Diapositives sans animation disponibles sur Chamilo
(pour téléphone)

Projet : disponible dès maintenant

- À faire seul ou par binôme
- À rendre le 14 décembre 2025 sur teide
- Vous ne pourrez pas tout faire dès maintenant :
  - partie lexicale faisable tout de suite
  - partie grammaticale (cours 9 et 10)

Théorie des Langages

Cours 7 : Grammaires HC : arbres, preuves, propriétés

Lionel Rieg

Grenoble INP - Ensimag, 1re année

Exemples

Exemple (Expressions arithmétiques)

$$
\begin{array}{lcl}
\text{exp} & \rightarrow & \text{num} \mid \text{exp op exp} \mid (\text{exp}) \\
\text{op} & \rightarrow & + \mid - \mid * \mid / \\
\text{num} & \rightarrow & \text{chiffre} \mid \text{chiffre num} \\
\text{chiffre} & \rightarrow & \mathbf{0} \mid \ldots \mid \mathbf{9}
\end{array}
$$

Exemple (Extrait des instructions Python)

$$
\begin{array}{lcl}
\text{statement} & \rightarrow & \text{compound\_stmt} \mid \text{simple\_stmts} \\
\text{simple\_stmts} & \rightarrow & \text{simple\_stmt ; simple\_stmts} \\
 & & \mid \text{simple\_stmt} \mid \text{simple\_stmt ;} \\
\text{simple\_stmt} & \rightarrow & \text{assignment} \mid \text{return\_stmt} \mid \text{raise\_stmt} \\
 & & \mid \mathbf{pass} \mid \mathbf{break} \mid \mathbf{continue} \\
 & & \mid \ldots \\
 & \vdots &
\end{array}
$$

Theorème de décomposition

Rappels (sur une grammaire quelconque $G = \langle V_T, V_N, \mathsf{S}, R \rangle$)

- $xuy \Longrightarrow xvy$ si $u \to v \in R$
- $\Longrightarrow^p$ dérivations de longueur p et $\Longrightarrow^*$ de longueur quelconque
- $\mathcal{L}(u) = \{w \in V_T^* \mid u \Longrightarrow^* w\}$ et $\mathcal{L}(G) = \mathcal{L}(\mathsf{S})$
- Composition : si $u_1 \Longrightarrow^{p_1} v_1$ et $u_2 \Longrightarrow^{p_2} v_2$, alors $u_1 u_2 \Longrightarrow^{p_1+p_2} v_1 v_2$

Théorème (Décomposition des dérivations pour G HC)

Si $u_1 \dots u_n \Longrightarrow^p w$ *avec* $u_i \in V^*$ *et* $w \in V^*$,
alors $\exists w_1, \dots, w_n, \in V^*, \exists p_1, \dots, p_n \in \mathbb{N}$ *tels que*

- $w = w_1 \dots w_n$
- $p = p_1 + \dots + p_n$
- $\forall i, u_i \Longrightarrow^{p_i} w_i$

Démonstration du théorème de décomposition

Par récurrence sur p, la longueur de la dérivation :

Base $p = 0$:
Par définition de $\Longrightarrow^0$, $u_1 \dots u_n \Longrightarrow^0 w$ donne $w = u_1 \dots u_n$.
On pose donc $\forall i, w_i = u_i$ et $p_i = 0$ et on a bien :

- $w = u_1 \dots u_n = w_1 \dots w_n$
- $p = 0 = p_1 + \dots + p_n$
- $\forall i, u_i \Longrightarrow^0 w_i = u_i$

Démonstration du théorème de décomposition

Par récurrence sur p, la longueur de la dérivation :

Hérédité $p+1$:
Par définition de $\Longrightarrow^{p+1}$, $u_1 \dots u_n \Longrightarrow^{p+1} w$ donne
$u_1 \dots u_n \xrightarrow{\mathsf{X} \to t} v \Longrightarrow^{p} w$. Soit u_k le mot où la règle $\mathsf{X} \to t$ s'applique.
On a donc $u_k = \alpha \mathsf{X} \beta$ et $v = u_1 \dots u_{k-1} \alpha t \beta u_{k+1} \dots u_n$.
Par hypothèse de récurrence sur p pour $u_1, \dots, u_{k-1}, \alpha t \beta, u_{k+1}, \dots u_n$,
on obtient $w_1, \dots w_n$, $p_1, \dots p_n$ tels que

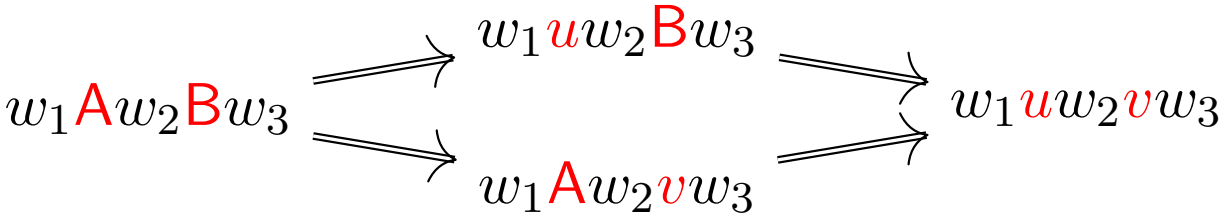
- $p_1 + \dots + p_n = p$,
- $w = w_1 \dots w_n$,
- $\forall i \neq k, u_i \Longrightarrow^{p_i} w_i$ et $\alpha t \beta \Longrightarrow^{p_k} w_k$.

Alors, les $w_1, \dots, w_n$ et $p_1, \dots, p_{k-1}, p_k + 1, p_{k+1}, \dots, p_n$ conviennent :

- $p + 1 = p_1 + p_{k-1} + (p_k + 1) + p_{k+1} + \dots + p_n$,
- $w = w_1 \dots w_n$,
- $\forall i \neq k, u_i \Longrightarrow^{p_i} w_i$ et $u_k \Longrightarrow \alpha t \beta \Longrightarrow^{p_k} w_k$ donc $u_k \Longrightarrow^{p_k+1} w_k$.

Conséquences du théorème

- Faux pour les grammaires générales : $a\mathsf{A} \to b \qquad \mathsf{B}a \to a$
  $\mathsf{B}a\mathsf{A} \Longrightarrow a\mathsf{A} \Longrightarrow b$
- L'ordre d'application des règles n'a pas d'importance :



- On choisit un ordre canonique (ex. : le non-terminal le plus à gauche).
- On peut faire des preuves « par morceaux ».

Corollaire le plus utile

Pour $u_i \in V$, $u_1 \dots u_n \Longrightarrow^* w \in V_T^*$,
On obtient $u_i \Longrightarrow^* w_i$ avec $w = w_1 \dots w_n$.

(en sachant que $u_i \in V_T$ donne $w_i = u_i$)

Dérivation canonique

Pour éviter de considérer comme différentes des dérivations où on permute des dérivations indépendantes, on fixe un ordre sur les non terminaux, en général de gauche à droite.

Définition (Dérivation gauche)

On transforme toujours le non -terminal le plus à gauche :
$\alpha \mathsf{A} \beta \Longrightarrow \alpha v \beta \quad$ avec $\alpha \in V_T^* \quad$ et $\quad v, \beta \in V^*$

Exemple (Mots bien parenthésés)

Grammaire : S $\rightarrow$ SS | $\mathbf{aSb}$ | ε
Dérivations équivalentes :

- canonique : $\mathsf{S} \Longrightarrow \mathsf{SS} \Longrightarrow a\mathsf{S}b\mathsf{S} \Longrightarrow ab\mathsf{S} \Longrightarrow aba\mathsf{S}b \Longrightarrow abab$
- non canonique : $\mathsf{S} \Longrightarrow \mathsf{SS} \Longrightarrow a\mathsf{S}b\mathsf{S} \Longrightarrow a\mathsf{S}ba\mathsf{S}b \Longrightarrow a\mathsf{S}bab \Longrightarrow abab$

Une dérivation différente (mais canonique) :
$\mathsf{S} \Longrightarrow \mathsf{SS} \Longrightarrow a\mathsf{S}b\mathsf{S} \Longrightarrow ab\mathsf{S} \Longrightarrow ab\mathsf{SS} \Longrightarrow aba\mathsf{S}b\mathsf{S} \Longrightarrow abab\mathsf{S} \Longrightarrow abab$

Arbres de dérivation

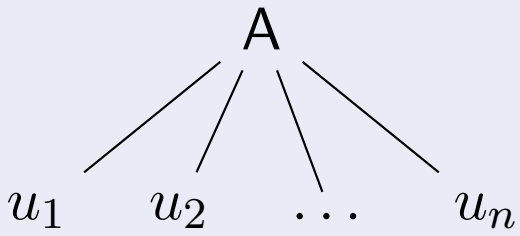
Représentation alternative d'une dérivation : arbre de dérivation

Intérêts :

- toujours canonique
- permet de visualiser le parallélisme

Définition (Arbre de dérivation)

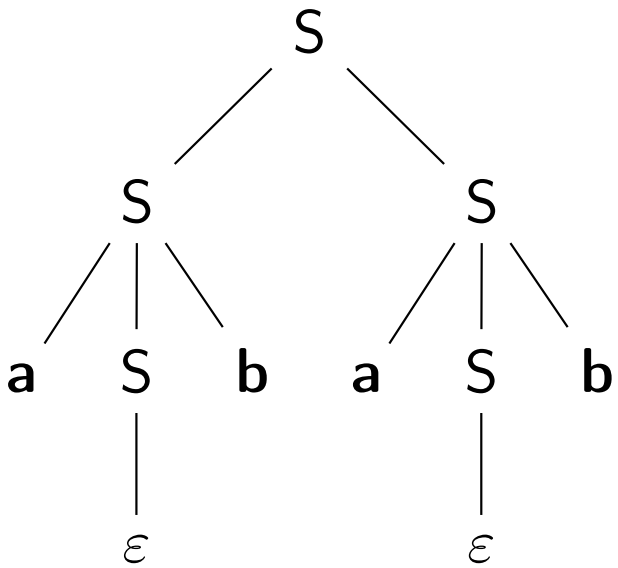
- nœuds internes étiquetés par un non terminal
- feuilles étiquetées par un symbole terminal ou ε
- racine = axiome S
- appliquer la règle A $\rightarrow$ $u_1 u_2 \dots u_n$ donne le sous-arbre



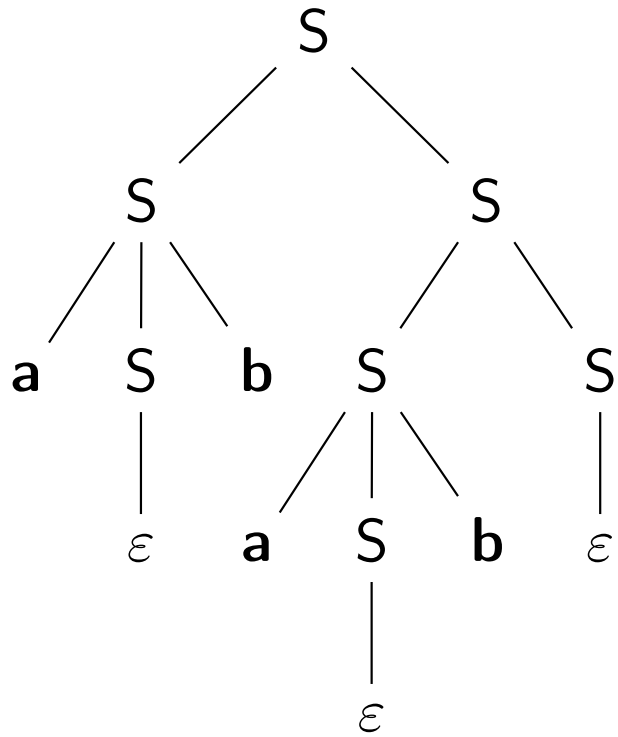
- le mot engendré se lit sur les feuilles de gauche à droite

Exemple d'arbres de dérivation

Retour sur l'exemple des mots bien parenthésés :



Les deux dérivations équivalentes



La troisième dérivation

Valable uniquement pour les grammaires hors-contexte

Interprétation du théorème de décomposition : sous-arbre / sous-forêt

Grammaires HC et définition inductive

Le langage de chaque symbole non terminal X d'une grammaire hors-contexte peut se définir par induction structurelle : chaque règle $\mathsf{X} \to w$ devient :

- un cas de base w si $w \in V_T^*$ (pas de non terminal à droite)
- un constructeur inductif si $w \notin V_T^*$

Exemple (Mots bien parenthésés) : $\mathsf{S} \to \mathsf{SS} \mid \mathbf{aSb} \mid \varepsilon$

$$\begin{array}{lll} \mathsf{S} \to \varepsilon & & b_1 = \varepsilon \\ \mathsf{S} \to \mathsf{SS} & & \kappa_1 : \quad \mathcal{L}(\mathsf{S}) \times \mathcal{L}(\mathsf{S}) \to \mathcal{L}(\mathsf{S}) \\ & \text{devient} & \qquad\quad (u, v) \qquad\quad \mapsto uv \\ \mathsf{S} \to \mathbf{aSb} & & \kappa_2 : \quad \mathcal{L}(\mathsf{S}) \qquad\quad \to \mathcal{L}(\mathsf{S}) \\ & & \qquad\quad u \qquad\quad \mapsto aub \end{array}$$

Conséquence importante : on pourra faire des preuves sur le langage d'une grammaire par induction structurelle

Preuves sur des grammaires

Correction de grammaires

Étant donné une grammaire $G = \langle V_T, V_T, \mathsf{S}, R \rangle$
et un langage L défini par une propriété P : $L = \{w \in V_T^* \mid P(w)\}$,
comment montrer que $\mathcal{L}(G) = L$?

- Correction de G : $\mathcal{L}(G) \subseteq L$

  tous les mots de G satisfont P

- complétude de G : $L \subseteq \mathcal{L}(G)$

  tous les mots de L peuvent être engendrés par G

Méthodes de preuve pour une grammaire HC

- Correction : preuve par induction structurelle

  car $\mathcal{L}(G)$ peut se définir par induction structurelle

  ou par récurrence forte sur la hauteur de l'arbre de dérivation

  ou par récurrence forte sur la longueur de la dérivation

- Complétude : pas de méthode générale

  Touver un découpage des mots de L qui correspond aux règles de G

  $\rightsquigarrow$ en général, récurrence forte sur la longueur du mot

Correction de grammaires HC

Principe

1. Pour chaque non-terminal A de la grammaire, caractériser $\mathcal{L}(A)$ par une propriété P_A : $\mathcal{L}(A) = \{w \in V_T^* \mid P_A(w)\}$
2. Montrer par induction structurelle que :

$$\forall A \in V_N, \forall w \in V_T^*, \quad A \Longrightarrow^* w \quad \Longrightarrow \quad P_A(w)$$

Pour cela, on regarde quelle est la première règle employée et on utilise l'hypothèse d'induction pour les sous-arbres.

Exemple (Mots bien parenthésés)

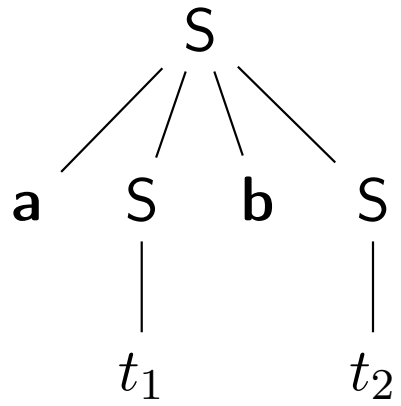
$$S \ \rightarrow \ \mathbf{aSbS} \ \mid \ \varepsilon$$

1. $\mathcal{L}(S) = \{w \in \{a,b\}^* \mid P_1(w) \wedge P_2(w)\}$
   avec $P_1(w) \stackrel{\text{def}}{=} f(w) = 0$
   $P_2(w) \stackrel{\text{def}}{=} \forall u$ préfixe de $w, f(u) \geqslant 0$

$f(x) \stackrel{\text{def}}{=} |x|_a - |x|_b$

Exemple (suite)

2. Montrons par induction structurelle :
   $\forall w \in V_T^*, S \Longrightarrow^* w \quad \Longrightarrow \quad P_1(w) \wedge P_2(w)$



Base $S \Longrightarrow \varepsilon$

donc $w = \varepsilon$ et ε satisfait P_1 et P_2

Induction $S \Longrightarrow \mathbf{aSbS} \Longrightarrow^* w$

$P_1(w)$ D'après le théorème de décomposition, w s'écrit aw_1bw_2.
L'hypothèse d'induction sur t_1 donne $P_1(w_1)$ et $P_2(w_1)$.
L'hypothèse d'induction sur t_2 donne $P_1(w_2)$ et $P_2(w_2)$.
$f(aw_1bw_2) = 1+f(w_1)-1+f(w_2) = f(w_1)+f(w_2) = 0$
donc $P_1(w)$.

$P_2(w)$ Soit u un préfixe de w. On peut avoir :

- $u = \varepsilon$: $f(\varepsilon) = 0 \geqslant 0$
- $u = av$ avec v préfixe de w_1 : $f(u) = 1 + f(v) \geqslant 1 \geqslant 0$
- $u = aw_1bv$ avec v préfixe de w_2 :
  $f(u) = 1 + f(w_1) - 1 + f(v) = f(v) \geqslant 0$

donc $P_2(w)$.

Complétude de grammaires HC

Réciproque : $\forall w, P_1(w) \wedge P_2(w) \implies S \Longrightarrow^* w$

Par récurrence forte sur $|w|$.

Base $|w| = 0$ donne $w = \varepsilon$ et $S \Longrightarrow \varepsilon$

Hérédité $|w| > 0$:

Analysons la forme de w.

La première lettre de w est un a : si c'était un b, on aurait $f(b) = -1 < 0$ et w ne vérifierait pas P_2.

Soit u le plus petit préfixe différent de ε de w tel que $f(u) = 0$ (il existe forcément car $f(w) = 0$).

On a $|u| \geqslant 2$ donc il s'écrit $au'x$ avec $x \in V_T$. En effet, $|u| = 1$ donnerait $u = a$ (car préfixe de w) et $f(u) = 1$, contradiction.

De plus, $x = b$. En effet, $x = a$ donnerait $f(au') = -1$ or c'est un préfixe de w qui vérifie donc $f(au') \geqslant 0$.

Ainsi, $w = au'bv$.

Montrons que u' et v satisfont P_1 et P_2.

Complétude de grammaires HC

Réciproque : $\forall w, P_1(w) \wedge P_2(w) \implies S \Longrightarrow^* w$

Par récurrence forte sur $|w|$.

Base $|w| = 0$ donne $w = \varepsilon$ et $S \Longrightarrow \varepsilon$

Hérédité $|w| > 0$:

Montrons que u' et v satisfont P_1 et P_2. $(w = au'bv)$

- P_1 : $f(u') = f(au'b) = f(u) = 0$
  et $f(v) = f(w) - f(u) = 0$
- $P_2(v)$: Soit x un préfixe de v.
  $f(x) = f(u) + f(x) = f(ux) \geqslant 0$ car ux préfixe de w.
- $P_2(u')$: Soit x un préfixe de u'. $f(ax) > 0$ car c'est un préfixe de u (donc de w d'où $f(ax) \geqslant 0$) et u est le plus petit préfixe de w tel que $f(u) = 0$, d'où $f(x) \geqslant 0$.

Comme $|u'| < |w|$ et $|v| < |w|$, par hypothèse de récurrence sur u' et v, on a $S \Longrightarrow^* u'$ et $S \Longrightarrow^* v$.

On peut alors construire la dérivation suivante pour w :

$S \Longrightarrow \mathbf{aSbS} \Longrightarrow^* \mathbf{a}u'\mathbf{bS} \Longrightarrow^* \mathbf{a}u'\mathbf{b}v = w$

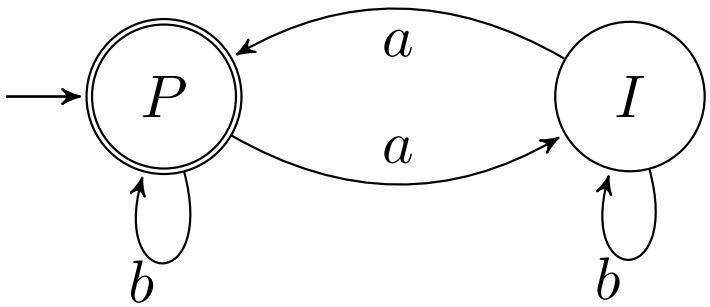
À vous !

Exercice (nombre pair de a)

Soit G définie par P $\to$ **a**I $\mid$ **b**P $\mid$ ε I $\to$ **a**P $\mid$ **b**I.
Soit $L = \{w \in \{a,b\}^* \mid |w|_a$ est pair$\}$.
On souhaite montrer que $\mathcal{L}(G) = L$.

Questions :

1. Comment a-t-on trouvé G ?
2. Montrer sa correction : $\mathcal{L}(G) \subseteq L$.
3. Montrer sa complétude : $L \subseteq \mathcal{L}(G)$.



Propriétés définissant les langages :

- $P_P(w) \stackrel{\text{def}}{=} |w|_a$ est pair
- $P_I(w) \stackrel{\text{def}}{=} |w|_a$ est impair

Exercice : correction de G

On veut montrer que :

$$\begin{aligned} \forall w \in V_T^*, (P \Longrightarrow^* w \quad &\Longrightarrow \quad P_P(w)) \\ \wedge (\, I \Longrightarrow^* w \quad &\Longrightarrow \quad P_I(w)) \end{aligned}$$

Par induction structurelle :

- $P \to \varepsilon$ $|\varepsilon|_A = 0$ qui est pair
- $P \to \mathbf{a}\,\mathsf{I}$ Par théorème de décomposition, $w = aw'$ avec $\mathsf{I} \Longrightarrow^* w'$.
  Par hypothèse d'induction (HI), $P_I(w')$ c.-à-d. $|w'|_a$ est impair.
  Donc $|aw'|_a = 1 + |w'|_a$ est pair, c.-à-d. $P_P(aw')$.
- $P \to \mathbf{b}\,\mathsf{P}$ $|bw'|_a = |w'|_a$ et par HI, $|w'|_a$ est pair.
- $I \to \mathbf{a}\,\mathsf{P}$ Par HI, $|w'|_a$ est pair donc $|aw'|_a = 1 + |w'|_a$ est impair.
- $I \to \mathbf{b}\,\mathsf{I}$ $|bw'|_a = |w'|_a$ et par HI, $|w'|_a$ est impair.

Exercice : complétude de G

On veut montrer que :

$$\begin{aligned} \forall w \in V_T^*, (P_{\mathsf{P}}(w) &\implies \mathsf{P} \Longrightarrow^* w) \\ \wedge (P_{\mathsf{I}}(w) &\implies \mathsf{I} \Longrightarrow^* w) \end{aligned}$$

Par récurrence sur $|w|$:

$|w| = 0$ On a $w = \varepsilon$ donc $P_{\mathsf{P}}(w)$ et $\mathsf{P} \Longrightarrow^* \varepsilon$.

$|w| > 0$ On a $w = xw'$. Distinguons la valeur de x et si w' satisfait P_{P} ou P_{I}.

- $x = a$ et $P_{\mathsf{P}}(w')$: Par hypothèse de récurrence (HR), $\mathsf{P} \Longrightarrow^* w'$.
  Or $P_{\mathsf{I}}(aw')$ donc $\mathsf{I} \Longrightarrow \mathbf{a}\mathsf{P} \Longrightarrow^* aw' = w$
- $x = b$ et $P_{\mathsf{P}}(w')$: Par HR, $\mathsf{P} \Longrightarrow^* w'$.
  Or $P_{\mathsf{P}}(bw')$ donc $\mathsf{P} \Longrightarrow \mathbf{b}\mathsf{P} \Longrightarrow^* aw' = w$
- $P_{\mathsf{I}}(w')$: idem

Lemme d'itération et propriétés de clôture

Lemmes d'itération

- Pour les langages réguliers (lemme de l'étoile) :

$$\forall L, \exists n, \forall z \in L, |z| \geqslant n \implies \exists u, v, w,\ z = uvw \wedge |uv| \leqslant n \wedge |v| \geqslant 1 \\ \wedge \forall i, uv^i w \in L$$

Idée : v est la trace d'un cycle dans l'automate

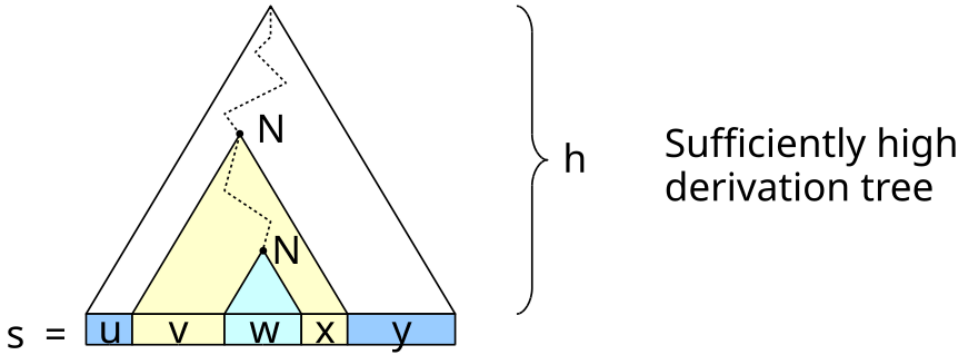
- Pour les langages hors-contexte :

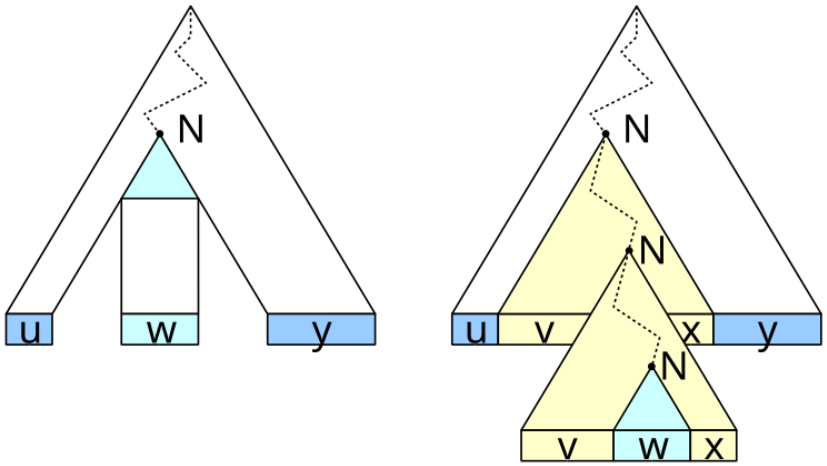
$$\forall L, \exists n, \forall z \in L, |z| \geqslant n \implies \exists u, v, w, x, y,\ z = uvwxy \wedge |vwx| \leqslant n \\ \wedge |vx| \geqslant 1 \wedge \forall i, uv^i wx^i y \in L$$

Idée : On repasse par un même non-terminal (symbole auto-imbriqué) et on répète le fragment de l'arbre entre les deux

Dans les deux cas, ce sont des conditions nécessaires mais non suffisantes.

Illustration





Generating $uv^0wx\,y^0$ Generating $uv^2wx\,y^2$

Source : Wikipédia

Que se passe-t-il si $\mathcal{L}(G)$ est fini ?

Application

Comme pour le lemme de l'étoile pour les langages réguliers.
Exemple : $L = \left\{a^k b^k c^k \mid k \in \mathbb{N}\right\}$ n'est pas hors-contexte.

Supposons par l'absurde que L soit hors-contexte.
Soit n l'entier donné par le lemme d'itération. On pose $z = a^n b^n c^n$. On a bien $z \in L$ et $|w| = 3n \geqslant n$. D'après le lemme d'itération, $z = uvwxy$ avec $|vx| \geqslant 1$, $|vwx| \leqslant n$ et $\forall i, uv^i wx^i y \in L$.
Comme $|vwx| \leqslant n$, on a : $vwx \in a^*b^*$ ou $vwx \in b^*c^*$.
(les cas à une seule lettre sont compris dans ces deux cas)
Choisir $i \geqslant 2$ permet d'ajouter des occurrences d'un ou deux symboles sur les trois, de sorte que $uv^i wx^i y$ ne vérifie plus les égalités $|uv^i wx^i y|_a = |uv^i wx^i y|_b = |uv^i wx^i y|_c$ donc $uv^i wx^i y \notin L$.
C'est absurde donc l'hypothèse faite sur L est fausse : L n'est pas hors-contexte.

Comme pour le lemme de l'étoile,

- on choisit : z, i
- on ne choisit pas : n, u, v, w, x, y

Propriétés de fermeture des langages HC

Les langages réguliers sont fermés (entre autres) par :

- concaténation, itération
- union, intersection, complémentaire

Et pour les langages HC ?

- concaténation : ✓ car $S \to S_1 S_2$
- itération : ✓ car $S \to S_1 S \mid \varepsilon$
- union : ✓ car $S \to S_1 \mid S_2$
- intersection : ✗ car $\{a^n b^n c^n\} = \{a^n b^n c^m\} \cap \{a^m b^n c^n\}$
  mais HC ∩ régulier = HC par produit d'automates
- complémentaire : ✗ sinon $L \cap M = \overline{\overline{L} \cup \overline{M}}$ serait HC

On doit avoir $V_{T_1} = V_{T_2}$, $V_{N_1} \cap V_{N_2} = \emptyset$ et $S \notin V_{N_1} \cup V_{N_2}$