Notes de cours Option Info MP/MP* : Automates (suite) : le théorème de Kleene

Le théorème de Kleene établit l'équivalence entre les langages rationnels et ceux reconnus par automates finis.

Théorème (Kleene): Un langage L est rationnel si et seulement si il existe un automate $\mathcal{A}$ tel que $L(\mathcal{A}) = L$.

Le but de cette section est de prouver ce théorème, en utilisant de nouvelles notions : les langages locaux, les expressions rationnelles linéaires et les automates locaux.

Langages locaux

Considérons un langage L et définissons:

- $P(L)$ l'ensemble des premières lettres (préfixes de longueur 1) des mots de L.
- $S(L)$ l'ensemble des dernières lettres (suffixes de longueur 1) des mots de L.
- $F(L)$ l'ensemble des facteurs de longueur 2 des mots de L.
- $N(L) = A^2 \setminus F(L)$ l'ensemble des facteurs de longueur 2 "interdits".

Nous avons l'inclusion suivante (essayer de la comprendre):

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

Un langage est dit local si cette inclusion est une égalité. Plus précisément:

Définition: Un langage L est dit local s'il existe deux parties P et S de A et une partie N de A^2 (ensemble des mots de longueur 2) tels que:

$$L \setminus \{\epsilon\} = (PA^* \cap A^*S) \setminus (A^*NA^*).$$

Remarque: Dans ce cas on a nécessairement $P = P(L)$, $S = S(L)$, $N = N(L)$.

Exemples: Sur l'alphabet $A = \{a, b\}$, déterminer pour chacun des langages suivants les langages P, S et N et dire s'ils sont locaux:

1. $L(a^*)$
2. $L((ab)^*)$
3. Les langages $L_1 = L(a^* + (ab)^*)$ et $L_2 = L(a^*(ab)^*)$

Remarque: Le dernier exemple montre que les langages locaux ne sont pas stables par union et concaténation. On va cependant voir d'autres propriétés de clôture.

Propriété 1: L'ensemble des langages locaux est stable par intersection.

Preuve (à compléter): Soient L_1 et L_2 deux langages locaux, et $L = L_1 \cap L_2$. On pose: $P = P(L) =$ ....., $S = S(L) =$ ..... et $N = N(L) =$ ..... Alors on a:

$$\begin{aligned} L \setminus \{\epsilon\} &= (L_1 \setminus \{\epsilon\}) \cap (L_2 \setminus \{\epsilon\}) \\ &= ((P(L_1)A^* \cap A^*S(L_1)) \setminus (A^*N(L_1)A^*)) \cap ((P(L_2)A^* \cap A^*S(L_2)) \setminus (A^*N(L_2)A^*)) \\ &= (P(L_1)A^* \cap A^*S(L_1) \cap P(L_2)A^* \cap A^*S(L_2)) \setminus (A^*N(L_1)A^* \cup A^*N(L_2)A^*) \\ &= (PA^* \cap A^*S) \setminus (A^*NA^*). \end{aligned}$$

Donc L est bien local.

Propriété 2: Si L_1 et L_2 sont deux langages locaux définis sur des alphabets **disjoints**, alors l'union $L_1 \cup L_2$ est encore un langage local.

Preuve (à compléter): Notons A_1 et A_2 les alphabets sur lesquels sont définis L_1 et L_2, alors $L = L_1 \cup L_2$ est défini sur $A = A_1 \cup A_2$. On a: $P(L) =$ ....., $S(L) =$ ..... et $N(L) =$ ..... On doit montrer l'inclusion $(P(L)A^* \cap A^*S(L)) \setminus (A^*N(L)A^*) \subset L$ (l'inclusion réciproque est toujours vraie). Considérons donc un mot $u \in (P(L)A^* \cap A^*S(L)) \setminus (A^*N(L)A^*)$ que l'on décompose en lettres $u = a_0 \dots a_n$:

-$a_1 \in P(L) = P(L_1) \cup P(L_2)$, on peut supposer sans perte de généralité que $a_1 \in P(L_1)$, alors $a_1 \in A_1$.

-$a_1.a_2 \in A^2 \setminus N(L) = F(L_1) \cup F(L_2)$, or $a_1 \in A_1$ et les alphabets A_1 et A_2 sont disjoints, donc nécessairement $a_1a_2 \in F(L_1)$ et $a_2 \in A_1$.

-De proche en proche, on montre que $a_ia_{i+1} \in F(L_1)$ et $a_i \in A_1$ pour tout i.

-Enfin, $a_n \in S(L) = S(L_1) \cup S(L_2)$ et $a_n \in A_1$ donc $a_n \in S(L_1)$.

Finalement $u \in (P(L_1)A^* \cap A^*S(L_1)) \setminus (A^*N(L_1)A^*) = L_1$ car L_1 est local. Donc $u \in L_1 \cup L_2$ et L est bien un langage local.

Propriété 3: Si L_1 et L_2 sont deux langages locaux définis sur des alphabets disjoints, alors la concaténation $L_1.L_2$ est encore un langage local.

Preuve (à compléter): Reprenons les mêmes notations que précédemment, on a cette fois:

$$P(L_1L_2) = \begin{cases} ..... & \text{si } ..... \\ ..... & \text{sinon} \end{cases}$$

$$S(L_1L_2) = \begin{cases} ..... & \text{si } ..... \\ ..... & \text{sinon} \end{cases}$$

$F(L_1L_2) =$ .....

Comme précédemment on considère un mot $u \in (P(L)A^* \cap A^*S(L)) \setminus (A^*N(L)A^*)$ et on le décompose en lettres $u = a_0 \dots a_n$. On traite différents cas:

- Si $a_0 \in A_2$, alors $\epsilon \in L_1$ et $a_2 \in P(L_2)$. De proche en proche on montre que $a_ia_{i+1} \in F(L_2)$, $a_i \in A_2$ pour tout i et $a_n \in S(L_2)$, donc $u \in L_2$ car L_2 est local.

- Si $a_0 \in A_1$, alors $a_0 \in P(L_1)$. Notons $a_0 \dots a_k$ le plus long préfixe de u qui soit dans A_1^*. On montre de proche en proche que $a_ia_{i+1} \in F(L_1)$ pour $i < k$. Puis deux cas se présentent:

  - Si $k = n$ alors $a_n \in S(L_1)$, donc $u \in L_1$ car L_1 est local.
  - Si $k < n$, on a $a_ka_{k+1} \in S(L_1)P(L_2)$ donc $a_k \in S(L_1)$ et $a_{k+1} \in P(L_2)$. On prouve alors que $a_0 \dots a_k \in L_1$ et $a_{k+1} \dots a_n \in L_2$ avec les mêmes arguments.

  Finalement $u \in L_1.L_2$ qui est bien local.

Propriété 4: Si L est un langage local, L^* est aussi un langage local.

Preuve (à compléter): On a $P(L^*) =$ ....., $S(L^*) =$ ....., $F(L^*) =$ ..... On considère un mot $u = a_0 \dots a_n \in (P(L^*)A^* \cap A^*S(L^*)) \setminus (A^*N(L^*)A^*)$. En considérant ses facteurs de longueur 2 dans $S(L)P(L)$, on en déduit une décomposition de u en mots dans $(P(L)A^* \cap A^*S(L)) \setminus (A^*N(L)A^*)$ donc dans L car L est local. Donc $u \in L^*$ et L^* est local.

Expressions rationnelles linéaires

On a vu que les langages définis par des expressions rationnelles n'étaient pas tous locaux, en revanche ce sera le cas pour une classe particulière d'expressions rationnelles: les expressions rationnelles linéaires.

Définition: Une expression rationnelle e est dite linéaire lorsque toute lettre de A apparaît au plus une fois dans e.

Par exemple, l'expression $(ab)^*$ est linéaire mais pas $(ab)^*a^*$. Le théorème les reliant aux langages locaux est le suivant:

Théorème: Toute expression rationnelle linéaire définit un langage local.

Preuve: Par induction structurelle en utilisant les propriétés de clôture ci-dessus.

Remarque: La réciproque de ce théorème est fausse: le langage $L(aa^*)$ est local mais ne peut pas être représenté par une expression rationnelle linéaire.

Algorithmes de calcul des ensembles P, S et F

On procède par induction structurelle et on utilise les résultats obtenus lors des preuves de propriétés de clôture. On voit qu'on l'on a besoin d'une fonction qui détermine si le mot vide est dans le langage d'une expression rationnelle. *(à faire en TP)*

Automates locaux

Définition: Un automate fini déterministe $\mathcal{A} = (A, Q, q_0, F, E)$ est dit local si pour toute lettre $a \in A$, il existe un état $q \in Q$ tel que toutes les transitions étiquetées par a arrivent dans q. Il est dit standard s'il n'existe pas de transition dont la destination est q_0.

Théorème: Si L est un langage local défini sur un alphabet A, le langage $L \setminus \{\epsilon\}$ est reconnaissable par un automate local standard.

Preuve (à compléter): On pose $P = P(L)$, $S = S(L)$, $F = F(L)$. On considère l'automate $\mathcal{A} = (A, A \cup \{\epsilon\}, \{\epsilon\}, ..., E)$ où E est l'ensemble des $(.., .., ..)$ avec ..... et $(.., .., ..)$ avec ..... Par construction, un mot $u = a_0a_1 \dots a_n$ est reconnu par $\mathcal{A}$ si et seulement si $a_0 \in P$, $a_ia_{i+1} \in F$ pour tout $0 \leqslant i \leqslant n-1$ et $a_n \in S$. Comme L est local, on a bien $L(\mathcal{A}) = L \setminus \epsilon$.

Question: Si L contient le mot vide, comment obtenir un automate reconnaissant L ?

Exemple: Construire un automate local reconnaissant le langage défini par l'expression rationnelle linéaire $(a + b)^*c$.

Compléter: Le nombre d'états de l'automate local est égal à ..... et son nombre de transitions, qui est égal à ....., est majoré par .....

Automate de Glushkov et algorithme de Berry-Sethi

Nous allons voir dans cette section comment construire un automate reconnaissant le langage défini par une expression rationnelle dans le cas général (qu'on appelle automate de Glushkov). On commence par linéariser l'expression rationnelle:

Linéarisation d'une expression rationnelle: Cela consiste à numéroter les lettres qui apparaissent afin d'obtenir une expression rationnelle linéaire.

Exemple: Linéariser l'expression $ab + (ac)^*$.

Algorithme de Berry-Sethi: Etant donnée une expression rationnelle e, pour déterminer l'automate de Glushkov associé:

1. On linéarise l'expression, on obtient une expression rationnelle linéaire e'.
2. On détermine les ensembles P, S et F associés au langage local $L(e')$.
3. On détermine un automate local reconnaissant $L(e')$.
4. On supprime les marquages des étiquettes.

Fait: L'automate obtenu par l'algorithme de Berry-Sethi a pour langage reconnu $L(e)$ (essayer de le prouver).

Remarques (à compléter): L'automate de Glushkov est en général non déterministe (mais on peut le déterminiser). Il possède ..... états où A_e est l'alphabet "étendu" obtenu en faisant le marquage (ce qui correspond au nombre total de lettres dans e en comptant les répétitions), et au plus ..... transitions.

Exemple: Trouver un automate reconnaissant le langage associé à l'expression rationnelle $(ab+b)^*ba$.

Cela conclut la première partie de la preuve du théorème de Kleene.

Programmer en TP l'algorithme de Berry-Sethi.

Des automates aux expressions rationnelles (HP)

Pour prouver l'autre sens du théorème de Kleene, il nous faut montrer que le langage reconnu par un automate fini est rationnel. Pour cela, on utilise des automates dits "généralisés", c'est-à-dire étiquetés non plus par des lettres mais par des expressions rationnelles. Le principe de l'algorithme est le suivant: on ajoute un unique état initial et un unique état terminal, puis on élimine un à un les états jusqu'à ce qu'il n'en reste que deux (l'initial et le terminal) *(voir exemple au tableau).*

Application : propriétés de clôture des langages rationnels

A l'aide du théorème de Kleene et des automates on peut montrer facilement que les langages rationnels sont clos par intersection et complémentation:

Théorème: Soit L un langage rationnel, alors son complémentaire $\bar{L}$ est aussi rationnel.

Preuve (à faire): .....

Théorème: Soient L_1 et L_2 deux langages rationnels, alors $L_1 \cap L_2$ est aussi rationnel.

Preuve (à faire): .....

Remarque: Par construction des expressions rationnelles, on sait aussi que les langages rationnels sont stables par union, concaténation et passage à l'étoile (mais on pourrait trouver des automates reconnaissant l'union, la concaténation de deux langages ou l'étoile d'un langage).

Signalons une dernière propriété de clôture, le passage au miroir:

Théorème: Si L est un langage rationnel, son miroir (c'est-à-dire l'ensemble des miroirs des mots de L) est aussi rationnel.

Preuve (à faire): .....

Exemple de langage non rationnel : le langage L des mots sur $\{a, b\}$ comportant plus de a que de b. On montre avec la méthode de séparation d'états qu'un automate le reconnaissant aurait nécessairement un nombre infini d'états.

Compléments (HP)

Lemme de l'étoile

Pour prouver qu'un langage est rationnel, nous avons vu qu'il y a deux solutions: exhiber une expression rationnelle qui le définit ou un automate fini (déterministe ou non) qui le reconnaît. Par contre, pour prouver qu'un langage n'est pas rationnel, nous disposons du résultat suivant:

Lemme de l'étoile: Soit L un langage rationnel. Alors il existe un entier k tel que tout mot $m \in L$ de longueur supérieure ou égale à k se décompose $m = uvw$ avec $|uv| \leqslant k$, $v \neq \epsilon$ et pour tout $n \in \mathbb{N}$, $uv^n w \in L$.

Preuve: Soit $\mathcal{A}$ un automate fini reconnaissant L. Notons q son nombre d'états et $k = q + 1$. Si $m = a_1 \dots a_p \in L$ est de longueur p supérieure ou égale à q, soit $q_0 \to q_1 \to \cdots \to q_p$ un calcul acceptant pour m dans $\mathcal{A}$. Il comporte plus d'étapes que le nombre d'états, donc passe nécessairement deux fois par le même état. Notons i et j les premiers indices tels que $q_i = q_j$ $(i < j)$. Alors si on pose $u = a_1 \dots a_i$, $v = a_{i+1} \dots a_j$ et $w = a_{j+1} \dots a_p$, on a bien $v \neq \epsilon$ car $i < j$, $|uv| \leqslant k$ car on a pris i et j minimaux, et par construction, pour tout $n \in \mathbb{N}$, $uv^n w \in L$.

Exemple 1: Le langage $L = \{a^n b^n, n \in \mathbb{N}\}$ n'est pas rationnel. Le montrer avec le lemme de l'étoile.

Exemple 2: Si A est un alphabet avec au moins deux lettres a et b, l'ensemble des carrés $L = \{m^2, m \in A^*\}$ n'est pas rationnel. Le montrer avec le lemme de l'étoile.

Lemme d'Arden

Le lemme d'Arden nous donne une autre manière de trouver une expression rationnelle associée à un automate.

Lemme d'Arden: Soient L et M deux langages sur un alphabet A. On considère l'équation $(E) : X = LX + M$, d'inconnue $X \subset A^*$. Alors:

1. Le langage $X = L^*M$ est solution de (E), et c'est la plus petite solution pour l'inclusion.
2. Si L ne contient pas le mot vide, c'est l'unique solution de (E).

Preuve: Le langage L^*M est bien solution car $L^* = LL^* + \epsilon$. Si X est solution, on montre par récurrence sur n que $X = L^{n+1}X + \sum_{k=0}^{n} L^k M$. En particulier pour tout $k \in \mathbb{N}$, $L^k M \subset X$, donc $L^*M \subset X$ ce qui montre la minimalité de la solution L^*M. Si L ne contient pas ϵ, montrons que c'est l'unique solution. Soit $m \in X$, de longueur n. Alors $m \in L^{n+1}X + \sum_{k=0}^{n} L^k M$. Comme L ne contient pas le mot vide, les mots de $L^{n+1}X$ sont de longueur $> n$, donc $m \in \sum_{k=0}^{n} L^k M \subset L^*M$, donc $X \subset L^*M$ et comme on avait déjà l'autre inclusion, $X = L^*M$.

voir exemple au tableau