Chapitre 1

Rappels sur les langages formels

Ce chapitre constitue un rappel de certaines notions fondamentales issues de la théorie des langages. L'objectif est d'habituer le lecteur aux concepts utilisés et fixer certaines notations que l'on retrouvera tout au long de cet ouvrage. Ces notions ne prétendent aucunement remplacer un cours de théorie des langages. De ce fait, aucune démonstration théorique n'est rapportée. Néanmoins, ces notions ont le mérite de résumer l'essentiel des connaissances permettant d'aborder avec sérénité les techniques de compilation qui feront l'objet de plusieurs autres chapitres de cet ouvrage.

1 Définitions préliminaires

Définition 1.1 (Vocabulaire)

On appelle vocabulaire (ou alphabet) un ensemble fini non vide de symboles ou de caractères.

Par exemple :

- l'ensemble B = {0, 1} représente l'alphabet de base des nombres binaires dans le système de numération binaire pur.
- D = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} est le vocabulaire de base pour former des séquences de chiffres représentant des entiers naturels.
- V = {a, ..., z, A,..., Z} est l'alphabet permettant de former des mots des langues naturelles comme le Français, l'Anglais, l'Espagnol, etc.
- R = {I, V, X, L, C, D, M} est le vocabulaire de base du système de numération utilisé par les Romains de l'Antiquité.
- P = V $\cup$ D est l'alphabet de base des identificateurs dans un langage comme Fortran, Pascal, etc.

Définition 1.2 (Chaîne)

Une chaîne (ou mot) sur un alphabet A est une séquence finie éventuellement vide d'éléments de A.

Par exemple, les mots : "maison", "jardin", "classe", "ruelle", "1200$", "188E–2", représentent des chaînes de caractères.

Définition 1.3 (Longueur d'une chaîne)

La longueur d'une chaîne x définie sur un alphabet A est le nombre de symboles qui composent x. Elle est notée habituellement par $|x|$.

Par exemple, les longueurs des mots "maison" et "1788E–2" sont notées respectivement par $|\text{maison}| = 6$ et $|1788\text{E}–2| = 7$.

Définition 1.4 (Chaîne vide)

On appelle chaîne vide, une séquence de longueur nulle.

On la note habituellement (en théorie des langages) par le symbole ε (epsilon). La longueur de ε est évidemment égale à zéro ($|\varepsilon| = 0$).

On note V^+, l'ensemble de toutes les chaînes construites à partir des éléments d'un alphabet V donné. On écrit alors $V^+ = V^1 \cup V^2 \ ... \cup V^n \ ... = \bigcup_{i\geq 1} V^i$. La puissance « i » représente la longueur des mots de l'ensemble V^i. De même, on désigne par V^*, l'ensemble de toutes les chaînes de V^+, auquel on ajoute la chaîne vide, c'est-à-dire que $V^* = \bigcup_{i\geq 1} V^i \cup \{\varepsilon\} = \bigcup_{i\geq 0} V^i = V^0 \cup V^1 \cup V^2 \ ... \cup V^n \ ...$

On a alors les équations $V^* = V^+ \cup \{\varepsilon\}$ et $V^+ = V^* - \{\varepsilon\}$.

Par exemple :

- Soit $V_1 = \{a\}$ un ensemble contenant un seul caractère. Alors, l'ensemble de toutes les chaines de longueur quelconque formées à base du vocabulaire V_1, est représenté par l'ensemble $V_1^* = V_1^0 \cup V_1^1 \ ... \cup V_1^n ... = \{a\}^0 \cup \{a\}^1 \ ... \cup \{a\}^n ... = \{\varepsilon, a, aa, aaa, aaaa, ..., a^n, ...\} = \{a^n \mid n \geq 0\}$. V_1^* est dit infini dénombrable (ou infini énumérable), c'est-à-dire que ses éléments peuvent être listés sans omission ni répétition dans une liste indexée par des entiers. De manière formelle, un ensemble E est dit dénombrable quand il est équipotent à l'ensemble des entiers naturels N, c'est-à-dire qu'il existe une bijection de N sur l'ensemble E.

- Soit l'alphabet $V_2 = \{a, b\}$. L'ensemble V_2^* contient toutes les chaines constituées des lettres *a* et *b*, y compris la chaine vide ε. Cet ensemble est également infini dénombrable. Un aperçu sur la liste de ses éléments est : ε, *a*, *b*, *aa*, *ab*, *ba*, *bb*, *aaa*, *aab*, *aba* ...= $\{a, b\}^n$, avec $n \geq 0$.

Définition 1.5 (Concaténation)

Etant données deux chaînes v *et* w *éléments de l'ensemble* V^*. *On appelle concaténation des chaines* v *et* w, *la juxtaposition de* v *et* w. *On note habituellement cette opération par* vw *ou* v.w.

L'ensemble V^* est le monoïde libre engendré par l'alphabet V. En effet, l'opération de concaténation étant la loi de composition interne sur V^*. Par conséquent, quels que soient les éléments (chaînes) v et $w \in V^*$, leur juxtaposition donne lieu toujours à un élément (chaîne) interne à l'ensemble V^*, c'est-à-dire que $z = v.w \in V^*$. Cette opération est associative, car quelles que soient les chaines (u, v et w) $\in V^*$, on a toujours $(u.v).w = u.(v.w) = u.v.w$. En outre, l'élément neutre pour le monoïde V^* est la chaîne vide ε , puisque quelle que soit la chaine $v \in V^*$, on vérifie toujours la relation $v\varepsilon = \varepsilon v = v$.

La concaténation, comme on peut facilement remarquer, n'est pas commutative. En effet, en choisissant le contre-exemple $u = ab$ et $w = ba$, on vérifie aisément que la chaîne vw (*abba*) est différente de la chaîne wv (*baab*).

2 Langages et grammaires

Définition 2.1 (Notion de langage)

Un langage est un ensemble de mots construits à base d'un alphabet.

Par exemple, les mots d'une langue naturelle comme le Français, sont construits à partir de l'alphabet latin. Le langage des identificateurs est construit à base des caractères alphanumériques (lettres et chiffres).

Un langage est utilisé de deux façons différentes :

En mode générateur (*ou locuteur*)
Le langage est décrit par une grammaire qui suit des règles bien précises. Le *locuteur* est, par exemple, un programmeur qui rédige un programme dans le langage de son choix.

En mode récepteur (ou auditeur)
La réception sous-entend *reconnaissance*. Dans un système d'interaction homme-machine, cette *reconnaissance* peut être assurée par des machines virtuelles comme les automates. Un autre exemple est celui d'un compilateur qui peut être considéré comme l'*auditeur* qui reçoit un programme écrit dans un langage donné.

Définition 2.2 (Langage)

Etant donné un alphabet de base V. *On appelle langage* L *sur l'alphabet* V, *un sous-ensemble de chaînes de* V^*. *Autrement dit,* $L \subseteq V^*$.

Par exemple, étant donné l'alphabet $V = \{0, 1\}$. Le monoïde $V^* = \{0, 1\}^*$ correspond à l'ensemble de tous les mots formés à partir de l'alphabet de base V. Les langages suivants sont inclus dans V^* :

- L_1 est le langage formé de toutes les chaînes de V^*de longueur 2. Ce langage est représenté par l'ensemble noté V^2. La puissance ^{2} indique que la longueur des chaines est exactement égale à 2. Ainsi, $L_1 = V^2 = \{v \in V^* \mid |v| = 2\} = \{00, 01, 10, 11\}$.

- L_2 est l'ensemble des chaînes de longueur au plus égale à 2. On note L_2 par V^{*2}. La puissance ^{*2} indique que la longueur des chaines est bornée supérieurement par 2. Ainsi, $L_2 = V^{*2} = \{v \in V^* \mid |v| \leq 2\} = \{\varepsilon, 0, 1, 00, 01, 10, 11\}$.

- L_3 est le langage représentant des nombres binaires impairs. $L_3 = \{\{0, 1\}^*1\}$. On voit ici que la combinaison $\{0, 1\}^*1$ représente les éléments de V^* auxquels on concatène 1 par la droite. Autrement dit, le langage L_3 est composé de toutes les chaines binaires se terminant par 1, représentant, de toute évidence, des nombres binaires impairs.

- L_4 est le langage représentant des nombres binaires pairs. $L_4 = \{\{0, 1\}^*0\}$. De même, on voit très bien que les chaines de L_4 se terminent par 0, donc correspondant nécessairement à des nombres binaires pairs.

2.1 Opérations sur les langages

Etant donné un alphabet V et soient L_1 et L_2 deux langages sur V, c'est-à-dire L_1 et L_2 sont inclus ($\subseteq$) dans V^*. On définit les opérations sur les langages, comme en théorie des ensembles, de la manière suivante :

- *Union* $L_1 \cup L_2 = \{w \in V^* \mid w \in L_1 \text{ ou } w \in L_2\}$
- *Intersection* $L_1 \cap L_2 = \{w \in V^* \mid w \in L_1 \text{ et } w \in L_2\}$
- *Produit* (*Concaténation*) $L_1.L_2 = \{vw \in V^* \mid v \in L_1 \text{ et } w \in L_2\}$

Par exemple, soient V = {a, b}, L_1 = {aa, bb, ab, ba} et L_2 = {ab, bb, ba}. On a alors :

$L_1 \cup L_2$ = {aa, ab, ba, bb} = L_1
$L_1 \cap L_2$ = {ab, ba, bb} = L_2
$L_1.L_2$ = {aaab, aabb, aaba, bbab, bbbb, bbba, abab, abbb, abba, baab, babb, baba}.

- *Itération et étoile d'un langage*
  - On définit *l'itération* par la relation $L^i = L.L^{i-1}$ avec $i \geq 1$. Par convention, $L^0 = \{\varepsilon\}$, à ne pas confondre avec $\varnothing$ qui est le langage vide, ne contenant aucune chaine, pas même la chaine vide ε.
  - *L'étoile* d'un langage est définie par $L^* = \bigcup_{i \geq 0} L^i$, $L^* = L^0 \cup L^1 + \ldots = \bigcup_{i \geq 0} L^i$
    $L^+ = L^1 \cup L^2 \cup \ldots = \bigcup_{i \geq 1} L^i$, alors $L^* = L^+ \cup L^0$ et $L^+ = L\,L^* = L^*L$.
- *Reflet miroir*

  On définit l'opération miroir de L par $L^R = \{w \mid w = v^R \text{ avec } v \in L\}$; v^R est le reflet miroir ou mot miroir de v.

  Ainsi, si $v = a_1 \ldots a_n$, alors $v^R = a_n \ldots a_1$.

- *Complémentarité*

  On définit le complémentaire d'un langage L par $L_C = \{w \in V^* \mid w \notin L\}$.

- *Différence*

  La différence de L_1 et L_2 est le langage noté $L_1 - L_2$ constitué des mots appartenant à L_1 et n'appartenant pas à L_2. $L_1 - L_2 = \{w \in V^* \mid w \in L_1 \text{ et } w \notin L_2\}$.

Par exemple, si l'on reconsidère L_1 = {aa, bb, ab, ba} et L_2 = {ab, bb, ba}, on aura :

- L_1^R = {aa, bb, ba, ab} = L_1 / * langage palindrome * /
- L_2^R = {ba, bb, ab} = L_2 / * langage palindrome * /
- $L_{1C} = V^* - L_1 = \{\varepsilon, a, b, \{a, b\}^n \mid n \geq 3\}$
- $L_{2C} = V^* - L_2 = \{\varepsilon, a, b, aa, \{a, b\}^n \mid n \geq 3\}$
- $L_1 - L_2$ = {aa}
- $(L_1 - L_2)^* = \{aa\}^* = \{\varepsilon, aa \ldots (aa)^n \ldots\} = \{(aa)^n \mid n \geq 0\} = \{a^{2n} \mid n \geq 0\}$.

Définition 2.3 (Grammaire formelle)

On appelle grammaire formelle, le quadruplet noté $G = (V_N, V_T, S, P)$ *où* :

- V_N est un vocabulaire (ou alphabet) non-terminal (on dit aussi auxiliaire), qui est un ensemble fini non vide ($\neq\emptyset$) de symboles non-terminaux.
- V_T est un vocabulaire terminal (ou alphabet de base), qui consiste en un ensemble fini non vide ($\neq\emptyset$) de symboles terminaux.
  On note que l'intersection de V_T et V_N est toujours vide ($V_T \cap V_T = \emptyset$).
- $S \in V_N$ est le symbole initial ou l'axiome de la grammaire G.
- P est l'ensemble des règles de production de G, défini par $\{\alpha \rightarrow \beta\}$ avec $\alpha \in V^+$ et $\beta \in V^*$ tel que $V = V_T \cup V_N$. La flèche au niveau de la règle $\alpha \rightarrow \beta$ indique que le membre gauche α produit ($\rightarrow$) le membre droit β.

Remarque 2.1

Le symbole α est appelé membre gauche et β le membre droit de la règle de production. Par ailleurs, si α possède plusieurs alternatives (β_1, ..., β_n : membres droits) comme, par exemple, $\alpha \rightarrow \beta_1$; $\alpha \rightarrow \beta_2$; $\alpha \rightarrow \beta_3$.... $\alpha \rightarrow \beta_n$, on simplifie cette écriture en utilisant la barre verticale « | » ; on écrira dans ce cas : $\alpha \rightarrow \beta_1 \mid \beta_2 \mid \beta_3 ... \mid \beta_n$. Le symbole « | » indique alors un choix.

2.2 Classification des grammaires

Chomsky a introduit en 1956 une hiérarchie dans laquelle les grammaires sont classées en 4 types qui sont alors transmis au type du langage. Ainsi, un langage de *type i* est engendré par une grammaire de *type i*. Les grammaires peuvent être classées en fonction de la nature de leurs règles de production. Les quatre types sont hiérarchiquement imbriqués (*type 3* $\subset$ *type 2* $\subset$ *type 1* $\subset$ *type 0)* de sorte que les langages de type 0 (dits **généraux**) incluent les langages de type 1 (dits **contextuels**) qui incluent eux-mêmes les langages de type 2 (dits **hors-contexte**) qui, à leur tour, incluent les langages de type 3 (dits **réguliers**). Les 4 types sont définis en fonction de la nature des règles de production. En effet, une grammaire définie par le quadruplet $G = (V_N, V_T, S, P)$ sera dite de :

- *Type 3* :
  - *Régulière à droite* : Si toutes ses règles de production sont de la forme : $A \rightarrow \alpha B$ ou $A \rightarrow \alpha$, avec $\alpha \in V_T^*$, A et $B \in V_N$.
  - *Régulière à gauche* : Si toutes ses règles de production sont de la forme : $A \rightarrow B\alpha$ ou $A \rightarrow \alpha$, avec $\alpha \in V_T^*$, A et $B \in V_N$.

- *Type 2* : Si toutes ses règles de production sont de la forme $A \rightarrow \alpha$ avec $A \in V_N$ et $\alpha \in (V_T \cup V_N)^*$.

- *Type 1* : Si toutes les règles sont de la forme $\alpha \rightarrow \beta$, sachant que $|\alpha| \leq |\beta|$ avec $\alpha \in V^* V_N V^*$ et $\beta \in V^+$. En d'autres termes, on peut aussi avoir $\lambda A \delta \rightarrow \lambda \gamma \delta$, avec $A \in V_N$, λ, $\delta \in V^*$ et $\gamma \in V^+$. Mais, pour permettre à ce type de grammaire de générer le mot vide (lorsque le langage engendré par cette grammaire contient le mot vide), on introduit l'exception $S \rightarrow \varepsilon$, mais l'axiome S ne doit apparaitre dans aucun membre droit des autres règles.

- *Type 0* : Si ses productions ne sont l'objet d'aucune restriction, c'est-à-dire $\alpha \rightarrow \beta$ avec α et $\beta \in V^*$ si et seulement si $|\alpha| \geq 1$. Autrement dit, il ne pourrait exister de règle de la forme $\varepsilon \rightarrow \beta$.

Par exemple, les règles de production suivantes :

- S→0S | 1S | 0 | 1, définissent une grammaire de type 3 (régulière à droite).
- S→S0 | S1 | 0 | 1, définissent une grammaire de type 3 (régulière à gauche).
- S→S+A | A ; A→(S) | a, définissent une grammaire de type 2 (hors contexte).
- S→CSA | CDc, cA→Bc, B→A, D→b, bA→b Dc, C→a, définissent une grammaire de type 1 (contextuelle).
- S→RT | ε, R→aRA | bRB | ε, AT→aT, BT→bT, Ba→aB, Bb→bB, Aa→aA, Ab→bA, aT→a, bT→b, définissent une grammaire de type 0 (générale).

Remarque 2.2
Dans la suite de ce chapitre, on emploie le terme grammaire, toujours pour désigner une grammaire hors contexte (ou à contexte libre) ou une grammaire régulière. On utilisera également indifféremment les termes production et règle, pour désigner une règle de production.

Définition 2.4 (Dérivation)
Soit $G = (V_N, V_T, P, S)$ une grammaire formelle.

- **Dérivation directe**
  On appelle dérivation directe (α, β), *et on la note par* $\alpha \Rightarrow \beta$. *On dit aussi que* β *dérive directement de* α *si* $\exists$ *la production* $\alpha \rightarrow \beta \in P$.

- **Dérivation indirecte**
  On appelle dérivation indirecte, et on la note par $\alpha \Rightarrow^* \beta$ *ou* $\alpha \Rightarrow^+ \beta$. *On dit aussi que* β *dérive indirectement de* α *si* $\exists$ *une succession de dérivations directes entre* α *et* β. *Autrement dit*, $\alpha \Rightarrow \beta_1 \Rightarrow \beta_2 \Rightarrow \beta_3 \Rightarrow \beta i \Rightarrow ... \Rightarrow \beta$.

Avec $\Rightarrow^+$, on a un nombre strictement positif de dérivations directes (itération positive) permettant de passer de α à β ($\alpha \Rightarrow \beta 1 \Rightarrow \beta 2 \Rightarrow ... \beta i \Rightarrow ... \Rightarrow \beta$). En revanche, avec $\Rightarrow^*$, on a un nombre positif ou nul de dérivations directes (transition positive et réflexive). Autrement dit, outre la succession de dérivations directes, il peut y avoir aussi la dérivation $\alpha = \beta$. En bref, ces deux dérivations ($\Rightarrow^+$ et $\Rightarrow^*$) sont liées par les deux relations suivantes :

- $\alpha \Rightarrow^* \beta \equiv (\alpha \Rightarrow^+ \beta) \vee (\alpha = \beta)$.
- $\alpha \Rightarrow^+ \beta \equiv (\alpha \Rightarrow^* \beta) \wedge (\alpha \neq \beta)$.

Remarque 2.3
Par abus de langage, on entend souvent par dérivation, une dérivation indirecte.

Par exemple, on donne la grammaire $G = (V_N, V_T, S, P)$ où :

- $V_N = \{S\}$
- $V_T = \{0, 1\}$
- $P = \{S \rightarrow 0S \mid 1S \mid 0 \mid 1\}$
- S est l'axiome, et il est l'unique non-terminal de V_N.

Cette grammaire génère l'ensemble des nombres binaires purs. Les mots 0 et 1 sont obtenus par des dérivations directes. En effet, on a bien $S \Rightarrow 0$ et $S \Rightarrow 1$, car il existe les règles de production : $S \rightarrow 0$ et $S \rightarrow 1$.

Les chaines 0001S et 10011 sont obtenues par des dérivations indirectes. En effet, on a :

- $S \Rightarrow 0S \Rightarrow 00S \Rightarrow 000S \Rightarrow 0001S$
- $S \Rightarrow 1S \Rightarrow 10S \Rightarrow 100S \Rightarrow 1001S \Rightarrow 10011$

A chaque pas $\Rightarrow$ (dérivation directe), une règle de production est appliquée. Avec la dérivation indirecte $S \Rightarrow^* 001S$, la règle $S \rightarrow 0S$ est appliquée trois fois consécutivement, à savoir, $S \Rightarrow 0S \Rightarrow 00S \Rightarrow 000S$; ensuite, c'est la règle $S \rightarrow 1S$ qui est appliquée, ce qui donne finalement $000S \Rightarrow 0001S$. Le *Tableau I* illustre la même démarche pour la deuxième dérivation, c'est-à-dire $S \Rightarrow^* 10011$.

dérivation en cours	règle appliquée	chaine obtenue
$S \Rightarrow 1S$	$S \rightarrow 1S$	1S
$S \Rightarrow 1S \Rightarrow 10S$	$S \rightarrow 0S$	10S
$S \Rightarrow 1S \Rightarrow 10S \Rightarrow 100S$	$S \rightarrow 0S$	100S
$S \Rightarrow 1S \Rightarrow 10S \Rightarrow 100S \Rightarrow 1001S$	$S \rightarrow 1S$	1001S
$S \Rightarrow 1S \Rightarrow 10S \Rightarrow 100S \Rightarrow 1001S \Rightarrow 10011$	$S \rightarrow 1$	10011

***Tableau I**- Exemple de dérivation indirecte*

2.3 Représentation de langages

Très souvent, on utilise les grammaires et les automates pour représenter ou décrire des langages. Mais, il existe d'autres modèles tout aussi puissants, voire parfois plus adaptés en pratique, comme les graphes syntaxiques, les expressions régulières, les diagrammes syntaxiques ou diagrammes de transition, etc., qui sont des variantes de ces deux systèmes. Formellement, il existe une distinction nette entre les grammaires, considérées comme des systèmes générateurs, et leurs homologues les automates, considérés comme des *reconnaisseurs*. Cependant, si des systèmes distincts, décrivent le même langage, ils seront considérés comme équivalents, même s'ils sont spécifiés par des formalismes différents.

Par exemple, on reconsidère la grammaire du paragraphe précédent, $G = (V_N, V_T, S, P)$ où :

- $V_N = \{S\}$
- $V_T = \{0, 1\}$
- $P = \{S \rightarrow 0S \mid 1S \mid 0 \mid 1\}$
- S est l'axiome, et il est l'unique non-terminal dans V_N.

Sur la *Figure 1* on présente le diagramme syntaxique équivalent à G, c'est-à-dire qui décrit le même langage que G, à savoir, le langage des nombres binaires dans le système de numération binaire pur.

En effet, à l'entrée **IN** du diagramme, la flèche permet de transiter, soit vers le cercle qui contient le 0, soit vers le cercle qui contient le 1. On peut sortir

immédiatement par **OUT** évidemment, et là, on a reconnu un 0 ou un 1 ; ce qui correspond respectivement aux productions $S \rightarrow 0$ et $S \rightarrow 1$. Sinon, l'autre flèche (en pointillés) au niveau de l'entrée **IN** du diagramme, permet de repartir à nouveau, soit avec un 0, soit avec un 1, et ainsi de suite ; ce qui correspond exactement à l'effet des règles de production $S \rightarrow 0S \mid 1S$.

En somme, cela signifie que le diagramme est susceptible de reconnaitre un nombre potentiellement infini de 0 et de 1 (l'ordre importe peu), c'est-à-dire qu'il accepte les chaines représentant des nombres binaires purs, tout comme la grammaire G. Ceci démontre intuitivement l'équivalence des deux systèmes, à savoir, le diagramme syntaxique de la *Figure 1* et la grammaire G.

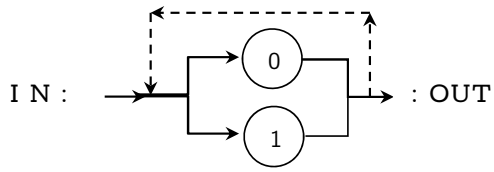

***Figure 1** : Diagramme syntaxique des nombres binaires*

Définition 2.5 (Langage engendré par une grammaire)

On appelle langage engendré par la grammaire $G = (V_N, V_T, P, S)$, *l'ensemble nommé* $L(G) = \{\omega \in V_T^* \mid S \Rightarrow^* \omega\}$.

Soit une grammaire $G = (V_N, V_T, P, S)$. Pour vérifier si une certaine chaîne ω appartient au langage L (G), on peut adopter, soit une stratégie d'analyse descendante, soit une stratégie d'analyse ascendante.

- *Avec l'analyse descendante*, on démarre à partir de l'axiome S, et par une succession de dérivations, on tente de faire apparaître le mot ω.
- *Avec l'analyse ascendante*, on démarre avec le mot ω et on tente de remonter vers l'axiome S par une succession de dérivations inverses nommées réductions.

Ces deux stratégies seront étudiées en détails dans le chapitre 6 qui est consacré aux méthodes d'analyse syntaxique.

Par exemple, on considère la grammaire $G = (V_N, V_T, P, E)$ avec :

- $V_N = \{E, T, F\}$
- $V_T = \{a, +, *, (,)\}$
- $P = \{E \rightarrow T + E ; E \rightarrow T ; T \rightarrow F * T ; T \rightarrow F ; F \rightarrow a ; F \rightarrow (E)\}$

Soit alors à vérifier si la chaine "a * (a + a)" appartient à L(G), en utilisant la stratégie d'analyse descendante. La dérivation à partir de l'axiome est illustrée dans le *Tableau II*.

La chaîne donnée "a * (a + a)" est engendrée par la grammaire G. Autrement dit, la chaine "a * (a + a)" $\in$ L(G) puisque on a : $E \Rightarrow^* a * (a + a)$. On note que L(G) est le langage des expressions arithmétiques très simplifié.