909 ①

NOM : BERNARD **Prénom** : Sophie **Jury** :

~~Algèbre~~ ← Entourez l'épreuve → ~~Analyse~~ Info

Sujet choisi : 909. Langages rationnels. Exemples et applications

Autre sujet :

On note A un alphabet

I_ PREMIERES PROPRIETES DES LANGAGES RATIONNELS

1. Définition et lien avec les expressions rationnelles

Def 1 : Soient K et L des langages. On définit :
(i) la concaténation : $KL = \{uv, u \in K, v \in L\}$
(ii) l'étoile : $L^0 = \{\varepsilon\}$, $L^{k+1} = LL^k$ et $L^* = \bigcup_{n \geq 0} L^n$ $(k \geq 0)$

Def 2 : L'ensemble $Rat(A^*)$ des langages rationnels sur A est la plus petite famille de langages vérifiant :
(i) $\emptyset \in Rat(A^*)$, $\forall a \in A$, $\{a\} \in Rat(A^*)$
(ii) $Rat(A^*)$ est close par les trois opérations rationnelles (union, concaténation, étoile)

Ex 3 : Les langages suivants sont rationnels : $\{\varepsilon\} = \emptyset^*$, A, le langage des mots de longueur paire : $(AA)^*$, les langages contenant un certain motif $m \in A^*$: $A^* m A^*$.

Def 4 : On définit de manière inductive l'ensemble des expressions rationnelles $RatE(A^*)$ par :
(i) $\emptyset \in RatE(A^*)$, $\varepsilon \in RatE(A^*)$, et $\forall a \in A$, $a \in RatE(A^*)$
(ii) $\forall E, F \in RatE(A^*)$, $E+F \in RatE(A^*)$, $EF \in RatE(A^*)$ et $E^* \in RatE(A^*)$

Def 5 : Soit $E \in RatE(A^*)$. On note $\mathcal{L}(E)$ le langage reconnu par E, défini de manière inductive par :
(i) $\mathcal{L}(\emptyset) = \emptyset$, $\mathcal{L}(\varepsilon) = \{\varepsilon\}$ et $\forall a \in A$, $\mathcal{L}(a) = \{a\}$
(ii) $\forall E, F \in RatE(A^*)$, $\mathcal{L}(E+F) = \mathcal{L}(E) \cup \mathcal{L}(F)$, $\mathcal{L}(EF) = \mathcal{L}(E)\mathcal{L}(F)$ et $\mathcal{L}(E^*) = (\mathcal{L}(E))^*$.

Rmq 6 : Les expressions rationnelles sont un outil syntaxique très pratique pour utiliser sur ordinateurs. C'est seulement la définition 5 qui leur donne une sémantique.

Th 7 : Soit L un langage sur A.
$L \in Rat(A^*) \iff \exists E \in RatE(A^*), \mathcal{L}(E) = L$

Ex 8 : En posant $L = A^*$ avec $A = \{a, b\}$ alors :
$L = \mathcal{L}((a+b)^*) = \mathcal{L}((a^*b)^*a^*)$

Rmq 9 : Il n'y a pas forcément unicité de l'expression rationnelle associée à un langage (voir ex 8). Cela pose deux problèmes :
+ peut-on décider si pour $E, F \in RatE(A^*)$, alors $\mathcal{L}(E) = \mathcal{L}(F)$?
+ peut-on savoir si pour $E \in RatE(A^*)$, alors $\mathcal{L}(E) = A^*$?
× On utilise souvent les notations des expressions rationnelles pour les langages rationnels, au point de ne parfois pas faire la distinction.

2. Lien avec les automates finis

Def 10 : Un automate $\mathcal{A}$ sur un alphabet A est un quintuplet (Q, A, Δ, I, F) où Q est fini (ensemble des états), $I \subset Q$ (états initiaux), $F \subset Q$ (états finaux), et $\Delta \subset Q \times A \times Q$ (ensemble des transitions). Si $(p, a, q) \in \Delta$ on note $p \xrightarrow{a} q$ la transition.

Def 11 : Le langage reconnu par un automate $\mathcal{A}$ est l'ensemble ~~des~~ des mots qui sont l'étiquette d'un chemin fini acceptant de cet automate, c'est-à-dire d'une suite finie de transitions consécutives dont le premier état est initial, et le dernier état est final. On le note $\mathcal{L}(\mathcal{A})$.

Ex 12 : L'automate $\mathcal{A}_1 = (\{1,2\}, \{a,b\}, \{(1,b,1), (1,a,2), (2,b,2), (2,a,1)\}, \{1\}, \{1\})$ est représenté en annexe (Fig 1) et reconnaît le langage $(ab^*a + b)^*$.

Th 13 (Kleene) : Soit L un langage sur A.
$L \in Rat(A^*) \iff \exists \mathcal{A}$ automate fini sur A, $\mathcal{L}(\mathcal{A}) = L$.

Algo 14 (Thompson) : Construction inductive d'un automate avec ε-transitions reconnaissant le même langage qu'une expression régulière. On peut par la suite modifier l'automate pour qu'il n'ait plus d'ε-transitions. (Voir annexe Fig 2 pour la construction inductive).

Algo 15 (McNaughton et Yamada) : Calcul du langage rationnel reconnu par un automate fini en commençant par numéroter les états de 1 à n, puis calculer récursivement $L_{pq}^{(k)}$ pour $k \geq 0$, le langage reconnu entre les états p et q en ne passant que par des états dont le numéro est $\leq k$. On a alors $\bigcup_{i \in I, f \in F} L_{ij}^{(n)}$.

Def 16 : Un automate (Q, A, Δ, I, F) est déterministe s'il n'a qu'un seul état initial et si $\forall p \xrightarrow{a} q, p \xrightarrow{a} q' \in \Delta, q = q'$.
Un automate (Q, A, Δ, I, F) est complet si pour toute paire $(p, a) \in Q \times A$, il existe $q \in Q$, $p \xrightarrow{a} q$ est une transition.

Prop 17 : L'ensemble des langages reconnus par des automates finis déterministes ou déterministes complets est exactement l'ensemble des langages rationnels.

Ex 18 : Les automates de la figure 3 reconnaissent tous les trois le langage $(a+b)^*ab$

Prop 19 : $Rat(A^*)$ est clos par complémentation et intersection.

DEV 1 : Prop 20 : Soit $E \in RatE(A^*)$. Le problème de l'universalité de $\mathcal{L}(E)$ (savoir si $\mathcal{L}(E) = A^*$) est PSPACE-complet.

II - PROPRIETES LIÉES AUX AUTOMATES FINIS

1 - Lemmes de l'étoile

Lemme 21 (de l'étoile ou pompage ou itération): Soit L un langage rationnel sur A. Il existe n entier tel que pour tout mot f de L, si $|f| \geq n$ alors : $\exists u, v, w \in A^*$, $v \neq \varepsilon$, $f = uvw$ et $uv^*w \subset L$.

Ex 22: $L_1 = \{a^n b^n, n \geq 0\}$ n'est pas rationnel

Rmq 23: On a seulement une condition nécessaire pour être un langage rationnel.

Ex 24: $A_1 = \{a, b\}$, $L_1 + A_1^* ba A_1^*$ vérifie les conditions du lemme de l'étoile mais n'est pas rationnel.

Lemme 25 (de l'étoile fort ou par blocs): Soit L un langage rationnel sur A. Il existe n entier telque pour tout mot $f = uv_1v_2 \dots v_n w \in L$ et $v_i \neq \varepsilon$, alors: $\exists i, j$, $0 \leq i < j \leq n$, $uv_1 \dots v_i(v_{i+1} \dots v_j)^* v_{j+1} \dots v_n w \subset L$.

Def 26: Un langage L vérifie la propriété σ_k (resp. σ_k') pour $k > 0$ si pour toute factorisation $f = uv_1 \dots v_k w$ où tous les v_i sont des mots non vides, il existe deux indices $0 \leq i < j \leq k$ telsque :
Pour σ_k : $\forall n \geq 0$, $f \in L \Leftrightarrow uv_1 \dots v_i(v_{i+1} \dots v_j)^n v_{j+1} \dots v_k w \in L$
Pour σ_k' : $f \in L \Leftrightarrow uv_1 \dots v_i v_{j+1} \dots v_k w \in L$.

Th 27 (Ehrenfeucht, Parikh et Rozenberg): Soit L un langage sur A. Les propositions suivantes sont équivalentes:
(i) L est rationnel
(ii) Il existe $k > 0$ tel que L vérifie σ_k
(iii) Il existe $k > 0$ tel que L vérifie σ_k'

Rmq 28: On a donc deux nouvelles caractérisations des langages rationnels, et deux lemmes très utiles pour montrer la non-rationalité d'un langage.

2 - Résolution d'équations

Lemme 29 (Arden): Soient K et L deux langages sur A. On considère l'équation $X = KX + L$ d'inconnue X un langage.
(i) Si $\varepsilon \notin K$, la solution est $X = K^*L$.
(ii) Si $\varepsilon \in K$, les solutions sont de la forme $X = K^*(L+Y)$ où $Y \subset A^*$.

Algo 30 (Elimination de Gauss): Calcul du langage reconnu par un automate par utilisation systématique du lemme d'Arden. On note X_q l'ensemble des mots qui étiquettent un chemin de q à un état final. On obtient alors un système d'équations et finalement $\bigcup_{i \in I} X_i$, le langage reconnu par l'automate.

3 - Par les résiduels : automate minimal

Def 31: Soit L un langage sur A. Le résiduel ou quotient à gauche de L par un mot $u \in A^*$ est $u^{-1}L = \{v, uv \in L\}$. Le résiduel de L par un langage $K \subset A^*$ est $K^{-1}L = \bigcup_{k \in K} k^{-1}L$. L'ensemble des résiduels de L est $(A^*)^{-1}L$.

Prop 32: Pour toute lettre a de A, tous mots u, v et w de A^*, et tous langages K et L sur A, on a :
(i) $w^{-1}(K+L) = w^{-1}K + w^{-1}L$
(ii) $a^{-1}(KL) = (a^{-1}K)L + \varepsilon(K) a^{-1}L$ avec $\varepsilon(K) = \varepsilon \cap L$.
(iii) $w^{-1}(KL) = (w^{-1}K)L + \sum_{uv=w} \varepsilon(u^{-1}K) v^{-1}L$
(iv) $a^{-1}(L^*) = (a^{-1}L)L^*$
(v) $w^{-1}(L^*) = \sum_{uv=w} \varepsilon(u^{-1}L^*)(v^{-1}L)L^*$
(vi) $(uv)^{-1}L = v^{-1}(u^{-1}L)$.

Ex 33: On s'intéresse aux résiduels de $L = (ab^*a+b)^*$
$a^{-1}L = b^*aL$; $b^{-1}L = L$; $(b^*)^{-1}L = L$; $(ab^*)^{-1} = b^*aL$
$(b^*ab^*)^{-1}L = b^*aL$ et $(ab^*ab^*)^{-1}L = L$.
L n'a que deux résiduels : L et b^*aL.

Prop 34: Soit L un langage rationnel sur A. L a un nombre fini de résiduels

Def 35: Soit L un langage rationnel sur A. Son automate minimal est $\mathcal{A}_L = (Q, A, \Delta, I, F)$ avec : $Q = \{u^{-1}L, u \in A^*\}$, $I = \{L\}$, $F = \{u^{-1}L, u \in L\}$ et $\Delta = \{u^{-1}L \xrightarrow{a} (ua)^{-1}L, u \in A^*, a \in A\}$

Ex 36: En annexe, à la Fig 4 est représenté l'automate minimal du langage de l'exemple 33 : $L = (ab^*a+b)^*$

Lemme 37: Pour tout langage rationnel L sur A, $\mathcal{A}_L$ accepte exactement L, et il n'y a pas d'automate avec moins d'états qui reconnaît L.

Lemme 38: Un langage L est rationnel si et seulement si il a un nombre fini de résiduels.

4 - Construction de l'automate minimal

Def 39: Soit $\mathcal{A} = (Q, A, \Delta, I, F)$ un automate déterministe complet. On note pour toute paire $(p, a) \in Q \times A$, $p \cdot a$ l'unique état tel que $p \xrightarrow{a} q$ soit une transition de $\mathcal{A}$, puis, pour tout mot u de A^*, $p \cdot (ua) = (p \cdot u) \cdot a$. On appelle congruence de Nérode la relation d'équivalence : $\forall q, q' \in Q$, $q \sim q'$ si $\forall w \in A^*$, $q \cdot w \in F \Leftrightarrow q' \cdot w \in F$

Def 40: Soit $\mathcal{A} = (Q, A, \Delta, \{i\}, F)$ un automate déterministe complet. L'automate quotient $\mathcal{A}/\sim$ est $(Q/\sim, A, \Delta', \{[i]\}, \{[f], f \in F\})$ avec $\Delta' = \{[q] \xrightarrow{a} [q \cdot a], q \in Q, a \in A\}$.

Ex 41: En annexe, à la Fig 5, est représenté l'automate quotient du langage de l'exemple 33 : $L = (ab^*a+b)^*$

Prop 42: Soit L un langage sur A et $\mathcal{A}$ un automate déterministe complet reconnaissant L. Alors l'automate quotient $\mathcal{A}/\sim$ est isomorphe à l'automate minimal $\mathcal{A}_L$.

Prop 43: Soient E et F deux expressions rationnelles. Le problème de savoir si $\mathcal{L}(E) = \mathcal{L}(F)$ est décidable

Algo 44 (Hopcroft): Cet algorithme permet de calculer l'automate quotient de la congruence de Nérode d'un automate déterministe complet $\mathcal{A} = (Q, A, \Delta, I, F)$, en temps $O(|\mathcal{A}|\,|Q| \log |Q|)$, par raffinements de partitions.

DEV 2 :

III - AUTRES CARACTERISATIONS ET APPLICATIONS

1 - Lien avec les morphismes de monoïdes

Def 45: Un monoïde est un ensemble Π muni d'une loi de composition interne associative qui possède un élément neutre, noté 1_Π. Soient Π et Π' deux monoïdes. Un morphisme de Π dans Π' est une application $\mu : \Pi \to \Pi'$ vérifiant :
$\mu(1_\Pi) = 1_{\Pi'}$ et $\forall x, y \in \Pi$, $\mu(xy) = \mu(x)\mu(y)$

Def 46: Un langage L sur A est reconnu par un morphisme de monoïde $\mu : A^* \to \Pi$ s'il existe une partie P de Π telle que $L = \mu^{-1}(P)$. Un monoïde Π reconnaît le langage L s'il existe un morphisme $\mu : A^* \to \Pi$ qui reconnaît L.

Prop 47: Soit L un langage sur A. L est rationnel si et seulement si il ~~existe~~ est reconnu par un monoïde fini.

Ex 48: $\mathbb{Z}/2\mathbb{Z}$ reconnaît $(ab^*a+b)^*$, car :
$\mu : A^* \to \mathbb{Z}/2\mathbb{Z}$, $w \mapsto |w|_a \bmod 2$ est un morphisme de monoïde et
$\mu^{-1}(\{0\}) = (ab^*a+b)^*$.

Prop 49: Soient B un alphabet et μ un morphisme de A^* dans B^*. Si L (resp. K) un langage sur A (resp. B), est rationnel, alors $\mu(L)$ (resp. $\mu^{-1}(K)$) est rationnel

Appl S0: Soient B un alphabet, K un langage sur B. Soit σ une substitution de A^* dans B^*, c'est à dire un morphisme de A^* dans $\mathcal{P}(B^*)$, les parties de B^*, munies de l'opération de produit. Alors $\{w, \sigma(w) \cap K \neq \emptyset\}$ et $\{w, \sigma(w) \subset K\}$ sont rationnels

2. Recherche de motifs dans un texte

On veut chercher dans un document un mot, ou son nombre d'occurrences. Cela revient à essayer de reconnaitre le langage rationnel composé exactement de ce mot.

Algo S1 (Automate de Simon): C'est une amélioration de l'algorithme de Knuth-Morris-Pratt avec des automates, en deux phases. La première phase, dite de préparation, est de complexité $O(m)$ en espace et en temps, la seconde, de recherche, en $O(m+n)$ en temps, où m est la longueur du motif à chercher et n la longueur du texte dans lequel chercher.

Appl S2: En traitement de texte, pour implémenter la recherche ou le remplacement d'un motif. C'est il grep de linux qui permet de rechercher à partir d'une expression rationnelle.

3. Arithmétique de Presburger

Appl S3: On considère la théorie du premier ordre

DEV3: des entiers munis de l'addition dont la syntaxe est : $\phi = (x=y) \mid (x+y=z) \mid \phi \vee \phi' \mid \neg\phi \mid \exists x \phi$.
Alors pour toute formule close ϕ, il est décidable de savoir si ϕ est vraie.

4. Analyse lexicale

Appl S4: C'est la première étape d'un compilateur. On veut reconnaitre un ensemble de langages rationnels (les mots-clés, identificateurs, nombres,...) donc un langage rationnel. On peut le faire grâce à l'équivalence langages rationnels - automates et les algorithmes associés. C'est le principe des outils comme lex et yacc.

5. Lien avec les grammaires régulières.

Def S5: Une grammaire régulière à gauche (resp. à droite) est un quadruplet $G=(N, A, P, S)$ où N est un alphabet dit de symboles non terminaux, A un alphabet de symboles terminaux, P est l'ensemble des règles de production avec $P \subset N \times (NA + A + \varepsilon)$ (resp. $P \subset N \times (AN + A + \varepsilon)$), et S est le symbole initial, et $S \in N$.

Lemme S6: Les langages reconnus par une grammaire régulière sont exactement les langages rationnels.

Ex S7: La grammaire $G = (\{S, T\}, \{a, b\}, \{(S, b), (S, \varepsilon), (S, bS), (T, a), (S, aT), (T, bT), (T, aS)\}, S)$ reconnait le langage $(ab^*a + b)^*$ qui est rationnel. On peut aussi le représenter sous forme d'automate (Fig 6).

Fig 1_Ex 12 :

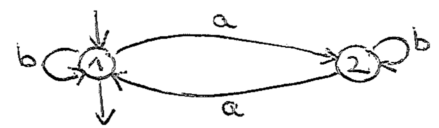
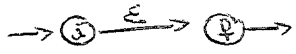


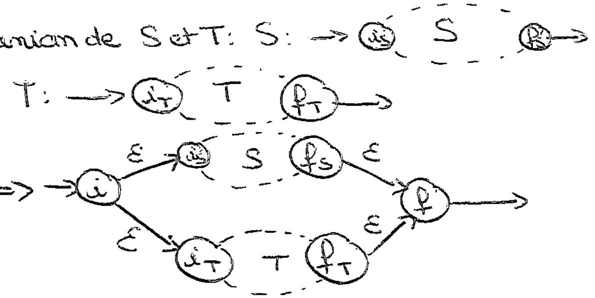
Fig 2_Algo 14 :

ε :

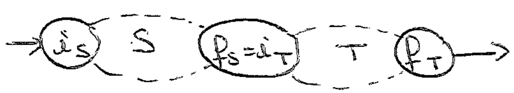


$a \in A$:

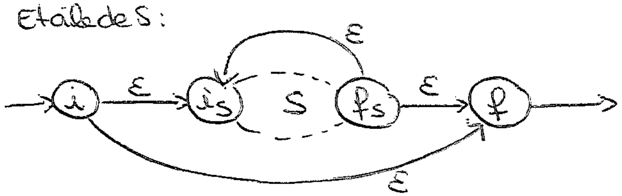
union de S et T :



concaténation de S et T :



Etoile de S :



~~Fig 3_Ex 18~~ :

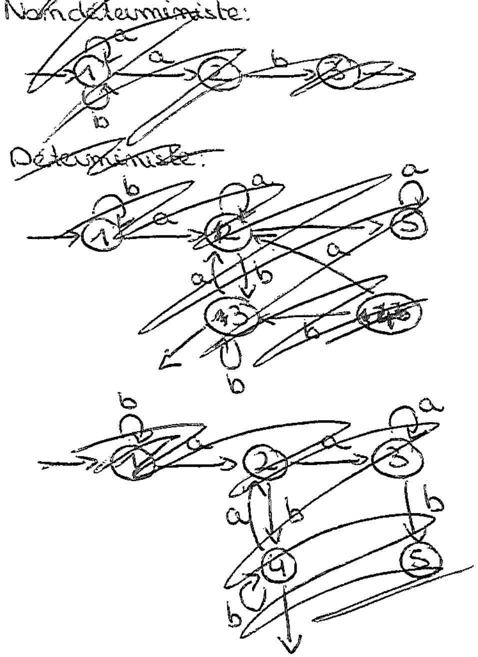
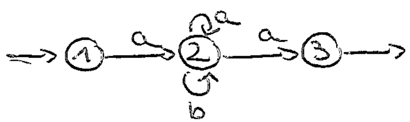
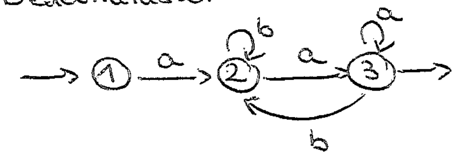


Fig3_ Ex18

Non déterministe:



Déterministe:



Déterministe complet:

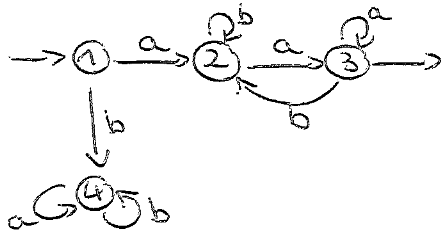


Fig4_ Ex 36_

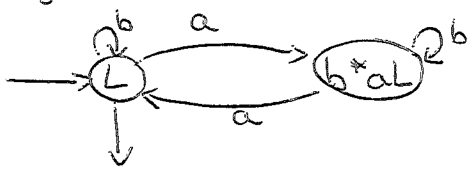


Fig 5_ Ex41_

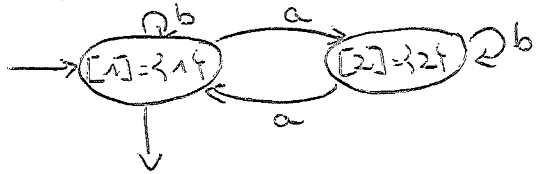
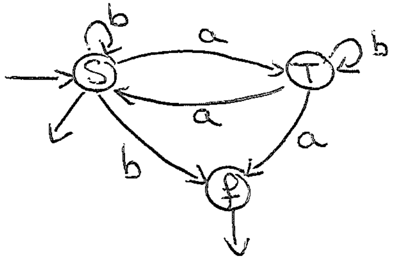


Fig6_ Ex 57_



Référence: Carton:
Langages formels, Calculabilité et Complexité