# Analyse LR

2005/2006

## 1 Rappels

### 1.1 Le problème

**Génératif vers recognitif**

Rappels du cours précédent :
- Syntaxe formalisée par une grammaire algébrique $\mathcal{G} = \langle N, \Sigma, P, S\rangle$.
- Construction automatique d'un automate à pile $\mathcal{A} = \langle Q, \Sigma, R, \varphi_s, F, \$, \|\rangle$ pour $\mathcal{G}$.

Une solution : l'automate à pile récursif descendant.

**Analyseur descendant**

**Définition 1.1.** L'*analyseur récursif descendant* pour une grammaire algébrique $\mathcal{G} = \langle N, \Sigma, P, S\rangle$ est un automate à pile $\mathcal{A}_{\text{desc}} = \langle Q, \Sigma, R, \varphi_s, F, \$, \|\rangle$ où
- $Q$ est l'ensemble des production pointées de $P'$, notées

$$[A{\rightarrow}\alpha{\bullet}\alpha'] \text{ si } A{\rightarrow}\alpha\alpha' \in P', \tag{1}$$

- $R$ est l'ensemble des règles

$$[A{\rightarrow}\alpha{\bullet}B\alpha']\| \vdash_{\text{predict}} [A{\rightarrow}\alpha B{\bullet}\alpha'][B{\rightarrow}{\bullet}\beta]\|,$$

$$[A{\rightarrow}\alpha{\bullet}a\alpha']\|a \vdash_{\text{match}} [A{\rightarrow}\alpha a{\bullet}\alpha']\|,$$

$$[A{\rightarrow}\alpha{\bullet}]\| \vdash \|,$$

- $\varphi_s = [S'{\rightarrow}{\bullet}S\$]$,
- $F = \{[S'{\rightarrow}S{\bullet}\$]\}$.

**Analyse descendante**

*Exemple* 1.2.

$$
\begin{aligned}
& \$[S'\to\bullet \text{Ph } \$]\|\text{Det N V Det N\$} \\
\vdash_{\text{predict}} \; & \$[S'\to\text{Ph}\bullet\$][\text{Ph}\to\bullet\text{SN SV}]\|\text{Det N V Det N\$} \\
\vdash_{\text{predict}} \; & \$[S'\to\text{Ph}\bullet\$][\text{Ph}\to\text{SN}\bullet\text{SV}][\text{SN}\to\bullet\text{Det N}]\|\text{Det N V Det N\$} \\
\vdash_{\text{match}} \; & \$[S'\to\text{Ph}\bullet\$][\text{Ph}\to\text{SN}\bullet\text{SV}][\text{SN}\to\text{Det}\bullet\text{N}]\|\text{N V Det N\$} \\
\vdash_{\text{match}} \; & \$[S'\to\text{Ph}\bullet\$][\text{Ph}\to\text{SN}\bullet\text{SV}][\text{SN}\to\text{Det N}\bullet]\|\text{V Det N\$} \\
\vdash \; & \$[S'\to\text{Ph}\bullet\$][\text{Ph}\to\text{SN}\bullet\text{SV}]\|\text{V Det N\$} \\
\vdash_{\text{predict}} \; & \$[S'\to\text{Ph}\bullet\$][\text{Ph}\to\text{SN SV}\bullet][\text{SV}\to\bullet\text{V SN}]\|\text{V Det N\$} \\
\vdash_{\text{match}} \; & \$[S'\to\text{Ph}\bullet\$][\text{Ph}\to\text{SN SV}\bullet][\text{SV}\to\text{V}\bullet\text{SN}]\|\text{Det N\$} \\
\vdash_{\text{predict}} \; & \$[S'\to\text{Ph}\bullet\$][\text{Ph}\to\text{SN SV}\bullet][\text{SV}\to\text{V SN}\bullet][\text{SN}\to\bullet\text{Det N}]\|\text{Det N\$} \\
\vdash_{\text{match}} \; & \$[S'\to\text{Ph}\bullet\$][\text{Ph}\to\text{SN SV}\bullet][\text{SV}\to\text{V SN}\bullet][\text{SN}\to\text{Det}\bullet\text{N}]\|\text{N\$} \\
\vdash_{\text{match}} \; & \$[S'\to\text{Ph}\bullet\$][\text{Ph}\to\text{SN SV}\bullet][\text{SV}\to\text{V SN}\bullet][\text{SN}\to\text{Det N}\bullet]\|\$ \\
\vdash \; & \$[S'\to\text{Ph}\bullet\$][\text{Ph}\to\text{SN SV}\bullet][\text{SV}\to\text{V SN}\bullet]\|\$ \\
\vdash \; & \$[S'\to\text{Ph}\bullet\$][\text{Ph}\to\text{SN SV}\bullet]\|\$ \\
\vdash \; & \$[S'\to\text{Ph}\bullet\$]\|\$
\end{aligned}
$$

**Faiblesses de l'analyse descendante**

1. L'automate à pile est non déterministe ; il nécessite un *backtrack* coûteux ;
2. l'ensemble de règles $R$ permet des règles de la forme $[A\to\bullet A\alpha]\| \vdash_{\text{predict}} [A\to A\bullet\alpha][A\to\bullet A\alpha]\|$ si la grammaire est récursive gauche.

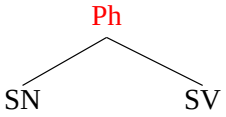
Le non-déterminisme est lié aux conflits entre deux règles *predict*. Quelles autres possibilités s'offrent à nous ?

# 2 Analyse ascendante

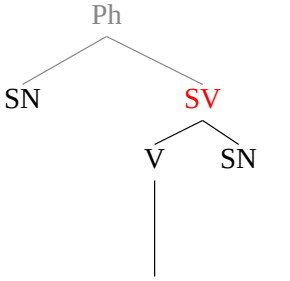
## 2.1 Automate à pile ascendant

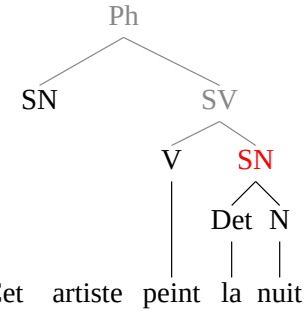
**Dérivations droites**

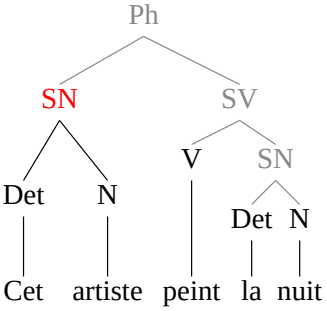
*Exemple* 2.1. Lors d'une dérivation droite :



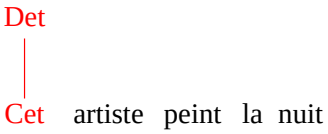

Cet artiste peint la nuit



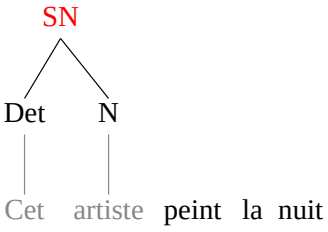



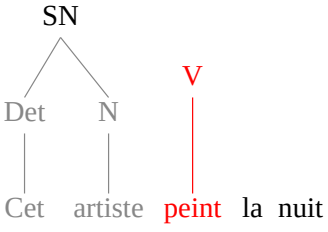



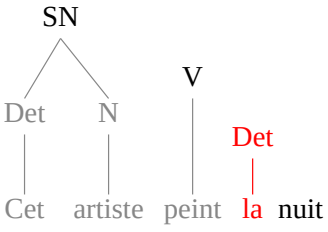

Cet artiste peint la nuit

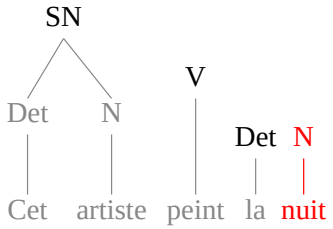

Det
Cet artiste peint la nuit

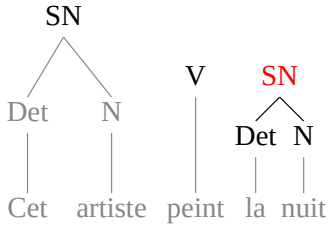

Det N
Cet artiste peint la nuit

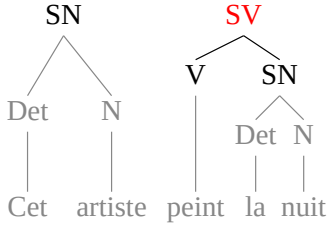

SN
Det N
Cet artiste peint la nuit

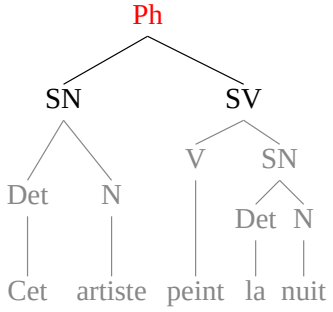

SN
V
Det N
Cet artiste peint la nuit


SN
V
Det
Det N
Cet artiste peint la nuit




Et si on inverse la relation : si on évalue $\underset{\text{rm}}{\Rightarrow}^{-1}$ depuis la chaîne terminale ?

**Analyseur ascendant**

**Définition 2.2.** L'*analyseur ascendant* pour une grammaire algébrique $\mathcal{G} = \langle N, \Sigma, P, S\rangle$ est un automate à pile $\mathcal{A}_{\text{asc}} = \langle V, \Sigma, R, \varphi_s, F, \$, \|\rangle$ où

– $R$ est l'ensemble des règles

$$\alpha\| \vdash_{A\to\alpha} A\|,$$
$$\|a \vdash_{\text{shift}} a\|,$$

– $\varphi_s = \varepsilon$,

– $F = \{S\}$.

**Définition 2.3.** Le *transducteur ascendant* $\langle \mathcal{A}, \tau \rangle$ est défini par

$$\tau(\alpha \| \vdash_{A \to \alpha} A \|) = A \to \alpha,$$
$$\tau(\| a \vdash_{\text{shift}} a \|) = \varepsilon.$$

**Analyse ascendante**

*Exemple* 2.4.

$$\begin{aligned}
& \$ \| \text{Det N V Det N} \$ \\
\models_{\text{shift}} \ & \$ \text{Det} \| \text{N V Det N} \$ \\
\models_{\text{shift}} \ & \$ \text{Det N} \| \text{V Det N} \$ \\
\models_{\text{SN} \to \text{Det N}} \ & \$ \text{SN} \| \text{V Det N} \$ \\
\models_{\text{shift}} \ & \$ \text{SN V} \| \text{Det N} \$ \\
\models_{\text{shift}} \ & \$ \text{SN V Det} \| \text{N} \$ \\
\models_{\text{shift}} \ & \$ \text{SN V Det N} \| \$ \\
\models_{\text{SN} \to \text{Det N}} \ & \$ \text{SN V SN} \| \$ \\
\models_{\text{SV} \to \text{V SN}} \ & \$ \text{SN SV} \| \$ \\
\models_{\text{Ph} \to \text{SN SV}} \ & \$ \text{Ph} \| \$
\end{aligned}$$

**Exercice**

*Exercice* 2.5. Soit la grammaire

$$E \to E + T \,|\, T, \ T \to T * F \,|\, F, \ F \to a \,|\, (E).$$

Donnez les étapes de l'analyse de la phrase $a + a * a$ par un automate à pile ascendant.

**Un amélioration ?**

– L'automate ascendant est lui aussi fortement non-déterministe, avec des conflits
  – *shift/reduce* dans une configuration $\alpha \| a$ si la production $A \to \alpha$ existe,
  – *reduce/reduce* dans une configuration $\beta\alpha \|$ si les productions distinctes $A \to \alpha$ et $B \to \beta\alpha$ existent.
– En revanche, il n'a pas de problème avec la récursivité d'une grammaire acyclique.

## 2.2 Contexte

**Exercice**

*Exercice* 2.6. Soit la grammaire

$$S \to A \,|\, B, \ A \to cA \,|\, a, \ B \to cB \,|\, b.$$

Montrez que l'automate à pile ascendant pour cette grammaire est déterministe. Que dire de l'automate descendant ?

La décision de reconnaître une production de la grammaire est faite juste avant de reconnaître les constituants de cette production en analyse descendante, et une fois ces constituants reconnus en analyse descendante.

**Poignées**

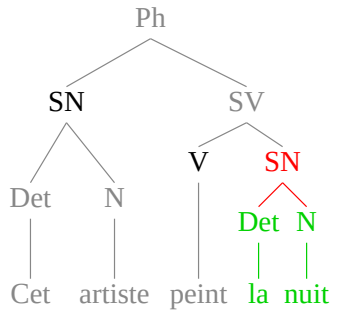
**Définition 2.7.** Une chaîne $\alpha$ est un *syntagme* (*phrase* en anglais) de la forme sententielle $\varphi\alpha\sigma$ si il existe $A \rightarrow \alpha$ dans $P$ et si $S \Rightarrow^* \varphi A \sigma \Rightarrow \varphi\alpha\sigma$.

En d'autres termes, la réduction d'$\alpha$ à $A$ dans $\varphi\alpha\sigma$ nous donne bien une forme sententielle.

**Définition 2.8.** Un syntagme d'une forme sententielle droite dont la réduction donne à nouveau une forme sententielle droite est appelé une *poignée* (*handle* en anglais) : $\alpha$ est une poignée dans la forme sententielle $\delta\alpha x$ si il existe $A \rightarrow \alpha$ dans $P$ et si $S \underset{\text{rm}}{\Rightarrow}^* \delta A x \underset{\text{rm}}{\Rightarrow} \delta\alpha x$.

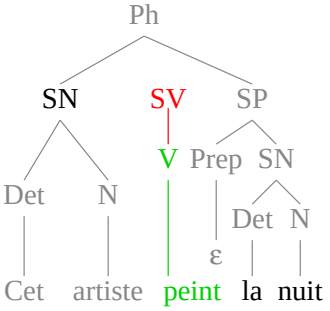
**Poignées**

*Exemple* 2.9.




$$\text{Ph} \underset{\text{rm}}{\Rightarrow}^* \text{SN V SN} \underset{\text{rm}}{\Rightarrow} \text{SN V Det N} \tag{2}$$

*Exemple* 2.10.




$$\text{Ph} \underset{\text{rm}}{\Rightarrow}^* \text{SN SV Det N} \underset{\text{rm}}{\Rightarrow} \text{SN V Det N} \tag{3}$$

La poignée d'une forme sententielle droite est unique pour une grammaire non ambiguë.

**Contexte d'une poignée**

**Définition 2.11.** Soit $\alpha$ une poignée de la forme sententielle $\delta\alpha x$ ; on appelle $\delta\alpha$ son *contexte gauche* et $x$ son *contexte droit*.

De nombreuses méthodes d'analyse ascendante déterministe font jouer des relations entre symboles de la fin du contexte gauche et symboles du début du contexte droit :
- analyse par *précédence* de symboles [Flo63, WW66, IM70],
- analyse considérant une portion *bornée* des contextes gauches et droits [Flo64].

Elles sont tombées dans l'oubli face à LR...

# 3 Analyse LR

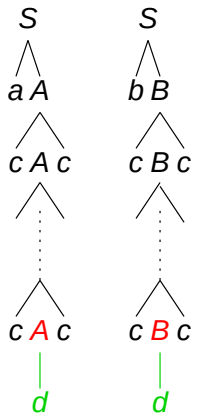
## 3.1 Grammaire caractéristique

Le principe fondateur de LR est d'exploiter la totalité du contexte gauche des poignées. Nous commençons par donner un autre point de vue sur ce contexte gauche.

### Préfixe viable

*Exemple* 3.1.

$$S{\to}aA\,|\,bB,\ A{\to}cAc\,|\,d,\ B{\to}cBc\,|\,d.$$




La séquence $ac^nd$ est un préfixe viable complet de la grammaire :

*Définition* 3.2. Si $S \underset{\text{rm}}{\Rightarrow}{}^{*} \delta Ax \underset{\text{rm}}{\Rightarrow} \delta\alpha\beta x$ dans $\mathcal{G}$, alors $\delta\alpha$ est un *préfixe viable* de $\mathcal{G}$, et est *complet* si $\beta = \varepsilon$.

L'intérêt de cette notion de préfixe viable est qu'elle est reflétée par une notion similaire de pile viable de l'analyseur ascendant.

### Pile viable

**Définition 3.3.** La chaîne $\varphi$ de $Q^*$ est une *pile viable* pour l'automate à pile $\mathcal{A} = \langle Q, \Sigma, R, \varphi_s, F, \$, \| \rangle$ si $\$\varphi_s \| w\$ \models^* \$\varphi \| x\$ \models^* \$\varphi_f \|\$$ pour $\varphi_f$ dans $F$ et $w$ et $x$ dans $\Sigma^*$.

**Théorème 3.4.** *Soit $\mathcal{G}$ une grammaire et $\mathcal{A}_{asc}$ son automate à pile ascendant. Tout préfixe viable de $\mathcal{G}$ est une pile viable de $\mathcal{A}_{asc}$ et réciproquement.*

Toute la force de la méthode LR réside dans le fait que les préfixes viables d'une grammaire forment un langage régulier, et qu'ils peuvent donc être reconnus par un automate à états finis déterministe.

### Grammaire caractéristique [Knu65]

**Définition 3.5.** Soit une grammaire $\mathcal{G} = \langle N, \Sigma, P, S \rangle$. La *grammaire caractéristique* de $\mathcal{G}$ est $\mathcal{G}_0 = \langle N_0, V, P_0, [S] \rangle$ où
- $N_0 = \{[A] \mid A \in N\}$ et
- $P_0 = \{[A]{\to}\alpha \mid A{\to}\alpha \in P\} \cup \{[A]{\to}\alpha[B] \mid A{\to}\alpha B\beta \in P\}$.

*Exemple* 3.6. La grammaire caractéristique pour

$$S{\to}aA\,|\,bB,\ A{\to}cAc\,|\,d,\ B{\to}cBc\,|\,d$$

est

$$[S]{\to}a[A]\,|\,aA\,|\,b[B]\,|\,bB,\ [A]{\to}c[A]\,|\,cAc\,|\,d,\ [B]{\to}c[B]\,|\,cBc\,|\,d.$$

La grammaire caractéristique d'une grammaire $\mathcal{G}$ génère tous les préfixes viables complets de $\mathcal{G}$.

La grammaire caractéristique est une grammaire linéaire à droite ; le langage des préfixes viables complet est donc un langage rationnel. L'exercice suivant est l'occasion d'exploiter cette propriété.

**Exercice**

*Exercice* 3.7. Donner un automate à états finis déterministe qui reconnaît les phrases du générées par la grammaire caractéristique

$$[S]{\to}a[A]\,|\,aA\,|\,b[B]\,|\,bB,\ [A]{\to}c[A]\,|\,cAc\,|\,d,\ B{\to}c[B]\,|\,cBc\,|\,d.$$

## 3.2 Items LR($k$)

Cette construction de l'automate à états finis est formalisée par la notion d'items LR. Nous utilisons une notion plus forte que le simple préfixe viable complet : nous souhaitons connaître exactement quelles chaînes de longueur $k$ sont acceptables après ce préfixe.

### Item LR($k$) valide [Knu65, AU72]

**Définition 3.8.** Un *$k$-item* est un couple de la forme $[A{\to}\alpha\bullet\beta, u]$ où $A{\to}\alpha\beta$ est une production de $P$ et $u$ une chaîne de $\Sigma^k$. Il est *LR(k)-valide* pour un préfixe $\gamma$ si

$$S'\underset{\text{rm}}{\Rightarrow}{}^*\delta Ax\underset{\text{rm}}{\Rightarrow}\delta\alpha\beta x=\gamma\beta x \text{ et } k:x=u. \tag{4}$$

Si $\gamma$ est un préfixe viable de $\mathcal{G}$, alors il existe un item valide pour $\gamma$, et réciproquement.

*Exemple* 3.9. Les items $[A{\to}c\bullet Ac, c]$ et $[A{\to}\bullet d, c]$ sont valides pour $ac^n$, $n>1$ pour la grammaire

$$S{\to}aA\,|\,bB,\ A{\to}cAc\,|\,d,\ B{\to}cBc\,|\,d\ :$$

$$S'\underset{\text{rm}}{\Rightarrow}{}^*ac^{n-1}Ac^{n-1}\$\underset{\text{rm}}{\Rightarrow}ac^nAc^n\$\underset{\text{rm}}{\Rightarrow}ac^ndc^n\$ \text{ et } 1:c^n=c. \tag{5}$$

L'ensemble des $k$-items valides pour $\gamma$ est noté

$$\text{Valid}_k(\gamma)=\{\iota\mid\iota \text{ est un } k\text{-item valide pour } \gamma\}. \tag{6}$$

### Relations entre items LR($k$)

**Lemme 3.10.** *Si $S{\to}\sigma$ est une règle de $P$, alors $[S{\to}\bullet\sigma, \$^k]$ est un $k$-item valide pour $\varepsilon$.*

**Lemme 3.11.** *Si $[A{\to}\alpha\bullet X\beta, u]$ est un $k$-item valide pour $\gamma$, alors $[A{\to}\alpha X\bullet\beta, u]$ est un $k$-item valide pour $\gamma X$. On note $[A{\to}\alpha\bullet X\beta, u]\overset{X}{\curvearrowright}[A{\to}\alpha X\bullet\beta, u]$.*

**Lemme 3.12.** *Si $[A{\to}\alpha\bullet B\beta, u]$ est un $k$-item valide pour $\gamma$, alors $[B{\to}\bullet\delta, \mathit{First}_k(\beta u)]$ est un $k$-item valide pour $\gamma$. On note $[A{\to}\alpha\bullet B\beta, u]\overset{\varepsilon}{\curvearrowright}[B{\to}\bullet\delta, \mathit{First}_k(\beta u)]$.*

**Théorème 3.13.**

$$\mathit{Valid}_k(\gamma)=\{\iota\mid[S{\to}\bullet\sigma, \$^k]\overset{\gamma}{\curvearrowright}\iota\}. \tag{7}$$

**Exercice**

*Exercice* 3.14. Donnez les ensembles $\text{Valid}_k$ pour la grammaire

$$S{\to}aA\,|\,bB,\ A{\to}cAc\,|\,d,\ B{\to}cBc\,|\,d$$

### Équivalence LR($k$)

**Définition 3.15.** La chaîne $\gamma_1$ est *LR(k)-équivalente* à la chaîne $\gamma_2$, dénoté par $\gamma_1\equiv_{\text{LR}(k)}\gamma_2$, si et seulement si $\text{Valid}_k(\gamma_1)=\text{Valid}_k(\gamma_2)$.

**Théorème 3.16.** *Pour toute grammaire $\mathcal{G}=\langle N,\Sigma,P,S\rangle$ et entier naturel $k$, la relation $\equiv_{LR(k)}$ est une relation d'équivalence sur $V^*$. De plus, $\equiv_{LR(k)}$ a un index fini et borné par*

$$2^{|\mathcal{G}||\Sigma|^k}. \tag{8}$$

*Exercice* 3.17. Prouver le théorème précédent.

*Exercice* 3.18. Reformulez la définition d'une grammaire caractéristique pour utiliser la fenêtre de $k$ symboles.

## 3.3 Automate LR($k$)

L'équivalence LR($k$) est l'équivalence induite par l'automate d'états finis reconnaissant $\mathcal{G}_k$. On note les classes d'équivalence de $\equiv_{\mathrm{LR}(k)}$ par $[\gamma]_k$ où $\gamma$ est un élément de la classe d'équivalence, et on note l'ensemble des classes d'équivalence $E_k$.

### Automate LR($k$)

**Définition 3.19.** L'*automate LR(k)* de la grammaire $\mathcal{G}$ est l'automate à pile ascendant $\mathcal{A}_{\mathrm{LR}(k)} = \langle E_k, \Sigma, R, [\varepsilon], \{[S]\}, \$, \| \rangle$ où

- $E_k$ est l'ensemble des classes d'équivalences $[\delta]_k$ de $\equiv_{\mathrm{LR}(k)}$,
- l'ensemble des règles $R$ est restreint aux règles

$$[\delta]_k[\delta X_1]_k \dots [\delta X_1 \dots X_n]_k \| u \vdash_{A \to X_1 \dots X_n} [\delta]_k[\delta A]_k \| u, \tag{9}$$

si un item $[A \to X_1 \dots X_n \bullet, u]$ est LR($k$)-valide pour $\delta$, et

$$[\delta]_k \| au \vdash_{\mathrm{shift}} [\delta]_k[\delta a]_k \| u, \tag{10}$$

si un item $[A \to \alpha \bullet a\beta, v]$ est LR($k$)-valide pour $\delta$ et si $u \in \mathrm{First}_{k-1}(\beta v)$.

### Analyse ascendante LR

*Exemple* 3.20. Soit la grammaire

$$S \to aA \mid bB,\ A \to cAc \mid d,\ B \to cBc \mid d\ :$$

$$\begin{aligned}
&\$[\varepsilon]\|accdcc\$ \\
\vdash_{\mathrm{shift}} &\$[\varepsilon][a]\|ccdcc\$ \\
\vdash_{\mathrm{shift}} &\$[\varepsilon][a][ac]\|cdcc\$ \\
\vdash_{\mathrm{shift}} &\$[\varepsilon][a][ac][acc]\|dcc\$ \\
\vdash_{\mathrm{shift}} &\$[\varepsilon][a][ac][acc][accd]\|cc\$ \\
\vdash_{\mathrm{shift}} &\$[\varepsilon][a][ac][acc][accd]\|cc\$ \\
\vdash_{A \to d} &\$[\varepsilon][a][ac][acc][accA]\|cc\$ \\
\vdash_{\mathrm{shift}} &\$[\varepsilon][a][ac][acc][accA][accAc]\|c\$ \\
\vdash_{\mathrm{shift}} &\$[\varepsilon][a][ac][acc][accA][accAc]\|c\$ \\
\vdash_{A \to cAc} &\$[\varepsilon][a][ac][acA]\|c\$ \\
\vdash_{\mathrm{shift}} &\$[\varepsilon][a][ac][acA][acAc]\|\$ \\
\vdash_{\mathrm{shift}} &\$[\varepsilon][a][ac][acA][acAc]\|\$ \\
\vdash_{A \to cAc} &\$[\varepsilon][a][aA]\|\$ \\
\vdash_{A \to cAc} &\$[\varepsilon][a][aA]\|\$ \\
\vdash_{S \to aA} &\$[\varepsilon][S]\|\$
\end{aligned}$$

### Grammaires LR($k$)

**Définition 3.21.** Une grammaire est LR($k$) si

1. $S \underset{\mathrm{rm}}{\Rightarrow}{}^* \delta A x \underset{\mathrm{rm}}{\Rightarrow} \delta\alpha x$,
2. $S \underset{\mathrm{rm}}{\Rightarrow}{}^* \gamma B y \underset{\mathrm{rm}}{\Rightarrow} \gamma\beta z y = \delta\alpha z y$ et
3. $k : x = k : zy$

impliquent $\delta A x = \gamma B y$.

**Théorème 3.22.** *Une grammaire $\mathcal{G}$ est LR(k) si et seulement si son automate LR(k) est déterministe.*

**Théorème 3.23.** *Si une grammaire $\mathcal{G}$ est LR(k), alors il existe une grammaire $\mathcal{G}'$ LR(1) équivalente.*

**Langages déterministes**

**Théorème 3.24.** *Les langages générés par les grammaires LR(1) sont exactement les langages reconnus par un automate à pile déterministe. On appelle ces langages les* langages déterministes.

# References


[AU72] Alfred V. Aho and Jeffrey D. Ullman. *The Theory of Parsing, Translation, and Compiling*, volume I: Parsing of *Series in Automatic Computation*. Prentice Hall, Englewood Cliffs, New Jersey, 1972.

[Flo63] Robert W. Floyd. Syntactic analysis and operator precedence. *Journal of the ACM*, 10(3):316–333, 1963.

[Flo64] Robert W. Floyd. Bounded context syntactic analysis. *Communications of the ACM*, 7(2):62–67, 1964.

[IM70] J. D. Ichbiah and S. P. Morse. A technique for generating almost optimal Floyd-Evans productions for precedence grammars. *Communications of the ACM*, 13(8):501–508, 1970.

[Knu65] Donald E. Knuth. On the translation of languages from left to right. *Information and Control*, 8:607–639, 1965.

[WW66] Niklaus Wirth and Helmut Weber. EULER: a generalization of ALGOL and it formal definition. *Communications of the ACM*, 9(1):Part I: 13–25, and Part II: 89–99, 1966.