Chapitre 1 : Introduction à la Théorie des Langages

Dr. Mandicou BA

mandicou.ba@esp.sn

http://www.mandicouba.net

Diplôme D'Ingénieur de Conception (DIC, 2^{e} année)

Option Informatique

ECOLE SUPERIEURE POLYTECHNIQUE

www.esp.sn

Plan du Chapitre

Sommaire

Symboles, alphabets et mots

Définition 1 (Symboles)

- Les *symboles* sont des éléments indivisibles qui vont servir de briques de base pour la construction d'un mot

Exemple 1

☛ *Caractère de l'alphabet ou Chiffre de 0 à 9 est un symbole.*

Définition 2 (Alphabet)

- *Un alphabet est un ensemble fini et non vide de symboles.*
- Conventionnellement, un alphabet est noté $\sum$

Exemple 2

1. $\sum = \{0, 1\}$ *: l'alphabet binaire*
2. $\sum = \{a, b, \cdots, z\}$ *: l'ensemble des caractères minuscules*
3. *L'ensemble des caractères ASCII*

Symboles, alphabets et mots

Définition 3 (Mot)

- Un mot *m* sur $\sum$ (encore appelé chaîne) est une suite de symboles, appartement à l'alphabet $\sum$, mis bout à bout

☛ 01101 est un mot de l'alphabet binaire $\sum = \{0, 1\}$

Longueur d'un mot

- La longueur d'un mot m, noté $|m|$, représente le nombre de symboles qui le constituent

☛ Le mot 01101 est de longueur 5.

Nombre de symboles dans un mot

- Le nombre de symbole *s* que possède un mot *m* est noté $|m|_s$.

☛ $|01101|_1 = 3$.

Symboles, alphabets et mots

Mot vide

- Le mot vide d'un alphabet $\sum$, noté ε, est le mot de longueur zéro.

i^{ieme} symbole d'un mot m

- Soit m est un mot et i tel que $1 \leq i \leq |m|$.
- Le i^{ieme} symbole de m est le symbole situé à la position i et est noté $m[i]$.

Puissance d'un alphabet

- $\forall \sum$, $\sum^k$ désigne l'ensemble de tous les mots de longueur k.

Exemple 3 (Soit l'alphabet binaire $\sum = \{0, 1\}$)

1. $\sum^1 = \{0, 1\}; \sum^2 = \{00, 01, 10, 11\}$
2. $\sum^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$
3. *NB : quel que soit l'alphabet $\sum$, alors on obtient $\sum^0 = \{\varepsilon\}$*

Symboles, alphabets et mots

Ensemble de tous les mots d'un alphabet

- On note $\sum^*$ l'ensemble de tous les mots d'un alphabet $\sum$.

Exemple 4 (pour l'alphabet $\sum = \{0, 1\}$)

- $\sum^* = \{0, 1\}^* = \{\varepsilon, 0, 00, 10, 110, 000, \cdots\}$
- *Autrement dit :* $\sum^* = \sum^0 \cup \sum^1 \cup \sum^2 \cup \sum^3 \cup \cdots$

Ensemble de tous les mots non vides d'un alphabet

- On note $\sum^+$ l'ensemble de tous les mots non vides de l' alphabet $\sum$.

Conséquence 1

- $\sum^+ = \sum^1 \cup \sum^2 \cup \sum^3 \cup \cdots$
- $\sum^* = \sum^+ \cup \{\varepsilon\}$

Concaténation de mots

Définition

- Soient α et β deux mots définis dans $\sum$. La concaténation de α et β, notée $\alpha \cdot \beta$ (ou simplement $\alpha\beta$), consiste à juxtaposer les symboles du mot β à la suite de ceux du mot α.

Exemple 5

☛ *Soient* $\alpha = a_1 a_2 \cdots a_i$ *et* $\beta = b_1 b_2 \cdots b_j$ *deux mots de* $\sum$

☛ $\alpha\beta = a_1 a_2 \cdots a_i b_1 b_2 \cdots b_j$

Propriété 1 (Propriétés de la concaténation)

- *La concaténation est associative :* $(\alpha\beta)\gamma = \alpha(\beta\gamma)$
- ε *est l'élément neutre pour la concaténation :* $\varepsilon\alpha = \alpha\varepsilon = \alpha$
- *L'itération de la concaténation d'un mot m donne les puissance de m :*
    - $m^0 = \varepsilon$
    - $\forall n \in \mathbb{N}, m^{n+1} = mm^n$

Facteurs de mots

- Soient α, β et γ trois mots définis sur un alphabet $\sum$.
  - β est un facteur (ou sous-chaîne) du mot $\alpha\beta\gamma$
  - α est un facteur gauche (ou préfixe) du mot $\alpha\beta$
  - β est un facteur droit (ou suffixe) du mot $\alpha\beta$
  - Si $\alpha \neq \beta$ et α est un préfixe de β, alors α est dit *préfixe propre* de β
  - Si $\alpha \neq \beta$ et α est un suffixe de β, alors α est dit *suffixe propre* de β
  - ε est suffixe, préfixe et facteur de tout mot *m* sur tout alphabet $\sum$

Sommaire

Définition d'un langage

Définition 4 (Langage $\mathcal{L}$ sur un alphabet $\sum$)

- Un ensemble de mots **construits sur** l'alphabet $\sum$

Conséquence 2

- *Si $\sum$ est un alphabet et $\mathcal{L} \subseteq \sum^*$, alors $\mathcal{L}$ est un langage sur $\sum$*

  Autrement dit, tout langage $\mathcal{L}$ défini sur $\sum$ est une partie de $\sum^*$

- *L'ensemble de tous les langages qui peuvent être définis sur $\sum^*$ est dit l'ensemble des parties de $\sum^*$ et est noté $\mathcal{P}(\sum^*)$*

Quelques langages particuliers

- $\sum^*$ est un langage pour tout alphabet.
- $\emptyset$, le langage vide, est un langage pour tout alphabet
- $\{\varepsilon\}$, le langage constitué du mot vide, est un langage pour tout alphabet
- NB : $\emptyset \neq \{\varepsilon\}$

Opérations sur les langages

Soient $\mathcal{L}_1$ et $\mathcal{L}_1$ deux langages définis sur un alphabet $\sum$

- Union de deux langages : $\mathcal{L}_1 \cup \mathcal{L}_2 \equiv \{x | x \in \mathcal{L}_1 \vee x \in \mathcal{L}_2\}$
- Intersection de deux langages : $\mathcal{L}_1 \cap \mathcal{L}_2 \equiv \{x | x \in \mathcal{L}_1 \wedge x \in \mathcal{L}_2\}$
- Différence de deux langages : $\mathcal{L}_1 - \mathcal{L}_2 \equiv \{x | x \in \mathcal{L}_1 \wedge x \notin \mathcal{L}_2\}$
- Complément d'un langage : $\overline{\mathcal{L}} \equiv \{x \in \sum^* | x \notin \mathcal{L}\}$
- Concaténation de deux langages : $\mathcal{L}_1 \mathcal{L}_2 \equiv \{xy | x \in \mathcal{L}_1 \wedge y \in \mathcal{L}_2\}$
- Auto concaténation d'un langage : $\overbrace{\mathcal{L} \cdots \mathcal{L}}^{n} \equiv \mathcal{L}^n$
- Fermeture de Kleene : $\mathcal{L}^* \equiv \bigcup_{k \geq 0} \mathcal{L}^k$

Exemples langages

Exemple 6 (Soit $\sum = \{0, 1\}$ l'alphabet binaire)

- *Le langage de tous les mots constitués de* n 0 *suivit de* n 1*, avec* $n \geq 0$
  - $\mathcal{L}_1 = \{\varepsilon, 01, 001, 00011, 00111, \cdots\}$
- *Le langage de tous les mots constitués d'un nombre égal de* 0 *et* 1
  - $\mathcal{L}_2 = \{\varepsilon, 01, 0011, 000111, 1001, \cdots\}$
- *Le langage de tous les mots binaires qui sont des nombres premiers*
  - $\mathcal{L}_3 = \{\varepsilon, 01, 11, 101, 1011, \cdots\}$

Exemples langages

Exemple 7 (Le langage français)

- $\sum$ *est l'ensemble de tous les mots du français*
- *Une phrase du français peut être représentée par un mot* $m \in \sum *$
- $\mathcal{L}$ *est l'ensemble des phrases syntaxiquement correctes du français*
  1. *Ceci est une phrase correcte* $\in \mathcal{L}$
  2. *Ceci est une correcte phrase* $\notin \mathcal{L}$
  3. *Le cœur a ses raisons que la raison ne connaît point* $\in \mathcal{L}$

Exemples langages

Exemple 8 (Le langage arithmétique)

- $\sum = \{0 \cdots 9, +, \times, /, -, (,)\}$
- *Une expression arithmétique peut être représenté e par un mot m* $\in \sum *$
- $\mathcal{L}$ *est l'ensemble des expressions arithmétiques bien formées*
  1. $(1+3) \times 45 \in \mathcal{L}$
  2. $/43 \notin \mathcal{L}$
  3. $(1+234 \notin \mathcal{L}$

Exemples langages

Exemple 9 (Un exemple de langage logique)

- $\sum = \{a, \cdots z, \vee, \wedge, \urcorner, \rightarrow, \longleftrightarrow\}$
- *Une formule de la logique des propositions peut être représentée par un mot* $m \in \sum *$
- $\mathcal{L}$ *est l'ensemble des formules de la logique des propositions bien formées*
  1. $a \vee b \wedge c \in \mathcal{L}$
  2. $\vee bc \notin \mathcal{L}$

Exemples langages

Exemple 10 (Le langage de programmation C)

- $\sum$ *est l'ensemble des identifiants, des constantes, des opérateurs et des mots clefs du langage C*
- *Un programme en langage C peut être représenté par un mot* $m \in \sum *$
- $\mathcal{L}$ *est l'ensemble des programmes syntaxiquement corrects du langage de programmation C*
  1. `main(void){printf("Hello word");}` $\in \mathcal{L}$
  2. `main(void){printf("Hello word)}` $\notin \mathcal{L}$

Problème de décision en théorie des langages

Problème de décision

- **Entrée** : $x \in \mathcal{S}$
- **Question** : x satisfait-il à la propriété $\mathcal{P}$ de $\mathcal{S}$?

- $\mathcal{S}$ étant un ensemble quelconque, dont les éléments sont appelés instances ou entrée du problème.
- $\mathcal{P}$ est une propriété de $\mathcal{S}$
- Représentons $x \in S$ par un mot m
- Appelons $\mathcal{L}$ l'ensemble des mots qui représentent les instances qui vérifient $\mathcal{P}$.

Problème de décision en théorie des langages

- **Entrée** : un mot m
- **Question** : m appartient-il à $\mathcal{L}$?

Description d'un langage

Comment décrire un langage

- **Description littérale**
  - Ensemble de tous les mots construits l'alphabet $\sum = \{a, b\}$, de longueur paire
- **Énumération**
  - $\mathcal{L} = \{\varepsilon, aa, bb, ab, ba, aaaa, aaab, aaba, \cdots\}$
- **Expressions régulières**
  - Formule permettant de dénoter de manière concise tous les mots du langage : $((a+b)(a+b))*$.
- **Reconnaisseur**
  - Machine permettant de reconnaitre tous les mots du langage
- **Grammaire de réécriture**
  - Système permettant de générer tous les mots du langage

Sommaire

Composition d'un reconnaisseur

1. Bande de lecture
2. Tête de lecture
3. Mémoire
4. Unité de contrôle

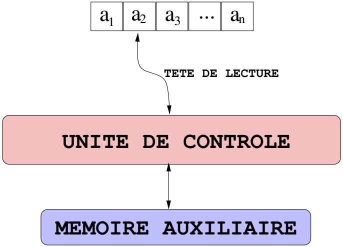


Figure: Illustration schématique d'un reconnaisseur

Les éléments d'un reconnaisseur

Bande de lecture

1. Composée d'une succession de cases
2. Chaque ne peut contenir qu'un seul symbole d'un alphabet d'entrée
3. Les symboles du mot à reconnaître sont écris sur les cases de la bande de lecture

Tête de lecture

1. Elle ne peut lire qu'une seule case à un instant donné
2. La case sur laquelle se trouve la tête de lecture est appelée case courante
3. La tête peut être déplacée par le reconnaisseurs pour se positionner sur la case située immédiatement à gauche ou à droite de la case courante
4. Un reconnaisseur qui ne peut déplacer la tête de lecture que de la gauche vers la droite est appelé reconnaisseur à sens unique

Les éléments d'un reconnaisseur

Mémoire auxiliaire

1. La mémoire peut prendre des formes différentes
2. On suppose un alphabet de mémoire finie et que la mémoire contient uniquement des symboles de cet alphabet
3. On suppose qu'à un instant donné, il est possible de déterminer avec exactitude le contenu et la structure de la mémoire
4. On suppose également que la taille de la mémoire peut varier aléatoirement au cours du temps
5. Deux fonctions principales :
   - fonction de correspondance
   - fonction de transition

Les éléments d’un reconnaisseur

Unité de contrôle

1. Elle est le cœur du reconnaisseurs : représente le programme qui dicte au reconnaisseur son comportement
2. Elle est définie par :
   1. un ensemble d’états finis
   2. associé à une fonction de transition qui décrit le passage d’un état vers un autre en fonction :
      1. du contenu de la case courante de la bande de lecture
      2. et du contenu de la mémoire
3. Elle décide :
   1. du sens de déplacement de la tête de lecture
   2. et des symboles à stocker dans la mémoire

Fonctionnement d'un reconnaisseur

Principe de fonctionnement

- Un reconnaisseur opère par des séquences de déplacement
- Au début de chaque déplacement, la case courante et l'état de la mémoire sont évalué par la fonction de transition
- Ainsi, le symbole courant, l'information issue de la mémoire et l'état de l'unité de contrôle détermine le sens de déplacement . Ce déplacement consiste :
  1. Déplacement de la tête de lecture vers la case de gauche, ou vers la case de droite, ou immobilisation sur la case courante
  2. Stockage des informations sur la mémoire
  3. Changement de l'état de l'unité de contrôle

Fonctionnement d'un reconnaisseur

Configurations d'un reconnaisseur

- Le comportement d'un reconnaisseur peut être, *conventionnellement*, décrit en termes de configuration
- Une configuration est une « cartographie » du reconnaisseur décrivant :
  1. L'état de l'unité de contrôle
  2. Le contenu de la bande de lecture
  3. La position de la tête de lecture
  4. Le contenu de la mémoire

Unité de Contrôle de Déterministe vs Unité de Contrôle de Non Déterministe

- L'unité de contrôle est dit déterministe si et seulement si à chaque configuration, un et un seul déplacement est possible.
- L'unité de contrôle est dit non déterministe si à chaque configuration, plusieurs déplacements sont possibles.

Configuration initiale et configuration finale

Configuration initiale

- La configuration initiale d'un reconnaisseur est une configuration où :
  1. L'unité de contrôle se trouve dans son état initial
  2. La tête de lecture est positionnée sur la case la plus à gauche de la bande de lecture
  3. La mémoire a le contenu initial spécifié

Configuration finale

- La configuration finale d'un reconnaisseur est une configuration où :
  1. L'unité de contrôle se trouve un état d'acception spécifié
  2. La tête de lecture est positionnée sur la case la plus à droite de la bande de lecture
  3. La mémoire satisfait les conditions de la configuration finale

Condition d'acceptation d'un reconnaisseur

Acceptation

- ☛ Un reconnaisseur accepte un mot *m* si et seulement si, en partant d'une configuration initiale où *m* est placé sur la bande de lecture, il effectue une séquence de déplacements successifs qui termine sur une configuration finale.

Cas d'un reconnaisseur non déterministe

- ☛ Si au moins une des séquences de déplacements possibles conduit à une configuration finale, alors la mot passé en entrée va être accepté

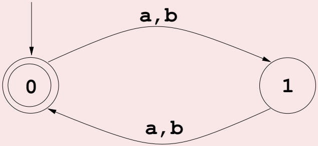
Langage défini par un reconnaisseur

- ☛ Un langage défini par un reconnaisseur représente l'ensemble des mots acceptés par ce reconnaisseur

Exemple de reconnaisseur simple

Exemple 11

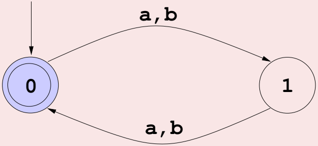
- *Soit le langage $\mathcal{L}$ défini par un reconnaisseur ci-dessous*
- *Est-ce que le mot ababaa $\in \mathcal{L}$?*



Exemple de reconnaisseur simple

Exemple 11

- *Soit le langage $\mathcal{L}$ défini par un reconnaisseur ci-dessous*
- *Est-ce que le mot ababaa $\in \mathcal{L}$?*

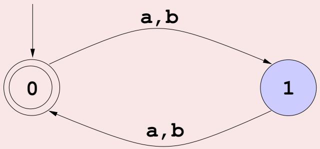


(0,ababaa)

Exemple de reconnaisseur simple

Exemple 11

- *Soit le langage $\mathcal{L}$ défini par un reconnaisseur ci-dessous*
- *Est-ce que le mot ababaa $\in \mathcal{L}$?*



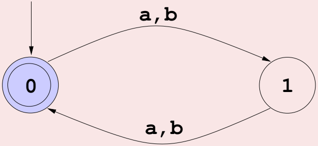
(0,ababaa)

$\vdash$ (1,babaa)

Exemple de reconnaisseur simple

Exemple 11

- *Soit le langage $\mathcal{L}$ défini par un reconnaisseur ci-dessous*
- *Est-ce que le mot ababaa $\in \mathcal{L}$?*

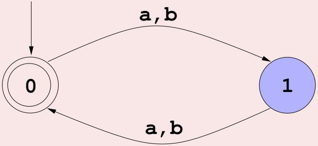


$$
\begin{aligned}
& (0, \texttt{ababaa}) \\
\vdash\ & (1, \texttt{babaa}) \\
\vdash\ & (0, \texttt{abaa})
\end{aligned}
$$

Exemple de reconnaisseur simple

Exemple 11

- *Soit le langage $\mathcal{L}$ défini par un reconnaisseur ci-dessous*
- *Est-ce que le mot ababaa $\in \mathcal{L}$?*

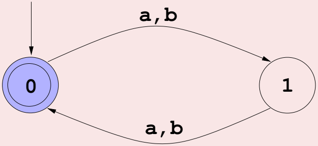


$$
\begin{aligned}
& (0,\mathtt{ababaa}) \\
\vdash\ & (1,\mathtt{babaa}) \\
\vdash\ & (0,\mathtt{abaa}) \\
\vdash\ & (1,\mathtt{baa})
\end{aligned}
$$

Exemple de reconnaisseur simple

Exemple 11

- *Soit le langage $\mathcal{L}$ défini par un reconnaisseur ci-dessous*
- *Est-ce que le mot ababaa $\in \mathcal{L}$?*

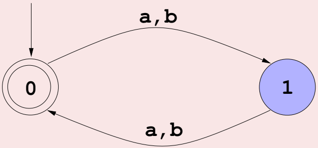


$$
\begin{aligned}
& (0, \texttt{ababaa}) \\
\vdash\ & (1, \texttt{babaa}) \\
\vdash\ & (0, \texttt{abaa}) \\
\vdash\ & (1, \texttt{baa}) \\
\vdash\ & (0, \texttt{aa})
\end{aligned}
$$

Exemple de reconnaisseur simple

Exemple 11

- *Soit le langage $\mathcal{L}$ défini par un reconnaisseur ci-dessous*
- *Est-ce que le mot ababaa $\in \mathcal{L}$?*

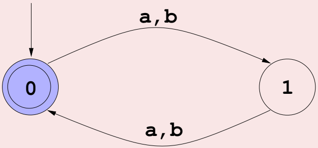


$$
\begin{aligned}
& \mathtt{(0,ababaa)} \\
\vdash\; & \mathtt{(1,babaa)} \\
\vdash\; & \mathtt{(0,abaa)} \\
\vdash\; & \mathtt{(1,baa)} \\
\vdash\; & \mathtt{(0,aa)} \\
\vdash\; & \mathtt{(1,a)}
\end{aligned}
$$

Exemple de reconnaisseur simple

Exemple 11

- *Soit le langage $\mathcal{L}$ défini par un reconnaisseur ci-dessous*
- *Est-ce que le mot ababaa $\in \mathcal{L}$?*



(0,ababaa)

$\vdash$ **(1,babaa)**

$\vdash$ **(0,abaa)**

$\vdash$ **(1,baa)**

$\vdash$ **(0,aa)**

$\vdash$ **(1,a)**

$\vdash$ **(0,ε)**

Sommaire

Description générale d'une grammaire

- Les grammaires sont, probablement, la plus importante classe de générateur de langage
- Une grammaire est un système mathématique permettant de générer tous les mots d'un langage
- Pour un langage $\mathcal{L}$, une grammaire se fonde sur deux ensembles disjoints de symboles
  1. L'ensemble des symboles non-terminaux, noté $\mathcal{N}$ et appelé alphabet non-terminal
     - *Les non-terminaux représentent les symboles élémentaires du langage défini par la grammaire*
  2. L'ensemble des symboles terminaux, noté $\sum$ et appelé alphabet terminal
     - *Les terminaux représentent des mots constitués de non-terminaux et utilisés dans la génération des mots du langages*

Description générale d'une grammaire

Règles de production

- Le cœur d'une grammaire est un ensemble fini, noté $\mathcal{P}$, de règles de production (encore appelé *production*)
- Les règles de production décrivent comment une phrase du langage définie par une grammaire va être générée
- Une règle de production (α, β) (ou règle de réécriture) un élément de l'ensemble
  - $(\mathcal{N} \cup \Sigma)^* \mathcal{N} (\mathcal{N} \cup \Sigma)^* \times (\mathcal{N} \cup \Sigma)^*$
    1. $(\mathcal{N} \cap \Sigma)^* \mathcal{N} (\mathcal{N} \cup \Sigma)^*$: désigne n'importe quelle chaîne contenant au moins un non-terminal
    2. $(\mathcal{N} \cup \Sigma)^*$: désigne n'importe quelle chaîne

Convention

- Si (α, β) est une règle de production, alors la notation suivante est utilisée : $\alpha \to \beta$

Description formelle d'une grammaire

Définition 5

- Une grammaire de réécriture est un 4-uplet $\langle \mathcal{N}, \Sigma, \mathcal{P}, \mathcal{S} \rangle$ où :
  - $\mathcal{N}$ est un ensemble fini de symboles non-terminaux, appelé l'alphabet terminal
  - Σ est un ensemble de symboles terminaux, appelé alphabet terminal, tel que $\mathcal{N} \cap \Sigma = \emptyset$
  - $\mathcal{P}$ est un sous ensemble fini de :

    $$(\mathcal{N} \cup \Sigma)^* \mathcal{N} (\mathcal{N} \cup \Sigma)^* \times (\mathcal{N} \cup \Sigma)^*$$

    Un élément de (α, β) $\mathcal{P}$, aussi noté $\alpha \rightarrow \beta$, est une règle de production
- $\mathcal{S}$ est un élément de $\mathcal{N}$ appelé l'axiome de la grammaire

Exemples de grammaire de réécriture

Exemple 12

- *Soit le langage $\mathcal{L}$ défini par la grammaire suivante*
- $\mathcal{G} = \langle \{P, I\}, \{a, b\}, \{P \to \varepsilon, P \to aI, P \to bI, I \to aP, I \to bP\}, P \rangle$
- *Le mot ababaa* $\in \mathcal{L}$ *?*

Réponse

$$
\begin{aligned}
P &\Rightarrow aI \\
&\Rightarrow abP \\
&\Rightarrow abaI \\
&\Rightarrow ababP \\
&\Rightarrow ababaI \\
&\Rightarrow ababaaP \\
&\Rightarrow ababaa
\end{aligned}
$$

Exemples de grammaire de réécriture

Exemple 12

- *Soit le langage $\mathcal{L}$ défini par la grammaire suivante*
- $\mathcal{G} = \langle \{P, I\}, \{a, b\}, \{P \to \varepsilon, P \to aI, P \to bI, I \to aP, I \to bP\}, P \rangle$
- *Le mot ababaa $\in \mathcal{L}$?*

Réponse

$$\begin{aligned} P &\Rightarrow aI \\ &\Rightarrow abP \\ &\Rightarrow abaI \\ &\Rightarrow ababP \\ &\Rightarrow ababaI \\ &\Rightarrow ababaaP \\ &\Rightarrow ababaa \end{aligned}$$

Exemples de grammaire de réécriture

Exemple 12

- *Soit le langage $\mathcal{L}$ défini par la grammaire suivante*
- $\mathcal{G} = \langle \{P, I\}, \{a, b\}, \{P \to \varepsilon, P \to aI, P \to bI, I \to aP, I \to bP\}, P \rangle$
- *Le mot ababaa $\in \mathcal{L}$?*

Réponse

$$
\begin{aligned}
P &\Rightarrow aI \\
&\Rightarrow abP \\
&\Rightarrow abaI \\
&\Rightarrow ababP \\
&\Rightarrow ababaI \\
&\Rightarrow ababaaP \\
&\Rightarrow ababaa
\end{aligned}
$$

Exemples de grammaire de réécriture

Exemple 12

- *Soit le langage $\mathcal{L}$ défini par la grammaire suivante*
- $\mathcal{G} = \langle \{P, I\}, \{a, b\}, \{P \to \varepsilon, P \to aI, P \to bI, I \to aP, I \to bP\}, P \rangle$
- *Le mot ababaa* $\in \mathcal{L}$ *?*

Réponse

$$
\begin{aligned}
P &\Rightarrow aI \\
&\Rightarrow abP \\
&\Rightarrow abaI \\
&\Rightarrow ababP \\
&\Rightarrow ababaI \\
&\Rightarrow ababaaP \\
&\Rightarrow ababaa
\end{aligned}
$$

Exemples de grammaire de réécriture

Exemple 12

- *Soit le langage $\mathcal{L}$ défini par la grammaire suivante*
- $\mathcal{G} = \langle \{P, I\}, \{a, b\}, \{P \rightarrow \varepsilon, P \rightarrow aI, P \rightarrow bI, I \rightarrow aP, I \rightarrow bP\}, P \rangle$
- *Le mot ababaa* $\in \mathcal{L}$ *?*

Réponse

$$
\begin{aligned}
P & \Rightarrow aI \\
& \Rightarrow abP \\
& \Rightarrow abaI \\
& \Rightarrow ababP \\
& \Rightarrow ababaI \\
& \Rightarrow ababaaP \\
& \Rightarrow ababaa
\end{aligned}
$$

Exemples de grammaire de réécriture

Exemple 12

- *Soit le langage $\mathcal{L}$ défini par la grammaire suivante*
- $\mathcal{G} = \langle \{P, I\}, \{a, b\}, \{P \rightarrow \varepsilon, P \rightarrow aI, P \rightarrow bI, I \rightarrow aP, I \rightarrow bP\}, P \rangle$
- *Le mot ababaa $\in \mathcal{L}$?*

Réponse

$$
\begin{aligned}
P & \Rightarrow aI \\
& \Rightarrow abP \\
& \Rightarrow abaI \\
& \Rightarrow ababP \\
& \Rightarrow ababaI \\
& \Rightarrow ababaaP \\
& \Rightarrow ababaa
\end{aligned}
$$

Exemples de grammaire de réécriture

Exemple 12

- *Soit le langage $\mathcal{L}$ défini par la grammaire suivante*
- $\mathcal{G} = \langle \{P, I\}, \{a, b\}, \{P \rightarrow \varepsilon, P \rightarrow aI, P \rightarrow bI, I \rightarrow aP, I \rightarrow bP\}, P \rangle$
- *Le mot ababaa $\in \mathcal{L}$?*

Réponse

$$
\begin{aligned}
P &\Rightarrow aI \\
&\Rightarrow abP \\
&\Rightarrow abaI \\
&\Rightarrow ababP \\
&\Rightarrow ababaI \\
&\Rightarrow ababaaP \\
&\Rightarrow ababaa
\end{aligned}
$$

Exemples de grammaire de réécriture

Exemple 12

- *Soit le langage $\mathcal{L}$ défini par la grammaire suivante*
- $\mathcal{G} = \langle \{P, I\}, \{a, b\}, \{P \rightarrow \varepsilon, P \rightarrow aI, P \rightarrow bI, I \rightarrow aP, I \rightarrow bP\}, P \rangle$
- *Le mot ababaa* $\in \mathcal{L}$ *?*

Réponse

$$\begin{aligned} P &\Rightarrow aI \\ &\Rightarrow abP \\ &\Rightarrow abaI \\ &\Rightarrow ababP \\ &\Rightarrow ababaI \\ &\Rightarrow ababaaP \\ &\Rightarrow ababaa \end{aligned}$$

Chapitre 1 : Introduction à la Théorie des Langages

Dr. Mandicou BA
mandicou.ba@esp.sn
http://www.mandicouba.net

Diplôme D'Ingénieur de Conception (DIC, 2^{e} année)
Option Informatique

ECOLE SUPERIEURE POLYTECHNIQUE
www.esp.sn