Langages hors-contexte

Au-delà des langages réguliers

Les langages hors-contexte (angl. Context-Free Languages, CFL) ont de nombreuses applications dans l'étude des langages naturels, en compilation (syntaxe de langages de programmation), analyse de programmes, etc.

Exemples

Langages des parenthèses, des expressions arithmétiques, des palindromes, $\{a^n b^n \mid n \geq 0\}$, ...

Grammaires hors-contexte

Une grammaire hors-contexte (CGF) $G = \langle V, \Sigma, R, S \rangle$ consiste d'un ensemble (fini) V de symboles non-terminaux (ou variables), d'un alphabet (fini) Σ de symboles terminaux, d'un ensemble de règles $R \subseteq V \times (V \cup \Sigma)^*$ et d'une variable initiale (axiome) $S \in V$.

Langages hors-contexte

Exemple

<phrase>	$\rightarrow$	<GN><GV>
<GN>	$\rightarrow$	<nom-complexe> | <nom-complexe><CdN>
<nom-complexe>	$\rightarrow$	<article><nom>
<CdN>	$\rightarrow$	<adjectif> | ...
<GV>	$\rightarrow$	<verbe><GN>
<article>	$\rightarrow$	le | la | mon | ma ...
<nom>	$\rightarrow$	fille | garçon | danse
<adjectif>	$\rightarrow$	aîné(e) | ...
<verbe>	$\rightarrow$	aime | ...

Dérivation

L'application d'une règle $A \rightarrow v$ de R au mot $uAw \in (V \cup \Sigma)^*$ produit le mot uvw (on note $uAw \Rightarrow uvw$). On écrit $u \stackrel{*}{\Longrightarrow} v$ si l'on peut dériver le mot u du mot v, c-a-d. si on a soit $u = v$, ou s'il existent $u_1, \ldots, u_n \in (V \cup \Sigma)^*$ tels que $u \Rightarrow u_1 \Rightarrow \cdots \Rightarrow u_n \Rightarrow v$.

Le langage engendré par G est défini par $\mathcal{L}(G) = \{w \in \Sigma^* \mid S \stackrel{*}{\Longrightarrow} w\}$.

Langages hors-contexte

Exemple (dérivation)

<phrase> ⇒ <GN><GV> → <nom-complexe><CdN><GV> ⇒
<article><nom><CdN><GV> ⇒ ma <nom><CdN><GV> ⇒
ma fille<CdN><GV> ⇒ ma fille<adjectif><GV> ⇒
ma fille aîné(e)<GV> $\stackrel{*}{\Longrightarrow}$ ma fille aîné(e) aime la danse

Dérivation gauche

Une dérivation gauche est une dérivation où on remplace toujours la variable la plus à gauche (si possible), voir exemple.

Un arbre de dérivation pour un mot $w \in \Sigma^*$ à partir d'une variable $B \in V$ est un arbre ordonné étiqueté, dont les nœuds internes sont étiquetés par des variables dans V, et les feuilles par des terminaux (dans Σ) ou ϵ ; pour tout nœud v étiqueté par $A \in V$, si $A_1, \ldots, A_n \in V \cup \Sigma$ sont les étiquettes des enfants $v1, \ldots, vn$ de v, alors $A \to A_1 \cdots A_n$ est une règle de G. La racine est étiquetée par B, et w est la frontière de l'arbre.

Exemples - grammaires

- Une CFG qui engendre des expressions arithmétiques utilisant $+, *$ et les variables a, b, c (langage non-régulier !) :

$$\begin{array}{lcl} E & \rightarrow & E + T \mid T \\ T & \rightarrow & T * F \mid F \\ F & \rightarrow & (E) \mid a \mid b \mid c \end{array}$$

- Une CFG qui engendre le langage $\{a^n b^n \mid n \geq 0\}$:

$$S \rightarrow aSb \mid \epsilon$$

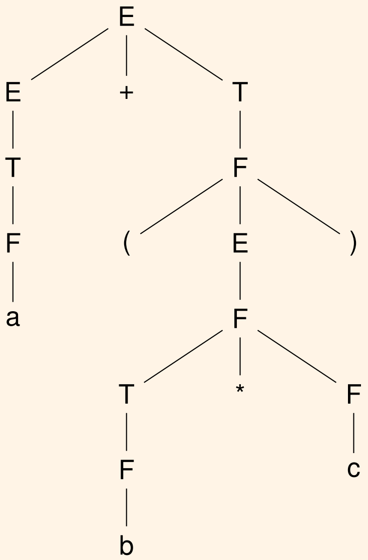
- Une CFG qui engendre les palindromes sur $\{a, b\}$:

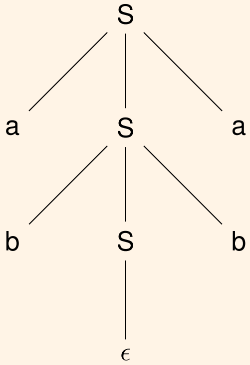
$$S \rightarrow aSa \mid bSb \mid \epsilon$$

Définition

Un langage de mots s'appelle hors-contexte (angl. context-free language, CFL), s'il est engendré par une grammaire hors-contexte.

Exemples - arbres de dérivation





REG $\subseteq$ CFL

Soit $L \subseteq \Sigma^*$ accepté par un NFA $\mathcal{A} = \langle Q, \Sigma, \delta, q_0, F \rangle$. Alors L est engendré par la CFG $G = \langle Q, \Sigma, R, q_0 \rangle$, où R contient toutes les règles de la forme $p \to aq$ si $q \in \delta(p, a)$, ainsi que $q \to \epsilon$ si $q \in F$.

Ambiguïté

Une CFG G telle qu'il existe un mot dans $\mathcal{L}(G)$ qui possède au moins 2 arbres de dérivation à partir de l'axiome, s'appelle ambiguë. Un langage hors-contexte L est ambigu, si toute CFG qui l'engendre est ambiguë.

Exemple

La CFG $G = \langle \{E\}, \{a, b, c\}, R, E \rangle$ où R contient les règles $E \to E + E \mid E * E \mid (E) \mid a \mid b \mid c$ est ambiguë. Le langage des expressions arithmétiques ne l'est pas (la CFG page 4 est non-ambiguë).

Chomsky

Une CFG est en forme normale de Chomsky si les toutes les règles ont la forme $A \rightarrow BC$ ou $A \rightarrow a$, avec A, B, C des variables et a un symbole terminal. On permet aussi la règle $S \rightarrow \epsilon$.

Proposition

Toute CFG G peut être transformée en une CFG G' équivalente (c-a-d., t.q. $\mathcal{L}(G) = \mathcal{L}(G')$) en forme normale de Chomsky.

Proposition

Etant donnée une CFG G en forme normale de Chomsky et un mot $w \in \Sigma^*$, on peut décider en temps polynomial en $|w|$ et G, si $w \in \mathcal{L}(G)$ (algorithme CYK : Cocke/Younger/Kasami).

Clôture

Opérations de clôture

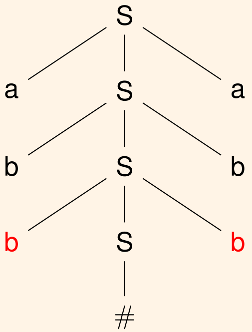
La famille des langages hors-contexte est fermée par les opérations rationnelles (union, produit, itération) et par intersection avec les réguliers. Elle n'est pas fermée ni par complémentaire, ni par intersection.

Lemme de l'étoile pour les CFL

Soit $L \subseteq \Sigma^*$ un langage hors-contexte. Alors il existe un entier $N > 0$ tel que pour tout mot $z \in L$ de longueur supérieure à N, il existe une décomposition $z = uvwxy$ avec les propriétés suivantes :

1. $|vwx| \leq N$.
2. $vx \neq \epsilon$.
3. Pour tout $k \geq 0$, le mot uv^kwx^ky appartient à L.

Exemple



$u = ab$, $v = b = x$, $w = \#$, $y = ba$

Applications

1. Pour toute CFG G il existe un entier $N > 0$ (qui dépend de G) t.q. $\mathcal{L}(G)$ est infini ssi il existe un mot $w \in \mathcal{L}(G)$ de longueur $|w| > N$.
2. Le lemme de l'étoile permet de démontrer qu'un langage n'est pas hors-contexte.

   L n'est pas CFL si :

   > *Quelque soit* $N > 0$ *il existe* $z_N \in \mathcal{L}(G)$ *de longueur supérieure à* N *t.q. quelque soit la décomposition* $z_N = uvwxy$ *satisfaisant* $|vwx| \leq N$ *et* vx *non-vide, il existe* $k \geq 0$ *t.q.* $uv^kwx^ky \notin L$.

- Exemple 1 : $\{a^n b^n c^n \mid n \geq 0\}$ n'est pas CFL.
  Par contre, $L_1 = \{a^n b^n c^m \mid m, n \geq 0\}$ et $L_2 = \{a^n b^m c^n \mid m, n \geq 0\}$ sont CFL - donc $L_1 \cap L_2$ ne l'est pas.
- Exemple 2 : $L = \{w\#w \mid w \in \{a, b\}^*\}$ n'est pas CFL.
  Par contre L^{co} est CFL (pas immédiat à voir...).

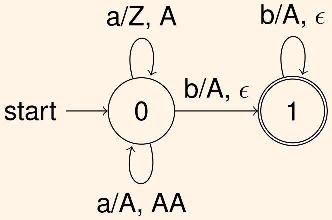
Définition

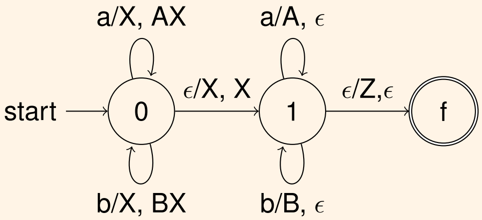
- Un automate à pile $\mathcal{A}$ est un automate fini, auquel on rajoute une mémoire sous forme de pile. Formellement, $\mathcal{A} = \langle Q, \Sigma, \Gamma, \delta, q_0, F, Z \rangle$, où Q est l'ensemble (fini) d'états, Σ l'alphabet de l'entrée, Γ l'alphabet de la pile, $\delta \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \times Q \times \Gamma^*$ la relation de transition, $q_0 \in Q$ l'état initial, $F \subseteq Q$ l'ensemble des états finaux et $Z \in \Gamma$ le fond de la pile.
- Une transition (p, a, A, q, v) lit le symbole actuel $a \in \Sigma$ (ou rien si $a = \epsilon$), et remplace le sommet $A \in \Gamma$ de la pile par le mot $v \in \Gamma^*$. L'état change de p à q.
- Une configuration (état généralisé) de $\mathcal{A}$ est un couple $(p, w) \in Q \times \Gamma^*$ consistant de l'état de contrôle p et le mot de pile $w \in \Gamma^*$ (le sommet étant le premier symbole de w). Une transition de $\mathcal{A}$ par (p, a, A, q, v) correspond donc au passage d'une configuration (p, Aw) à la configuration (q, vw), en lisant $a \in \Sigma \cup \{\epsilon\}$: $(p, Aw) \xrightarrow{a} (q, vw)$.
  On écrit $(p, w) \xRightarrow{u} (p', w')$ s'il existe une suite de transitions $(p, w) \xrightarrow{a_0} (p_1, w_1) \xrightarrow{a_1} \cdots \xrightarrow{a_{n-1}} (p_n, w_n) \xrightarrow{a_n} (p', w')$ telle que $u = a_0 \cdots a_n$.
- La configuration initiale est (q_0, Z). Le langage accepté par $\mathcal{A}$ est $\mathcal{L}(\mathcal{A}) = \{u \in \Sigma^* \mid (q_0, Z) \xRightarrow{u} (q, w),\ q \in F, w \in \Gamma^*\}$.

Remarques

- On peut définir le langage accepté par un automate à pile aussi par pile vide (avec ou sans état final) : on demande $(q_0, Z) \stackrel{u}{\Longrightarrow} (q, \epsilon)$ dans $\mathcal{L}(\mathcal{A})$. Ces variantes sont toutes équivalentes.
- Les automates à pile déterministes sont plus faibles. Pour eux, l'acceptation par état final ou par pile vide ne sont pas équivalentes.

Exemples





Le premier automate reconnaît $\{a^m b^n \mid m \geq n \geq 0\}$, le deuxième reconnaît les palindromes de longueur paire ($X \in \{A, B, Z\}$).

CFG, automates et arbres

Théorème

Pour tout langage hors-contexte il existe un automate à pile (à un état) qui le reconnaît (avec pile vide). Réciproquement, les langages reconnus par les automates à pile sont des langages hors-contexte.

Proposition

1. Soit G une CFG. L'ensemble des arbres de dérivation de G est reconnaissable.
2. Soit L un langage reconnaissable d'arbres. L'ensemble des frontières des arbres de L est un langage hors-contexte.