PDF-XChange Editor Click to BUY NOW! www.pdf-xchange.com

Université de DJIBOUTI **Département d'Informatique** **Année universitaire 2020/2021**

Module : Théorie des Langages

TD1 (Une séance)

Exercice 1 :

Exemples : Soient L_1, L_2 et L_3 trois langages définis par : $L_1 = \{\varepsilon, aa\}$, $L_2 = \{a^i b^j / i, j \geqslant 0\}$ et $L_3 = \{ab, b\}$.
Calculer : $L_1.L_2, L_1.L_3, L_1 \cup L_2, L_2 \cap L_3, L_1^{10}, L_1^*, L_1^+, L_2^R$.

Exercice 2 :

Soient les langages formels suivants :
$L1 = \{a^i b^j,\ i \geq j \geq 1\}$
$L2 = \{a, aa, \varepsilon\}$
$L3 = \{b, ba\}$
$L4 = \{\varepsilon\}$
$L5 = \{\omega \in \{a, b\}^+,\ |\omega|_a = |\omega|_b\}$
$L6 = \{a^i b^i,\ i \geq 1\}$
Trouver les langages : $L2 .L3$, $L2 .L1$, $L1 .L3$, $L5 \cap L1$, $L6 \cup L5$, $L1.(L2 \cap L4)$, $L1 .(L2 \cap L3)$, $(L1 .L2)^R$, $(L1)^R.(L2)^R$.

Exercice 3 :

1. Montrer que la concaténation des langages n'est pas distributive par rapport à l'intersection.
2. Montrer que la concaténation des langages n'est pas idempotente.
3. Montrer que la longueur (notée |...|) est un homomorphisme de monoïdes de $(X^*, ., \varepsilon)$ dans $(N, +, 0)$, X étant un alphabet.

Exercice 4 :

Soient les langages formels suivants :
$L1 = \{a^i b^j / i \geq j \geq 1\}$
$L2 = \{\omega \in \{a, b\}^+\}$
$L3 = \{\omega \in \{a, b\}^+,\ |\omega|_a = |\omega|_b\}$
$L4 = \{a^i b^j a^2 c^k / i > 1,\ j \geq 2,\ k > 1\}$
$L5 = \{\omega \in \{a, b\}^* / |\omega|_a \equiv 1\ [3]\}$
$L6 = \{a^i b^i / i \geq 1\}$
$L7 = \{\omega \in \{a, b\}^* / \exists \alpha, \beta \in \{a, b\}^+ \text{ tq} : \omega = \alpha^n \beta,\ n >= 2\}$
Parmi les mots suivants, préciser quels sont ceux qui appartiennent à quels langages :
ε, a, abba, abbaacc, aba, aabb, abb.

Exercice 1 : On considère les ensembles de mots E1 et E2 définis sur l'alphabet A = {0, 1,2} de la façon suivante :

- E1 est l'ensemble des mots de longueur paire,

- E2 est l'ensemble des mots comportant autant de 0 que de 1 et autant de 1 que de 2.

Définir de façon plus formelle ces deux ensembles et déterminer pour chacun d'eux si la concaténation est une loi interne et si le mot vide en est un élément.

RAPPEL :

Propriété de la concaténation

Soient w, w1 et w2 trois mots définis sur l'alphabet A :

- $|w1.w2| = |w1| + |w2|$;
- $\forall a \in A : |w1.w2|_a = |w1|_a + |w2|_a$;
- (w1.w2).w3 = w1. (w2.w3) (la concaténation est associative);
- w.ε = ε.w = w (ε est un élément neutre pour la concaténation);

Exercice 3 :

Soit la grammaire G = ({a}, {S}, S, {S → aS|ε}).

- Construire la chaîne de dérivation

Définition :

Soit une grammaire G = (V, N, S, R). On dit que le mot u appartenant à V^* est dérivé (ou bien généré) à partir de G s'il existe une suite de dérivation qui, partant de l'axiome S, permet d'obtenir u : $S \xrightarrow{*} u$ (on appelle $S \xrightarrow{*} u$). Le langage de tous les mots générés par la grammaire G est noté L(G). Notez qu'une expression, dérivée à partir de l'axiome, n'est considérée appartenant à L(G) que si elle ne comporte aucun non-terminal.

Exercice 4 :

Soit la grammaire G = ({a,b},{S,T},S,{S → aS|aT,T → bT|b}).

- Quels sont les mots générés par cette Grammaire G.
- Retrouver par la suite le langage généré par cette grammaire.

Si l'arbre syntaxique a comme racine S, alors il est dit arbre de dérivation du mot u tel que u est le mot obtenu en prenant les feuilles de l'arbre dans le sens gauche→droite et bas→haut.

Elle génère le mot aab selon la chaîne de dérivation S → aS → aaT → aab.

- Quel est donc donne l'arbre syntaxique correspondant.

Exercice 5 :

Trouver des grammaires qui génèrent les langages suivants.

- L1 = {ab, a, ba, bb} ;
- L2 = {an/n >0} ;
- L3 = {a2n/n >1}.

1.3.4 Définition du langage généré par une grammaire

Soit une grammaire G=(T,N,S,P), le langage L(G) généré (engendré) par la grammaire G est défini par : $L(G) = \{\omega / S \Rightarrow^* \omega, \omega \in T^*\}$.

Le langage généré par G contient exactement les mots dérivables à partir de l'axiome S par les règles de P et qui ne contiennent que des éléments de T.

1.3.5 Équivalence entre grammaires

Deux grammaires G_1 et G_2 sont équivalentes si et seulement si elles engendrent exactement le même langage.

PDF-XChange Editor Click to BUY NOW! www.pdf-xchange.com

Théorie des Langages

TD n°3 >Grammaires

Exercice1.1

Soit la grammaire formelle G = ({a,b},{S,T},P,S) dont les règles de P sont :

1) S→aS
2) S→bR
3) S→b (4)
4) R→aR
5) R→bS

1. Montrer que le mot abbb est généré par G.
2. Montrer que le mot abb n'est pas généré par G.
3. Montrer qu'un mot se terminant par a ne peut pas être généré par G.
4. Montrer que les mots générés par G ont tous un nombre impair de b.

Exercice1.2

Soit la grammaire formelle G = ({a, b}, {S, T}, P, S) dont les règles de P sont :

1. S→ε
2. S→a
3. S→b
4. T →aSa
5. T →bSb

1) Montrer que le mot abbabba est généré par G. Montrer que le mot baba n'est pas généré par G.
2) Montrer que G engendre l'ensemble L de tous les mots palindromes définis sur l'alphabet Σ= {a, b}.

Un palindrome est un mot identique à lui-même quand on lit ses symboles dans l'ordre inverse.

Exercice1.3

Le but final de l'exercice est de trouver la grammaire du langage L = Σ*\ {ab} (tous les mots possibles sauf ab) définis sur Σ= {a, b}.

1. Donner l'ensemble des mots de L qui sont de longueur inférieure ou égale à 2.
2. Déterminer une grammaire du langage Σ*.
3. En vous aidant de la question 2. ci-dessus, en déduire une grammaire du langage {w∈ Σ∗| |W|> 2}.
4. En vous aidant des questions 1. et 3., déterminer une grammaire du langage L.

Exercice 1.8 Pour les langages sur $X = \{a,b,c\}$ définis ci-dessous, définir une grammaire les engendrant. Vous donnerez le type des différentes grammaires définies.

— $L_1 = \{a^nb^n | n \geq 0\}$.
— $L_2 = \{a^nb^pc^q | n,p,q \geq 1\}$.
— $L_3 = \{a^nb^p | n \geq p \geq 0\}$.
— $L_4 = \{a^nb^m | n \neq m, n,m \geq 0\}$.
— $L_5 = \{a^nb^pc^nd^q | n,p,q > 0\} \cup \{a^pb^nc^qd^n | n,p,q > 0\}$.
— $L_6 = \{a^nb^pc^q | n,q \geq 0, p \geq n+q\}$.
— $L_7 = \{a^nb^p | n,p \geq 0, n \neq p+3\}$.
— $L_8 = \{a^nb^nc^p | n,p \geq 1\}$.
— $L_9 = \{a^nb^nc^n | n \geq 1\}$.

Type 3 (grammaire régulière ou grammaire linéaire) :

- ☐ à droite. Si toutes les règles de P sont de la forme :
  $A \to \alpha B | \alpha$ avec $A, B \in V_N$ et $\alpha \in V_T^*$
- ☐ à gauche. Si toutes les règles de P sont de la forme :
  $A \to B\alpha | \alpha$ avec $A, B \in V_N$ et $\alpha \in V_T^*$

Type 2 (grammaire à contexte libre, hors contexte, grammaire algébrique ou grammaire de Chomsky) : Si toutes les règles de P sont de la forme :

$$A \to \alpha \text{ avec } A \in V_N \text{ et } \alpha \in (V_T \cup V_N)^*$$

Type 1 (grammaire à contexte lié ou contextuelle) : Si toutes les règles sont de la forme :

$$\alpha A \beta \to \alpha w \beta \text{ avec } \alpha, \beta \in (V_T \cup V_N)^*, A \in V_N \text{ et } w \in (V_T \cup V_N)^*$$

Autrement dit, le symbole non terminal A est remplacé par w si on a les contextes α à gauche et β à droite. De plus, seul l'axiome peut générer le mot vide ($(S \to \varepsilon) \in P$) et dans ce cas S n'apparaît pas en partie droite d'une autre règle.

Autre définition des grammaires de type 1 :
Toutes les règles sont de la forme $\alpha \to \beta$ tels que $\alpha \in (V_T \cup V_N)^+$, $\beta \in (V_T \cup V_N)^*$ et $|\alpha| \leq |\beta|$. De plus, si ε apparait dans une règle en partie droite alors on a S (l'axiome) en partie gauche, c'est-à-dire que $(S \to \varepsilon) \in P$, les autres règles où apparait ε n'étant pas autorisées .

Type 0 (grammaire sans restriction) : pas de restriction su les règles :

$$\alpha \to \beta \text{ avec } \alpha \in (V_T \cup V_N)^+ \text{ et } \beta \in (V_T \cup V_N)^*$$

Remarque :

- Il existe une relation d'inclusion entre ces quatre types de grammaires. Autrement dit, on a :
  type 3 ⊆ type 2 ⊆ type 1 ⊆ type 0.

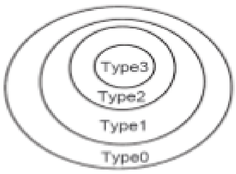

Fig. – hiérarchie de Chomsky

- Le type retenu pour une grammaire est le plus petit qui satisfait les conditions.

I.3. Classification des langages
La classification des grammaires ci-dessus va permettre de classer les langages selon le type de grammaires nécessaire à leur génération. Un langage pouvant être généré par une grammaire de type i et pas par une grammaire d'un type supérieur dans la hiérarchie, sera appelé un langage de type i.

A chaque type de grammaire est associé un type de langage:

Grammaire	Langage
Type 3 ou régulière	Régulier ou rationnel
Type 2 ou hors-contexte	Algébrique ou contexte libre
Type 1 ou contextuelle	Contextuel ou à contexte lié
Type 0	Récursivement énumérable ou décidable

Un langage peut être généré par différentes grammaires qui peuvent être de type différent. Un langage prend le plus petit type au sens de l'inclusion.

Le type retenu pour une grammaire est le plus petit qui satisfait les conditions. Pour trouver la classe d'un langage on procède cependant comme suit :

- Chercher une grammaire de type 3 qui le génère, si elle existe, le langage est de type 3 (ou régulier)
- Sinon, chercher une grammaire de type 2 qui le génère, si elle existe, le langage est de type 2 (ou algébrique)
- Sinon, chercher une grammaire de type 1 qui le génère, si elle existe, le langage est de type 1 (ou contextuel)
- Sinon, le langage est de type 0.