encore un ensemble

- **L'ensemble des mots de A^* dont la première lettre est dans $P(L)$, la dernière lettre est dans $S(L)$, tous les facteurs de longueur 2 sont dans $F(L)$ et ne sont pas dans $N(L)$ est :**

$$(P(L)A^* \cap A^*S(L)) \setminus (A^*N(L)A^*)$$

- Tout mot non vide de L a sa première lettre dans $P(L)$, sa dernière lettre dans $S(L)$, ses facteurs de longueur 2 dans $F(L)$ (et pas dans $N(L)$). Ce qui se traduit par l'inclusion :

$$L \setminus \{\varepsilon\} \subset (P(L)A^* \cap A^*S(L)) \setminus (A^*N(L)A^*)$$

Langage local

Définition 2 (langage local)

Soit L un langage sur un alphabet A. On dit que L est un **langage local** s'il existe deux parties P et S de A et une partie F de A^2 telles que :

$$L \smallsetminus \{\varepsilon\} = (PA^* \cap A^*S) \smallsetminus (A^*NA^*)$$

avec $N = A^2 \smallsetminus F$.
Si de telles parties existent, nécessairement $P = P(L)$, $S = S(L)$, $F = F(L)$.

  - arrtes@gmail.com

Langage local

- Pour vérifier qu'un mot u appartient à un langage local L, il suffit de vérifier que sa première lettre est dans P, sa dernière lettre dans S et tous ses facteurs de longueur 2 dans F et pas dans N.
- D'une certaine manière, le mot est analysé à travers une **fenêtre** de longueur 2. Les langages locaux sont particulièrement adaptés à la reconnaissance par automates. Les facteurs de longueur 2 du mot ne sont autres que les transitions des automates.

☛ **Pour s'entraîner :** exercice 3 (automates (1))

Linéarisation d'une expression

Expression rationnelle linéaire

Tous les langages rationnels ne sont pas locaux. Toutefois, il existe un critère pour savoir si un langage rationnel est local. Il convient d'abord de définir la notion d'**expression rationnelle linéaire**.

Définition 3 (Expression rationnelle linéaire)

Une **expression rationnelle** e sur un alphabet A est dite **linéaire** si pour tout $a \in A$, le nombre d'occurrences de a dans e est au plus 1.

- Toute expression rationnelle linéaire dénote un langage local.
- La réciproque de ce résultat est fausse. Par exemple, aa^* est un langage local mais il ne peut pas être dénoté par une expression rationnelle linéaire.

Linéarisation d'une expression

- Soit e un expression rationnelle. On dit qu'on **linéarise** e si on remplace chaque occurrence de lettre dans e par son rang d'apparition dans l'expression rationnelle (en lisant de gauche à droite).
- Pour que cette opération soit non destructive, il est nécessaire de conserver dans un tableau la lettre associée à chaque position. Ce tableau définit une **fonction de marquage**.
- La linéarisation est l'une des étapes de la construction d'un automate de reconnaissance d'une expression rationnelle. Nous verrons un algorithme associé appelé **algorithme de Berry-Sethi**.

Linéarisation d'une expression

Soit $A = \{a, b\}$ un alphabet et e l'expression rationnelle $(ab + b)ba$.

- Notons e' l'expression obtenue à partir de e par linéarisation.

$$e' = (r_1 r_2 + r_3) r_4 r_5$$

- Le tableau de la fonction de marquage associée est le suivant.

n	1	2	3	4	5
r_n	a	b	b	b	a

- Détermination des ensembles P, F, S.

$$P = \{r_1, r_3\} \qquad F = \{r_1 r_2, r_2 r_4, r_4 r_5, r_3 r_4\} \qquad S = \{r_5\}$$

De l'expression rationnelle à l'automate

Algorithme de Berry-Sethi et automate de Glushkov

Soit e une expression rationnelle.

- **Linéariser** e en remplaçant chacun de ses symboles par des symboles distincts x_i, numérotés à partir de 1, de gauche à droite.
- Déterminer les ensembles P, F, S du **langage local associé à l'expression linéarisée**.
- **Créer un automate** qui comporte des états q_i associés à chaque x_i et un état initial q_0.
- **Créer une transition** x_j de l'état q_i à l'état q_j si $x_i x_j \in F$.
- À l'aide de la **fonction de marquage**, **supprimer les marques** utilisées pour la linéarisation.
- Éventuellement, **déterminiser l'automate** obtenu.

Exercice

Déterminer un automate qui reconnaît l'expression rationnelle suivante.

$$e = (a + b)^* c$$

- Remarquer que cette expression est déjà linéaire.
- Expression linéarisée :

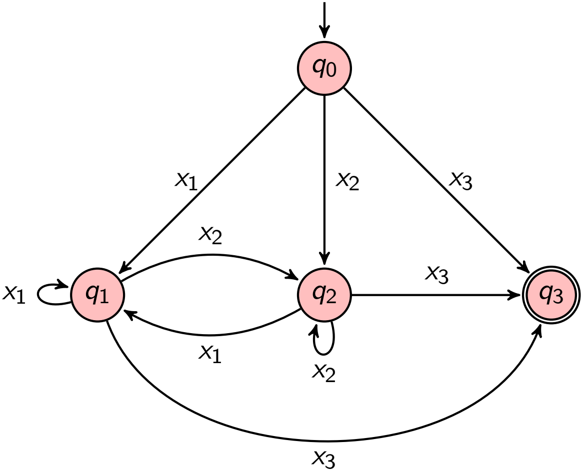
$$e' = (x_1 + x_2)^* x_3$$

- Ensembles P, F, S associés à cette expression linéarisée :

$$P = \{x_1, x_2, x_3\}$$
$$F = \{x_1x_1, x_1x_2, x_1x_3, x_2x_1, x_2x_2, x_2x_3\}$$
$$S = \{x_3\}$$

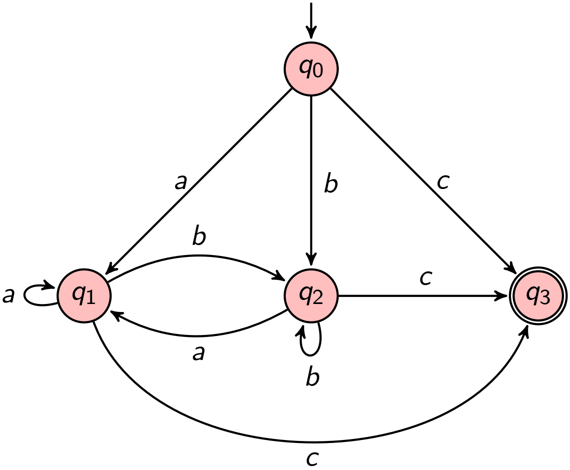
Exemple

Automate de Glushkov **avant suppression des marques**.



Exemple

Automate de Glushkov **après suppression des marques**.



☛ **Pour s'entraîner :** exercice 4 (automates (1))