Deuxième partie II

Analyse LR

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 $\mathcal{G}$.

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 $\mathcal{G}$.

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 $\mathcal{G}$.

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

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
$$[A \rightarrow \alpha \bullet \alpha'] \text{ si } A \rightarrow \alpha\alpha' \in P',$$
- R est l'ensemble des règles
$$[A \rightarrow \alpha \bullet B\alpha']\| \vdash_{\text{predict}} [A \rightarrow \alpha B \bullet \alpha'][B \rightarrow \bullet \beta]\|,$$
$$[A \rightarrow \alpha \bullet a\alpha']\|a \vdash_{\text{match}} [A \rightarrow \alpha a \bullet \alpha']\|,$$
$$[A \rightarrow \alpha \bullet]\| \vdash \|,$$
- $\varphi_s = [S' \rightarrow \bullet S\$]$,
- $F = \{[S' \rightarrow S \bullet \$]\}$.

Analyseur descendant

Définition

L'analyseur récursif descendant pour une grammaire algébrique $\mathcal{G} = \langle N, \Sigma, P, S \rangle$ est un automate à pile $\mathcal{A}_{\mathsf{desc}} = \langle Q, \Sigma, R, \varphi_s, F, \$, \| \rangle$ où

- Q est l'ensemble des production pointées de P', notées
$$[A \to \alpha \bullet \alpha'] \text{ si } A \to \alpha\alpha' \in P',$$
- R est l'ensemble des règles
$$[A \to \alpha \bullet B\alpha'] \| \vdash_{\mathsf{predict}} [A \to \alpha B \bullet \alpha'][B \to \bullet \beta] \|,$$
$$[A \to \alpha \bullet a\alpha'] \| a \vdash_{\mathsf{match}} [A \to \alpha a \bullet \alpha'] \|,$$
$$[A \to \alpha \bullet] \| \vdash \|,$$
- $\varphi_s = [S' \to \bullet S\$]$,
- $F = \{[S' \to S \bullet \$]\}$.

Analyseur descendant

Définition

L'analyseur récursif descendant pour une grammaire algébrique $\mathcal{G} = \langle N, \Sigma, P, S \rangle$ est un automate à pile $\mathcal{A}_{\mathsf{desc}} = \langle Q, \Sigma, R, \varphi_s, F, \$, \| \rangle$ où

- Q est l'ensemble des production pointées de P', notées
$$[A \rightarrow \alpha \bullet \alpha'] \text{ si } A \rightarrow \alpha\alpha' \in P',$$
- R est l'ensemble des règles
$$[A \rightarrow \alpha \bullet B\alpha'] \| \vdash_{\mathsf{predict}} [A \rightarrow \alpha B \bullet \alpha'][B \rightarrow \bullet \beta] \|,$$
$$[A \rightarrow \alpha \bullet a\alpha'] \| a \vdash_{\mathsf{match}} [A \rightarrow \alpha a \bullet \alpha'] \|,$$
$$[A \rightarrow \alpha \bullet] \| \vdash \|,$$
- $\varphi_s = [S' \rightarrow \bullet S\$]$,
- $F = \{[S' \rightarrow S \bullet \$]\}$.

Analyseur descendant

Définition

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
$$[A \rightarrow \alpha \bullet \alpha'] \text{ si } A \rightarrow \alpha\alpha' \in P',$$
- R est l'ensemble des règles
$$[A \rightarrow \alpha \bullet B\alpha'] \| \vdash_{\text{predict}} [A \rightarrow \alpha B \bullet \alpha'][B \rightarrow \bullet \beta] \|,$$
$$[A \rightarrow \alpha \bullet a\alpha'] \| a \vdash_{\text{match}} [A \rightarrow \alpha a \bullet \alpha'] \|,$$
$$[A \rightarrow \alpha \bullet] \| \vdash \|,$$
- $\varphi_s = [S' \rightarrow \bullet S\$]$,
- $F = \{[S' \rightarrow S \bullet \$]\}$.

Analyseur descendant

Définition

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

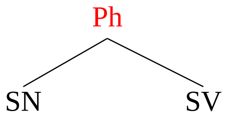
Ph

Cet artiste peint la nuit

$\$[S' \rightarrow \bullet \text{Ph } \$] \| \text{Det N V Det N}\$$

Analyse descendante

Exemple

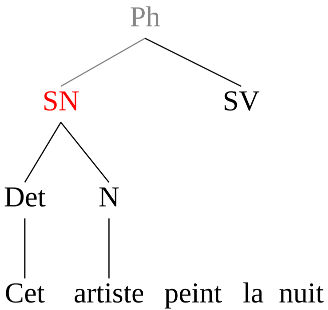


Cet artiste peint la nuit

$$\models_{\text{predict}} \$[S' \rightarrow \text{Ph} \bullet \$][\text{Ph} \rightarrow \bullet \text{SN SV}] \| \text{Det N V Det N}\$$$

Analyse descendante

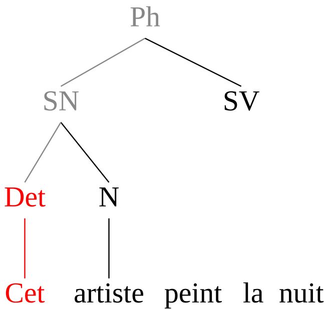
Exemple



$$\models_{\text{predict}} \$[S'\rightarrow \text{Ph}\bullet\$][\text{Ph}\rightarrow \text{SN}\bullet\text{SV}][\text{SN}\rightarrow\bullet\text{Det N}]\|\text{Det N V Det N}\$$$

Analyse descendante

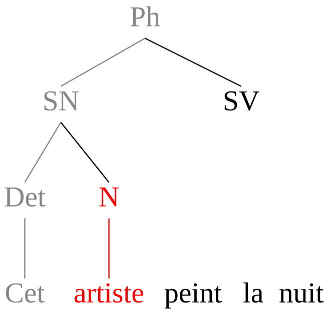
Exemple



$$\models_{\mathrm{match}} \$[S' \rightarrow \mathrm{Ph} \bullet \$][\mathrm{Ph} \rightarrow \mathrm{SN} \bullet \mathrm{SV}][\mathrm{SN} \rightarrow \mathrm{Det} \bullet \mathrm{N}] \| \mathrm{N}\ \mathrm{V}\ \mathrm{Det}\ \mathrm{N}\$$$

Analyse descendante

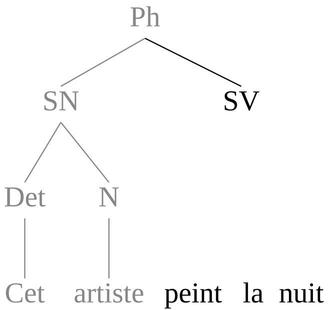
Exemple



$\models_{\text{match}}$ $[S' \rightarrow \text{Ph} \bullet \$][\text{Ph} \rightarrow \text{SN} \bullet \text{SV}][\text{SN} \rightarrow \text{Det N} \bullet] \| \text{V Det N}\$$

Analyse descendante

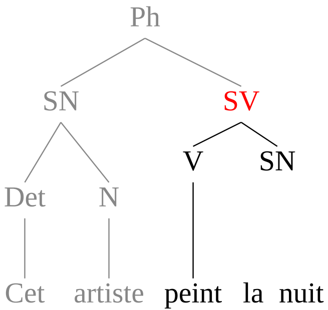
Exemple



$$\models \$[S'\rightarrow \mathrm{Ph}\bullet\$][\mathrm{Ph}\rightarrow \mathrm{SN}\bullet \mathrm{SV}]\|\mathrm{V}\ \mathrm{Det}\ \mathrm{N}\$$$

Analyse descendante

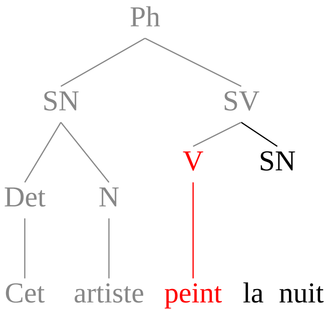
Exemple



$$\models_{\text{predict}} \$[S'\rightarrow \text{Ph}\bullet\$][\text{Ph}\rightarrow\text{SN SV}\bullet][\text{SV}\rightarrow\bullet\text{V SN}]\|\text{V Det N}\$$$

Analyse descendante

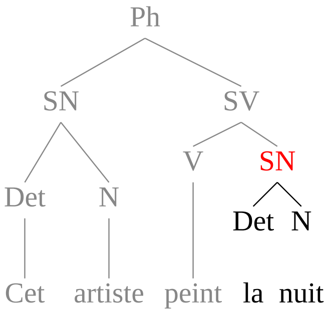
Exemple



$$\models_{\text{match}} \$[S' \rightarrow \text{Ph} \bullet \$][\text{Ph} \rightarrow \text{SN SV} \bullet][\text{SV} \rightarrow \text{V} \bullet \text{SN}] \| \text{Det N} \$$$

Analyse descendante

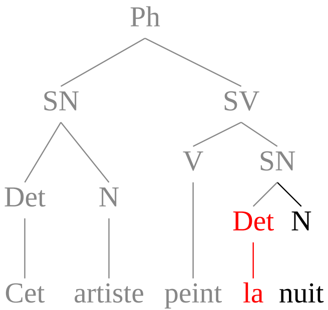
Exemple



$$\models_{\text{predict}} \$[S' \rightarrow \text{Ph} \bullet \$][\text{Ph} \rightarrow \text{SN SV} \bullet][\text{SV} \rightarrow \text{V } \textcolor{red}{\text{SN}} \bullet][\text{SN} \rightarrow \bullet \text{Det N}] \| \text{Det N}\$$$

Analyse descendante

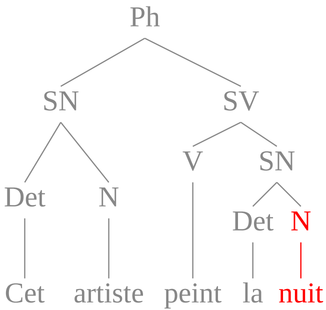
Exemple



$$\models_{\text{match}} \$[S' \to \text{Ph} \bullet \$][\text{Ph} \to \text{SN SV} \bullet][\text{SV} \to \text{V SN} \bullet][\text{SN} \to \text{Det} \bullet \text{N}] \| \text{N}\$$$

Analyse descendante

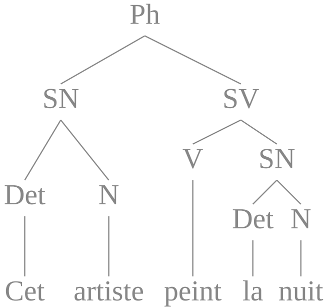
Exemple



$$\models_{\text{match}} \$[S'\rightarrow \mathsf{Ph}\bullet\$][\mathsf{Ph}\rightarrow \mathsf{SN}\ \mathsf{SV}\bullet][\mathsf{SV}\rightarrow \mathsf{V}\ \mathsf{SN}\bullet][\mathsf{SN}\rightarrow \mathsf{Det}\ \mathsf{N}\bullet]\|\$$$

Analyse descendante

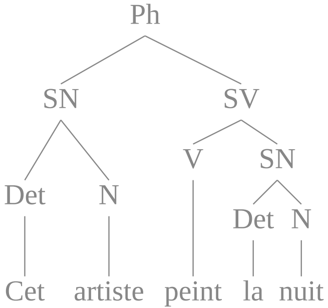
Exemple



$$\models \$[S'\rightarrow \mathrm{Ph}\bullet\$][\mathrm{Ph}\rightarrow \mathrm{SN}\ \mathrm{SV}\bullet][\mathrm{SV}\rightarrow \mathrm{V}\ \mathrm{SN}\bullet]\|\$$$

Analyse descendante

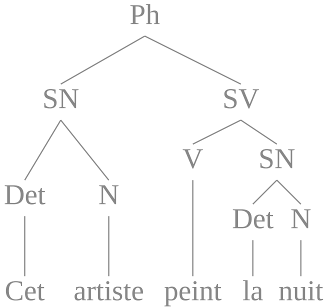
Exemple



$$\models \$[S'\rightarrow \mathrm{Ph}\bullet\$][\mathrm{Ph}\rightarrow \mathrm{SN}\ \mathrm{SV}\bullet]\|\$$$

Analyse descendante

Exemple



$$\models \$[S'\to \mathrm{Ph}\bullet\$]\|\$$$

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.

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 $[A \rightarrow \bullet A\alpha] \| \vdash_{\text{predict}} [A \rightarrow A \bullet \alpha][A \rightarrow \bullet A\alpha] \|$ si la grammaire est récursive gauche.

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 $[A \rightarrow \bullet A\alpha] \| \vdash_{\text{predict}} [A \rightarrow A \bullet \alpha][A \rightarrow \bullet A\alpha] \|$ si la grammaire est récursive gauche.

Le non-déterminisme est lié aux conflits entre deux règles *predict*.

Faiblesses de l'analyse descendante

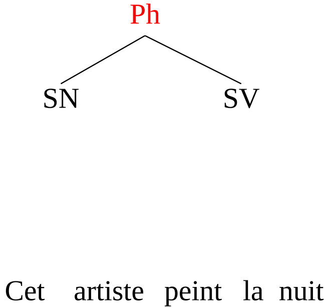
1. L'automate à pile est non déterministe ; il nécessite un *backtrack* coûteux ;
2. l'ensemble de règles R permet des règles de la forme $[A \rightarrow \bullet A\alpha] \| \vdash_{\text{predict}} [A \rightarrow A \bullet \alpha][A \rightarrow \bullet A\alpha] \|$ si la grammaire est récursive gauche.

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

Dérivations droites

Exemple

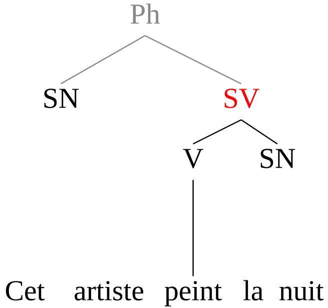
Lors d'une dérivation droite :



Dérivations droites

Exemple

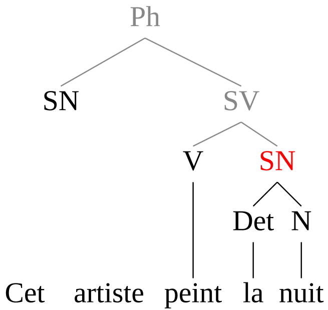
Lors d'une dérivation droite :



Dérivations droites

Exemple

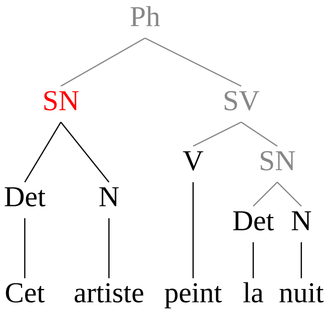
Lors d'une dérivation droite :



Dérivations droites

Exemple

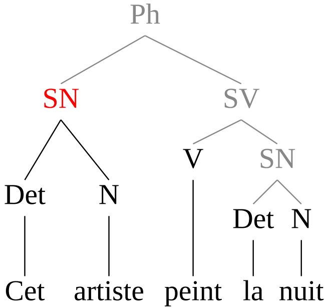
Lors d'une dérivation droite :



Dérivations droites

Exemple

Lors d'une dérivation droite :



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

Dérivations droites

Exemple

Cet artiste peint la nuit

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

Dérivations droites

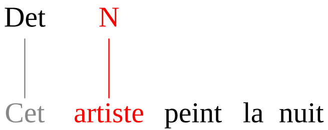
Exemple

$$\begin{array}{ccccc} \textcolor{red}{\text{Det}} & & & & \\ \textcolor{red}{|} & & & & \\ \textcolor{red}{\text{Cet}} & \text{artiste} & \text{peint} & \text{la} & \text{nuit} \end{array}$$

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

Dérivations droites

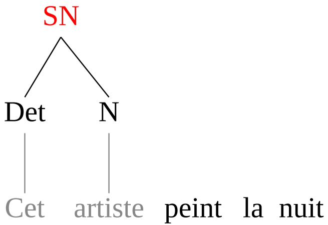
Exemple



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

Dérivations droites

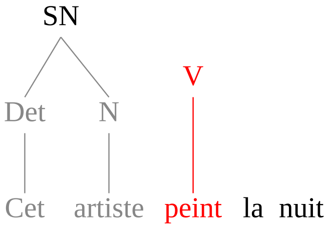
Exemple



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

Dérivations droites

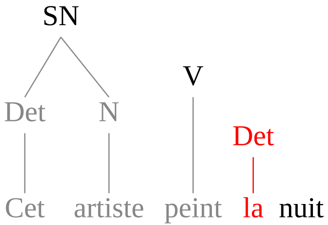
Exemple



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

Dérivations droites

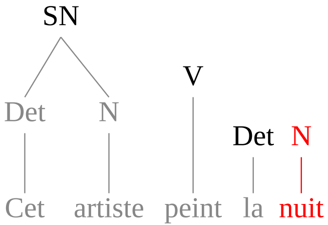
Exemple



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

Dérivations droites

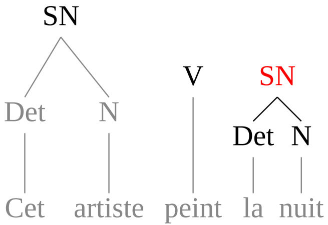
Exemple



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

Dérivations droites

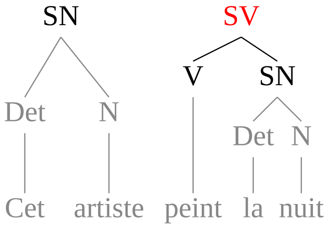
Exemple



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

Dérivations droites

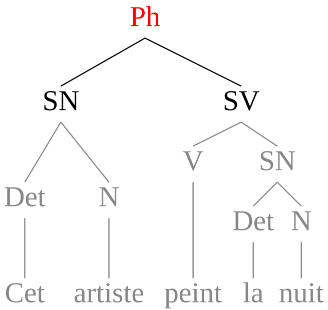
Exemple



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

Dérivations droites

Exemple



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

Analyseur ascendant

Définition

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

- R est l'ensemble des règles

$$\alpha\| \underset{A \to \alpha}{\vdash} A\|,$$
$$\|a \underset{\mathsf{shift}}{\vdash} a\|,$$

- $\varphi_s = \varepsilon$,
- $F = \{S\}$.

Analyseur ascendant

Définition

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

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

- $\varphi_s = \varepsilon$,
- $F = \{S\}$.

Analyseur ascendant

Définition

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

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

- $\varphi_s = \varepsilon$,
- $F = \{S\}$.

Analyseur ascendant

Définition

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

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

- $\varphi_s = \varepsilon$,
- $F = \{S\}$.

Analyseur ascendant

Définition

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

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

Cet artiste peint la nuit

$||Det N V Det N$

Analyse ascendante

Exemple

Det

Cet artiste peint la nuit

$$\models_{\text{shift}} \$\text{Det}\|\text{N V Det N}\$$$

Analyse ascendante

Exemple

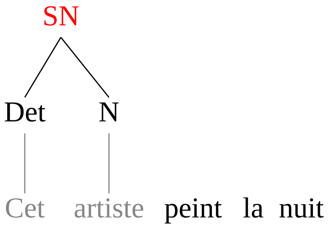
Det N

Cet artiste peint la nuit

$$\models_{\text{shift}} \$\text{Det N} \| \text{V Det N}\$$$

Analyse ascendante

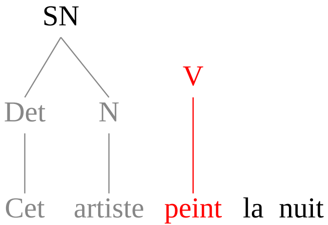
Exemple



$$\models_{\mathsf{SN}\rightarrow\mathsf{Det\ N}} \$\mathsf{SN}\|\mathsf{V\ Det\ N}\$$$

Analyse ascendante

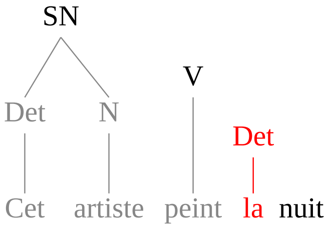
Exemple



$\models_{\text{shift}}$ \$SN V||Det N\$

Analyse ascendante

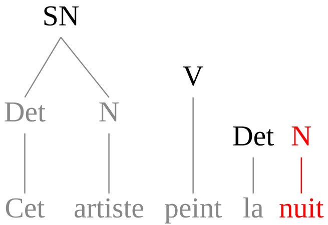
Exemple



$\models_{\text{shift}}$ \$SN V Det||N\$

Analyse ascendante

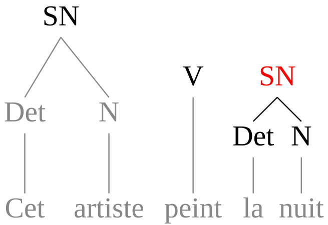
Exemple



$\models_{\text{shift}}$ \$SN V Det N||\$

Analyse ascendante

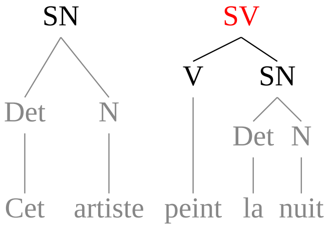
Exemple



$$\models_{SN \rightarrow Det\ N} \$SN\ V\ SN \| \$$$

Analyse ascendante

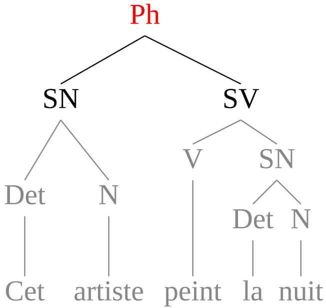
Exemple



$$\models_{\text{SV}\rightarrow\text{V SN}} \$\text{SN SV}\|\$$$

Analyse ascendante

Exemple



$$\models_{\mathsf{Ph} \to \mathsf{SN\ SV}} \$\mathsf{Ph} \| \$$$

Exercice

Exercice

Soit la grammaire

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

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

Exercice

Exercice

Soit la grammaire

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

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

Exercice

Exercice

Soit la grammaire

$$E \rightarrow E + T \mid T, \; T \rightarrow T * F \mid F, \; F \rightarrow a \mid (E).$$

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

$$\$ \| a + a * a + a \$$$

Exercice

Exercice

Soit la grammaire

$$E \rightarrow E + T \mid T, \;\; T \rightarrow T * F \mid F, \;\; F \rightarrow a \mid (E).$$

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

$$\begin{aligned} & \$\| a + a * a + a\$ \\ \models_{\text{shift}} \; & \$a \| + a * a\$ \end{aligned}$$

Exercice

Exercice

Soit la grammaire

$$E \rightarrow E + T \mid T, \;\; T \rightarrow T * F \mid F, \;\; F \rightarrow a \mid (E).$$

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

$$\begin{aligned} & \$\| a + a * a + a\$ \\ \models_{\text{shift}} \; & \$a \| + a * a\$ \\ \models_{F \rightarrow a} \; & \$F \| + a * a\$ \end{aligned}$$

Exercice

Exercice

Soit la grammaire

$$E \rightarrow E + T \mid T, \;\; T \rightarrow T * F \mid F, \;\; F \rightarrow a \mid (E).$$

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

$$\begin{aligned} & \$\|a + a * a + a\$ \\ \models_{\text{shift}} \; & \$a\| + a * a\$ \\ \models_{F \rightarrow a} \; & \$F\| + a * a\$ \\ \models_{T \rightarrow F} \; & \$T\| + a * a\$ \end{aligned}$$

Exercice

Exercice

Soit la grammaire

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

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

$$\begin{aligned}
& \$\|a + a * a + a\$ \\
\models_{\text{shift}} \; & \$a\| + a * a\$ \\
\models_{F \rightarrow a} \; & \$F\| + a * a\$ \\
\models_{T \rightarrow F} \; & \$T\| + a * a\$ \\
\models_{E \rightarrow T} \; & \$E\| + a * a\$
\end{aligned}$$

Exercice

Exercice

Soit la grammaire

$$E \rightarrow E + T \mid T, \;\; T \rightarrow T * F \mid F, \;\; F \rightarrow a \mid (E).$$

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

$$\begin{aligned}
& \$\| a + a * a + a\$ \\
\models_{\text{shift}} \;& \$a\| + a * a\$ \\
\models_{F \rightarrow a} \;& \$F\| + a * a\$ \\
\models_{T \rightarrow F} \;& \$T\| + a * a\$ \\
\models_{E \rightarrow T} \;& \$E\| + a * a\$ \\
\models_{\text{shift}} \;& \$E + \| a * a\$
\end{aligned}$$

Exercice

Exercice

Soit la grammaire

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

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

$$\begin{aligned}
&\models_{\text{shift}} \$a \| + a * a\$ \\
&\models_{F \rightarrow a} \$F \| + a * a\$ \\
&\models_{T \rightarrow F} \$T \| + a * a\$ \\
&\models_{E \rightarrow T} \$E \| + a * a\$ \\
&\models_{\text{shift}} \$E + \| a * a\$ \\
&\models_{\text{shift}} \$E + a \| * a\$
\end{aligned}$$

Exercice

Exercice

Soit la grammaire

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

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

$$\begin{array}{r}
\models_{F \rightarrow a} \$F \| + a * a\$ \\
\models_{T \rightarrow F} \$T \| + a * a\$ \\
\models_{E \rightarrow T} \$E \| + a * a\$ \\
\models_{\text{shift}} \$E + \| a * a\$ \\
\models_{\text{shift}} \$E + a \| * a\$ \\
\models_{F \rightarrow a} \$E + F \| * a\$
\end{array}$$

Exercice

Exercice

Soit la grammaire

$$E \rightarrow E + T \mid T, \;\; T \rightarrow T * F \mid F, \;\; F \rightarrow a \mid (E).$$

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

$$\begin{aligned}
&\models_{T \rightarrow F} \$T \| + a * a\$ \\
&\models_{E \rightarrow T} \$E \| + a * a\$ \\
&\models_{\text{shift}} \$E + \| a * a\$ \\
&\models_{\text{shift}} \$E + a \| * a\$ \\
&\models_{F \rightarrow a} \$E + F \| * a\$ \\
&\models_{T \rightarrow F} \$E + T \| * a\$
\end{aligned}$$

Exercice

Exercice

Soit la grammaire

$$E \rightarrow E + T \mid T,\ \ T \rightarrow T * F \mid F,\ \ F \rightarrow a \mid (E).$$

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

$$\begin{aligned}
&\models_{E \rightarrow T} \$E \| + a * a\$ \\
&\models_{\text{shift}} \$E + \| a * a\$ \\
&\models_{\text{shift}} \$E + a \| * a\$ \\
&\models_{F \rightarrow a} \$E + F \| * a\$ \\
&\models_{T \rightarrow F} \$E + T \| * a\$ \\
&\models_{\text{shift}} \$E + T * \| a\$
\end{aligned}$$

Exercice

Exercice

Soit la grammaire

$$E \rightarrow E + T \mid T, \;\; T \rightarrow T * F \mid F, \;\; F \rightarrow a \mid (E).$$

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

$$\begin{aligned}
&\models_{\text{shift}} \$E + \| a * a\$ \\
&\models_{\text{shift}} \$E + a \| * a\$ \\
&\models_{F \rightarrow a} \$E + F \| * a\$ \\
&\models_{T \rightarrow F} \$E + T \| * a\$ \\
&\models_{\text{shift}} \$E + T * \| a\$ \\
&\models_{\text{shift}} \$E + T * a \| \$
\end{aligned}$$

Exercice

Exercice

Soit la grammaire

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

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

$$\begin{aligned}
&\models_{\text{shift}} \$E + a \| * a\$ \\
&\models_{F \rightarrow a} \$E + F \| * a\$ \\
&\models_{T \rightarrow F} \$E + T \| * a\$ \\
&\models_{\text{shift}} \$E + T * \| a\$ \\
&\models_{\text{shift}} \$E + T * a \| \$ \\
&\models_{F \rightarrow a} \$E + T * F \| \$
\end{aligned}$$

Exercice

Exercice

Soit la grammaire

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

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

$$\begin{aligned}
&\models_{F \to a} \$E + F \| * a\$ \\
&\models_{T \to F} \$E + T \| * a\$ \\
&\models_{\text{shift}} \$E + T * \| a\$ \\
&\models_{\text{shift}} \$E + T * a \| \$ \\
&\models_{F \to a} \$E + T * F \| \$ \\
&\models_{T \to T * F} \$E + T \| \$
\end{aligned}$$

Exercice

Exercice

Soit la grammaire

$$E \rightarrow E + T \mid T, \;\; T \rightarrow T * F \mid F, \;\; F \rightarrow a \mid (E).$$

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

$$\begin{aligned}
&\models_{T \rightarrow F} \$E + T \| * a\$ \\
&\models_{\text{shift}} \$E + T * \| a\$ \\
&\models_{\text{shift}} \$E + T * a \| \$ \\
&\models_{F \rightarrow a} \$E + T * F \| \$ \\
&\models_{T \rightarrow T * F} \$E + T \| \$ \\
&\models_{E \rightarrow E + T} \$E \| \$
\end{aligned}$$

Un amélioration ?

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

Un amélioration ?

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

Un amélioration ?

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

Un amélioration ?

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

Un amélioration ?

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

Exercice

Exercice

Soit la grammaire

$$S \rightarrow A \mid B, \;\; A \rightarrow cA \mid a, \;\; B \rightarrow cB \mid b.$$

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

Exercice

Exercice

Soit la grammaire

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

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

Le langage reconnu par cette grammaire est $\mathcal{L} = \{c^n(a|b) \mid n \geq 0\}$.
L'analyse d'une phrase de ce langage est la suivante :

$$\begin{aligned}
& \$\|c^n(a|b)\$ \\
\models_{\text{shift}}^{\;n} \;& \$c^n\|(a|b)\$ \\
\models_{\text{shift}} \;& \$c^n(a|b)\|\$ \\
\models_{A\rightarrow a|B\rightarrow b} \;& \$c^n(A|B)\|\$ \\
\models_{A\rightarrow cA|B\rightarrow cB}^{\;n} \;& \$(A|B)\|\$ \\
\models_{S\rightarrow A|S\rightarrow B} \;& \$S\|\$
\end{aligned}$$

Exercice

Exercice

Soit la grammaire

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

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

L'analyse descendante n'est pas déterministe : il y a un conflit *predict/predict* entre $[S' \rightarrow \bullet S\$] \| \vdash_{\text{predict}} [S' \rightarrow S \bullet \$][S \rightarrow \bullet A] \|$ et $[S' \rightarrow \bullet S\$] \| \vdash_{\text{predict}} [S' \rightarrow S \bullet \$][S \rightarrow \bullet B] \|$. Ce conflit ne peut être résolu par l'utilisation d'une fenêtre de longueur bornée par un k donné.

Exercice

Exercice

Soit la grammaire

$$S \rightarrow A \mid B, \;\; A \rightarrow cA \mid a, \;\; B \rightarrow cB \mid b.$$

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

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

Poignées

Définition

Une chaîne α 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'α à A dans $\varphi\alpha\sigma$ nous donne bien une forme sententielle.

Poignées

Définition

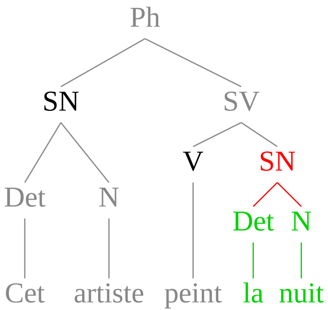
Une chaîne α est un syntagme (*phrase* en anglais) de la forme sententielle $\varphi\alpha\sigma$ si il existe $A \rightarrow \alpha$ dans P et si $S \Rightarrow^* \varphi A \sigma \Rightarrow \varphi\alpha\sigma$.

Définition

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

Poignées

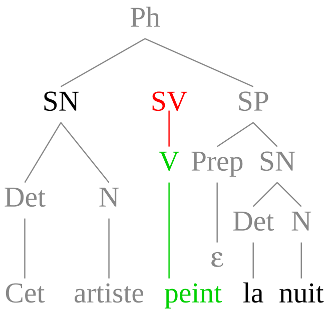
Exemple



$$\mathrm{Ph} \underset{rm}{\Rightarrow}^* \mathrm{SN\ V\ SN} \underset{rm}{\Rightarrow} \mathrm{SN\ V\ Det\ N}$$

Poignées

Exemple



$$\mathrm{Ph} \underset{\mathrm{rm}}{\Rightarrow}^{*} \mathrm{SN}\ \mathrm{SV}\ \mathrm{Det}\ \mathrm{N} \underset{\mathrm{rm}}{\Rightarrow} \mathrm{SN}\ \mathrm{V}\ \mathrm{Det}\ \mathrm{N}$$

Contexte d'une poignée

Définition

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

Contexte d'une poignée

Définition

Soit α 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].

Contexte d'une poignée

Définition

Soit α 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].

Contexte d'une poignée

Définition

Soit α 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].

Contexte d'une poignée

Définition

Soit α 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...

Préfixe viable

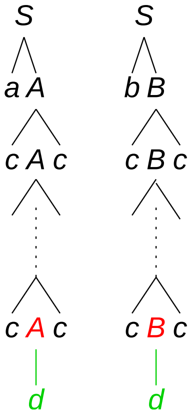
Exemple

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

Préfixe viable

Exemple

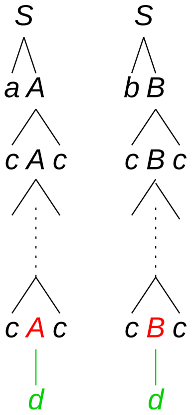
$$S \rightarrow aA \mid bB,\ \ A \rightarrow cAc \mid d,\ \ B \rightarrow cBc \mid d.$$



Préfixe viable

Exemple

$$S \rightarrow aA \,|\, bB, \;\; A \rightarrow cAc \,|\, d, \;\; B \rightarrow cBc \,|\, d.$$



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

Définition

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

Pile viable

Définition

La chaîne φ 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 φ_f dans F et w et x dans Σ^*.

Pile viable

Définition

La chaîne φ 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 φ_f dans F et w et x dans Σ^*.

Théorème

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.

Grammaire caractéristique [Knu65]

Définition

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
- $P_0 = \{[A] \rightarrow \alpha \mid A \rightarrow \alpha \in P\} \cup \{[A] \rightarrow \alpha [B] \mid A \rightarrow \alpha B \beta \in P\}$.

Grammaire caractéristique [Knu65]

Définition

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
- $P_0 = \{[A]\rightarrow\alpha \mid A\rightarrow\alpha \in P\} \cup \{[A]\rightarrow\alpha[B] \mid A\rightarrow\alpha B\beta \in P\}$.

Grammaire caractéristique [Knu65]

Définition

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
- $P_0 = \{[A] \rightarrow \alpha \mid A \rightarrow \alpha \in P\} \cup \{[A] \rightarrow \alpha [B] \mid A \rightarrow \alpha B \beta \in P\}$.

Grammaire caractéristique [Knu65]

Définition

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
- $P_0 = \{[A] \rightarrow \alpha \mid A \rightarrow \alpha \in P\} \cup \{[A] \rightarrow \alpha[B] \mid A \rightarrow \alpha B \beta \in P\}$.

Exemple

La grammaire caractéristique pour

$$S \rightarrow aA \,|\, bB, \;\; A \rightarrow cAc \,|\, d, \;\; B \rightarrow cBc \,|\, d$$

est

$$[S] \rightarrow a[A] \,|\, aA \,|\, b[B] \,|\, bB, \;\; [A] \rightarrow c[A] \,|\, cAc \,|\, d, \;\; [B] \rightarrow c[B] \,|\, cBc \,|\, d.$$

Grammaire caractéristique [Knu65]

Définition

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
- $P_0 = \{[A]{\rightarrow}\alpha \mid A{\rightarrow}\alpha \in P\} \cup \{[A]{\rightarrow}\alpha[B] \mid A{\rightarrow}\alpha B\beta \in P\}$.

Exemple

La grammaire caractéristique pour

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

est

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

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

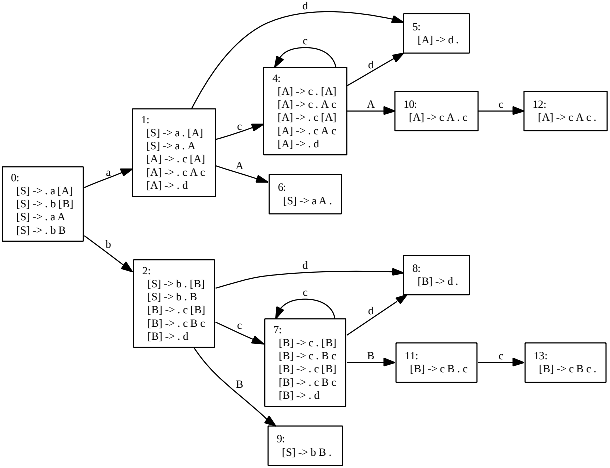
Exercice

Exercice

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

$$[S] \rightarrow a[A] \mid aA \mid b[B] \mid bB, \ [A] \rightarrow c[A] \mid cAc \mid d, \ B \rightarrow c[B] \mid cBc \mid d.$$

Exercice


0:
[S] -> . a [A]
[S] -> . b [B]
[S] -> . a A
[S] -> . b B
a
b
1:
[S] -> a . [A]
[S] -> a . A
[A] -> . c [A]
[A] -> . c A c
[A] -> . d
2:
[S] -> b . [B]
[S] -> b . B
[B] -> . c [B]
[B] -> . c B c
[B] -> . d
c
A
d
4:
[A] -> c . [A]
[A] -> c . A c
[A] -> . c [A]
[A] -> . c A c
[A] -> . d
5:
[A] -> d .
6:
[S] -> a A .
10:
[A] -> c A . c
12:
[A] -> c A c .
B
7:
[B] -> c . [B]
[B] -> c . B c
[B] -> . c [B]
[B] -> . c B c
[B] -> . d
8:
[B] -> d .
9:
[S] -> b B .
11:
[B] -> c B . c
13:
[B] -> c B c .

Item LR(k) valide [Knu65, AU72]

Définition

Un k-item est un couple de la forme $[A \rightarrow \alpha \bullet \beta, u]$ où $A \rightarrow \alpha\beta$ est une production de P et u une chaîne de Σ^k. Il est LR(k)-valide pour un préfixe γ si

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

Item LR(k) valide [Knu65, AU72]

Définition

Un k-item est un couple de la forme $[A \rightarrow \alpha \bullet \beta, u]$ où $A \rightarrow \alpha\beta$ est une production de P et u une chaîne de Σ^k. Il est LR(k)-valide pour un préfixe γ si

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

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

Item LR(k) valide [Knu65, AU72]

Définition

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 Σ^k. Il est LR(k)-valide pour un préfixe γ si

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

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

Exemple

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

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

$$S' \underset{\mathrm{rm}}{\Rightarrow}^* ac^{n-1}Ac^{n-1}\$ \underset{\mathrm{rm}}{\Rightarrow} ac^n Ac^n \$ \underset{\mathrm{rm}}{\Rightarrow} ac^n dc^n \$ \text{ et } 1 : c^n = c.$$

Item LR(k) valide [Knu65, AU72]

Définition

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 Σ^k. Il est LR(k)-valide pour un préfixe γ si

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

Si γ est un préfixe viable de $\mathcal{G}$, alors il existe un item valide pour γ, et réciproquement. L'ensemble des k-items valides pour γ est noté

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

Relations entre items LR(k)

Lemme

Si $S \rightarrow \sigma$ *est une règle de* P*, alors* $[S \rightarrow \bullet\sigma, \$^k]$ *est un* k*-item valide pour* ε*.*

Lemme

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

Lemme

Si $[A \rightarrow \alpha \bullet B\beta, u]$ *est un* k*-item valide pour* γ*, alors* $[B \rightarrow \bullet\delta, First_k(\beta u)]$ *est un* k*-item valide pour* γ*. On note* $[A \rightarrow \alpha \bullet B\beta, u] \overset{\varepsilon}{\curvearrowright} [B \rightarrow \bullet\delta, First_k(\beta u)]$*.*

Relations entre items LR(k)

Lemme

Si $S \rightarrow \sigma$ est une règle de P, alors $[S \rightarrow \bullet \sigma, \$^k]$ est un k-item valide pour ε.

Lemme

Si $[A \rightarrow \alpha \bullet X \beta, u]$ est un k-item valide pour γ, alors $[A \rightarrow \alpha X \bullet \beta, u]$ est un k-item valide pour γX. On note $[A \rightarrow \alpha \bullet X \beta, u] \overset{X}{\curvearrowright} [A \rightarrow \alpha X \bullet \beta, u]$.

Lemme

Si $[A \rightarrow \alpha \bullet B \beta, u]$ est un k-item valide pour γ, alors $[B \rightarrow \bullet \delta, First_k(\beta u)]$ est un k-item valide pour γ. On note $[A \rightarrow \alpha \bullet B \beta, u] \overset{\varepsilon}{\curvearrowright} [B \rightarrow \bullet \delta, First_k(\beta u)]$.

Relations entre items LR(k)

Lemme

Si $S \rightarrow \sigma$ *est une règle de* P*, alors* $[S \rightarrow \bullet \sigma, \$^k]$ *est un* k*-item valide pour* ε*.*

Lemme

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

Lemme

Si $[A \rightarrow \alpha \bullet B \beta, u]$ *est un* k*-item valide pour* γ*, alors* $[B \rightarrow \bullet \delta, First_k(\beta u)]$ *est un* k*-item valide pour* γ*. On note* $[A \rightarrow \alpha \bullet B \beta, u] \overset{\varepsilon}{\curvearrowright} [B \rightarrow \bullet \delta, First_k(\beta u)]$*.*

Relations entre items LR(k)

Lemme

Si $S \rightarrow \sigma$ *est une règle de* P*, alors* $[S \rightarrow \bullet \sigma, \$^k]$ *est un* k*-item valide pour* ε.

Lemme

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

Lemme

Si $[A \rightarrow \alpha \bullet B\beta, u]$ *est un* k*-item valide pour* γ*, alors* $[B \rightarrow \bullet \delta, First_k(\beta u)]$ *est un* k*-item valide pour* γ*. On note* $[A \rightarrow \alpha \bullet B\beta, u] \overset{\varepsilon}{\curvearrowright} [B \rightarrow \bullet \delta, First_k(\beta u)]$.

Relations entre items LR(k)

Lemme

Si $S \rightarrow \sigma$ *est une règle de* P*, alors* $[S \rightarrow \bullet \sigma, \$^k]$ *est un* k*-item valide pour* ε.

Lemme

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

Lemme

Si $[A \rightarrow \alpha \bullet B\beta, u]$ *est un* k*-item valide pour* γ*, alors* $[B \rightarrow \bullet \delta, First_k(\beta u)]$ *est un* k*-item valide pour* γ*. On note* $[A \rightarrow \alpha \bullet B\beta, u] \overset{\varepsilon}{\curvearrowright} [B \rightarrow \bullet \delta, First_k(\beta u)]$.

Théorème

$$Valid_k(\gamma) = \{\iota \mid [S \rightarrow \bullet \sigma, \$^k] \overset{\gamma}{\curvearrowright} \iota\}.$$

Exercice

Exercice

Donnez les ensembles Valid_k pour la grammaire

$$S \rightarrow aA \,|\, bB, \;\; A \rightarrow cAc \,|\, d, \;\; B \rightarrow cBc \,|\, d$$

Exercice

Exercice

Donnez les ensembles Valid$_k$ pour la grammaire

$$S \rightarrow aA \mid bB, \;\; A \rightarrow cAc \mid d, \;\; B \rightarrow cBc \mid d$$

Exercice

Exercice

Donnez les ensembles Valid$_k$ pour la grammaire

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

$$\mathrm{Valid}_k(\varepsilon) = \{[S \rightarrow \bullet aA, \$^k], [S \rightarrow \bullet bB, \$^k]\}$$

Exercice

Exercice

Donnez les ensembles Valid_k pour la grammaire

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

$$\mathrm{Valid}_k(\varepsilon) = \{[S \to \bullet aA, \$^k], [S \to \bullet bB, \$^k]\}$$

$$\mathrm{Valid}_k(a) = \{[S \to a \bullet A, \$^k], [A \to \bullet cAc, \$^k], [A \to \bullet d, \$^k]\}$$

Exercice

Exercice

Donnez les ensembles Valid$_k$ pour la grammaire

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

$$\text{Valid}_k(\varepsilon) = \{[S \to \bullet aA, \$^k], [S \to \bullet bB, \$^k]\}$$

$$\text{Valid}_k(a) = \{[S \to a \bullet A, \$^k], [A \to \bullet cAc, \$^k], [A \to \bullet d, \$^k]\}$$

$$\text{Valid}_k(ac^{n+1}) = \{[A \to c \bullet Ac, k : c^n\$^{k-n}], [A \to \bullet cAc, k : c^{n+1}\$^{k-n-1}],$$

$$[A \to \bullet d, k : c^{n+1}\$^{k-n-1}]\}$$

Exercice

Exercice

Donnez les ensembles Valid$_k$ pour la grammaire

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

$$\mathrm{Valid}_k(\varepsilon) = \{[S \rightarrow \bullet aA, \$^k], [S \rightarrow \bullet bB, \$^k]\}$$

$$\mathrm{Valid}_k(a) = \{[S \rightarrow a \bullet A, \$^k], [A \rightarrow \bullet cAc, \$^k], [A \rightarrow \bullet d, \$^k]\}$$

$$\mathrm{Valid}_k(ac^{n+1}) = \{[A \rightarrow c \bullet Ac, k : c^n \$^{k-n}], [A \rightarrow \bullet cAc, k : c^{n+1} \$^{k-n-1}],$$

$$[A \rightarrow \bullet d, k : c^{n+1} \$^{k-n-1}]\}$$

$$\mathrm{Valid}_k(ac^n d) = \{[A \rightarrow d \bullet, k : c^n \$^{k-n}]\}$$

Exercice

Exercice

Donnez les ensembles Valid_k pour la grammaire

$$S \rightarrow aA \mid bB, \;\; A \rightarrow cAc \mid d, \;\; B \rightarrow cBc \mid d$$

$$\mathrm{Valid}_k(a) = \{[S \rightarrow a \bullet A, \$^k], [A \rightarrow \bullet cAc, \$^k], [A \rightarrow \bullet d, \$^k]\}$$

$$\mathrm{Valid}_k(ac^{n+1}) = \{[A \rightarrow c \bullet Ac, k : c^n \$^{k-n}], [A \rightarrow \bullet cAc, k : c^{n+1} \$^{k-n-1}],$$

$$[A \rightarrow \bullet d, k : c^{n+1} \$^{k-n-1}]\}$$

$$\mathrm{Valid}_k(ac^n d) = \{[A \rightarrow d \bullet, k : c^n \$^{k-n}]\}$$

$$\mathrm{Valid}_k(aA) = \{[S \rightarrow aA \bullet, \$^k]\}$$

Exercice

Exercice

Donnez les ensembles Valid_k pour la grammaire

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

$$\mathrm{Valid}_k(ac^{n+1}) = \{[A \to c \bullet Ac, k : c^n \$^{k-n}], [A \to \bullet cAc, k : c^{n+1} \$^{k-n-1}],$$

$$[A \to \bullet d, k : c^{n+1} \$^{k-n-1}]\}$$

$$\mathrm{Valid}_k(ac^n d) = \{[A \to d \bullet, k : c^n \$^{k-n}]\}$$

$$\mathrm{Valid}_k(aA) = \{[S \to aA \bullet, \$^k]\}$$

$$\mathrm{Valid}_k(ac^{n+1}A) = \{[A \to cA \bullet c, k : c^n \$^{k-n}]\}$$

Exercice

Exercice

Donnez les ensembles Valid_k pour la grammaire

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

$$\mathrm{Valid}_k(ac^n d) = \{[A \rightarrow d \bullet, k : c^n \$^{k-n}]\}$$

$$\mathrm{Valid}_k(aA) = \{[S \rightarrow aA \bullet, \$^k]\}$$

$$\mathrm{Valid}_k(ac^{n+1}A) = \{[A \rightarrow cA \bullet c, k : c^n \$^{k-n}]\}$$

$$\mathrm{Valid}_k(ac^{n+1}Ac) = \{[A \rightarrow cAc \bullet, k : c^n \$^{k-n}]\}$$

Exercice

Exercice

Donnez les ensembles Valid_k pour la grammaire

$$S \rightarrow aA \mid bB, \; A \rightarrow cAc \mid d, \; B \rightarrow cBc \mid d$$

$$\text{Valid}_k(aA) = \{[S \rightarrow aA \bullet, \$^k]\}$$

$$\text{Valid}_k(ac^{n+1}A) = \{[A \rightarrow cA \bullet c, k : c^n \$^{k-n}]\}$$

$$\text{Valid}_k(ac^{n+1}Ac) = \{[A \rightarrow cAc \bullet, k : c^n \$^{k-n}]\}$$

$$\text{Valid}_k(b) = \{[S \rightarrow b \bullet B, \$^k], [B \rightarrow \bullet cBc, \$^k], [B \rightarrow \bullet d, \$^k]\}$$

Exercice

Exercice

Donnez les ensembles Valid_k pour la grammaire

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

$$\mathrm{Valid}_k(ac^{n+1}A) = \{[A \to cA \bullet c, k : c^n\$^{k-n}]\}$$

$$\mathrm{Valid}_k(ac^{n+1}Ac) = \{[A \to cAc \bullet, k : c^n\$^{k-n}]\}$$

$$\mathrm{Valid}_k(b) = \{[S \to b \bullet B, \$^k], [B \to \bullet cBc, \$^k], [B \to \bullet d, \$^k]\}$$

$$\mathrm{Valid}_k(bc^{n+1}) = \{[B \to c \bullet Bc, k : c^n\$^{k-n}], [B \to \bullet cBc, k : c^{n+1}\$^{k-n-1}],$$

$$[B \to \bullet d, k : c^{n+1}\$^{k-n-1}]\}$$

Exercice

Exercice

Donnez les ensembles Valid_k pour la grammaire

$$S \rightarrow aA \mid bB, \;\; A \rightarrow cAc \mid d, \;\; B \rightarrow cBc \mid d$$

$$\text{Valid}_k(ac^{n+1}Ac) = \{[A \rightarrow cAc\bullet, k : c^n\$^{k-n}]\}$$

$$\text{Valid}_k(b) = \{[S \rightarrow b\bullet B, \$^k], [B \rightarrow \bullet cBc, \$^k], [B \rightarrow \bullet d, \$^k]\}$$

$$\text{Valid}_k(bc^{n+1}) = \{[B \rightarrow c\bullet Bc, k : c^n\$^{k-n}], [B \rightarrow \bullet cBc, k : c^{n+1}\$^{k-n-1}],$$

$$[B \rightarrow \bullet d, k : c^{n+1}\$^{k-n-1}]\}$$

$$\text{Valid}_k(bc^nd) = \{[B \rightarrow d\bullet, k : c^n\$^{k-n}]\}$$

Exercice

Exercice

Donnez les ensembles Valid_k pour la grammaire

$$S \rightarrow aA \,|\, bB, \;\; A \rightarrow cAc \,|\, d, \;\; B \rightarrow cBc \,|\, d$$

$$\text{Valid}_k(b) = \{[S \rightarrow b \bullet B, \$^k], [B \rightarrow \bullet cBc, \$^k], [B \rightarrow \bullet d, \$^k]\}$$

$$\text{Valid}_k(bc^{n+1}) = \{[B \rightarrow c \bullet Bc, k : c^n \$^{k-n}], [B \rightarrow \bullet cBc, k : c^{n+1}\$^{k-n-1}],$$

$$[B \rightarrow \bullet d, k : c^{n+1}\$^{k-n-1}]\}$$

$$\text{Valid}_k(bc^n d) = \{[B \rightarrow d \bullet, k : c^n \$^{k-n}]\}$$

$$\text{Valid}_k(bB) = \{[S \rightarrow bB \bullet, \$^k]\}$$

Exercice

Exercice

Donnez les ensembles Valid_k pour la grammaire

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

$$\text{Valid}_k(bc^{n+1}) = \{[B \rightarrow c \bullet Bc, k : c^n\$^{k-n}], [B \rightarrow \bullet cBc, k : c^{n+1}\$^{k-n-1}],$$

$$[B \rightarrow \bullet d, k : c^{n+1}\$^{k-n-1}]\}$$

$$\text{Valid}_k(bc^n d) = \{[B \rightarrow d \bullet, k : c^n\$^{k-n}]\}$$

$$\text{Valid}_k(bB) = \{[S \rightarrow bB \bullet, \$^k]\}$$

$$\text{Valid}_k(bc^{n+1}B) = \{[B \rightarrow cB \bullet c, k : c^n\$^{k-n}]\}$$

Exercice

Exercice

Donnez les ensembles Valid$_k$ pour la grammaire

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

$$\mathrm{Valid}_k(bc^n d) = \{[B \rightarrow d \bullet, k : c^n \$^{k-n}]\}$$

$$\mathrm{Valid}_k(bB) = \{[S \rightarrow bB \bullet, \$^k]\}$$

$$\mathrm{Valid}_k(bc^{n+1} B) = \{[B \rightarrow cB \bullet c, k : c^n \$^{k-n}]\}$$

$$\mathrm{Valid}_k(bc^{n+1} Bc) = \{[B \rightarrow cBc \bullet, k : c^n \$^{k-n}]\}$$

Exercice

Exercice

Donnez les ensembles Valid_k pour la grammaire

$$S \rightarrow aA \,|\, bB, \;\; A \rightarrow cAc \,|\, d, \;\; B \rightarrow cBc \,|\, d$$

$$\text{Valid}_k(bB) = \{[S \rightarrow bB\bullet, \$^k]\}$$

$$\text{Valid}_k(bc^{n+1}B) = \{[B \rightarrow cB\bullet c, k : c^n\$^{k-n}]\}$$

$$\text{Valid}_k(bc^{n+1}Bc) = \{[B \rightarrow cBc\bullet, k : c^n\$^{k-n}]\}$$

$$\text{Valid}_k(\gamma) = \emptyset \text{ pour toute autre chaîne } \gamma \text{ de } V^*.$$

Équivalence LR(k)

Définition

La chaîne γ_1 est LR(k)-équivalente à la chaîne γ_2, dénoté par $\gamma_1 \equiv_{\mathrm{LR}(k)} \gamma_2$, si et seulement si $\mathrm{Valid}_k(\gamma_1) = \mathrm{Valid}_k(\gamma_2)$.

Équivalence LR(k)

Définition

La chaîne γ_1 est LR(k)-équivalente à la chaîne γ_2, dénoté par $\gamma_1 \equiv_{\mathrm{LR}(k)} \gamma_2$, si et seulement si $\mathrm{Valid}_k(\gamma_1) = \mathrm{Valid}_k(\gamma_2)$.

Théorème

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

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

Équivalence LR(k)

Définition

La chaîne γ_1 est LR(k)-équivalente à la chaîne γ_2, dénoté par $\gamma_1 \equiv_{\mathrm{LR}(k)} \gamma_2$, si et seulement si $\mathrm{Valid}_k(\gamma_1) = \mathrm{Valid}_k(\gamma_2)$.

Théorème

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

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

Exercice

Prouver le théorème précédent.

Automate LR(k)

Définition

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

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

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

$$[\delta]_k \| au \vdash_{\text{shift}} [\delta]_k[\delta a]_k \| u,$$

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

Automate LR(k)

Définition

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

$$[\delta]_k[\delta X_1]_k \ldots [\delta X_1 \ldots X_n]_k \| u \vdash_{A \rightarrow X_1 \ldots X_n} [\delta]_k[\delta A]_k \| u,$$

  si un item $[A \rightarrow X_1 \ldots X_n \bullet, u]$ est LR(k)-valide pour δ, et

$$[\delta]_k \| au \vdash_{\text{shift}} [\delta]_k[\delta a]_k \| u,$$

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

Automate LR(k)

Définition

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

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

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

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

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

Analyse ascendante LR

Exemple

Soit la grammaire

$$S \rightarrow aA \,|\, bB, \;\; A \rightarrow cAc \,|\, d, \;\; B \rightarrow cBc \,|\, d \;\; :$$

$\mathrm{Valid}_0(\varepsilon)$
$S \rightarrow . \, a \, A$ $S \rightarrow . \, b \, B$

a c c d c c

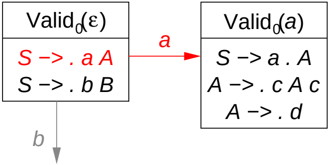
$$\$[\varepsilon] \| accdcc\$$$

Analyse ascendante LR

Exemple

Soit la grammaire

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



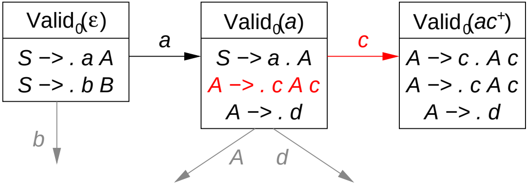
a c c d c c

Analyse ascendante LR

Exemple

Soit la grammaire

$$S \rightarrow aA \,|\, bB, \;\; A \rightarrow cAc \,|\, d, \;\; B \rightarrow cBc \,|\, d \quad :$$

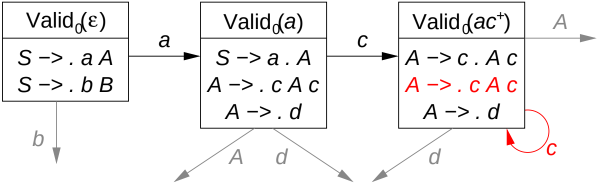


Analyse ascendante LR

Exemple

Soit la grammaire

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

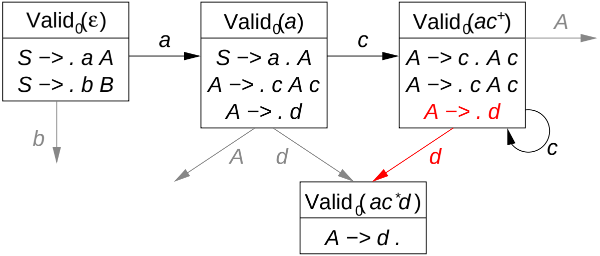


Analyse ascendante LR

Exemple

Soit la grammaire

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

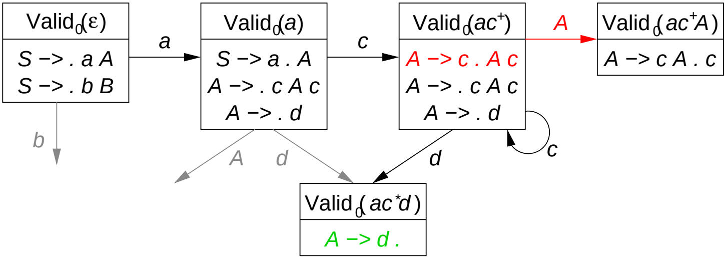


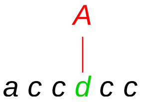
Analyse ascendante LR

Exemple

Soit la grammaire

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





$$\models_{\text{shift}} \$[\varepsilon][a][ac][acc][accd]\|cc\$$$

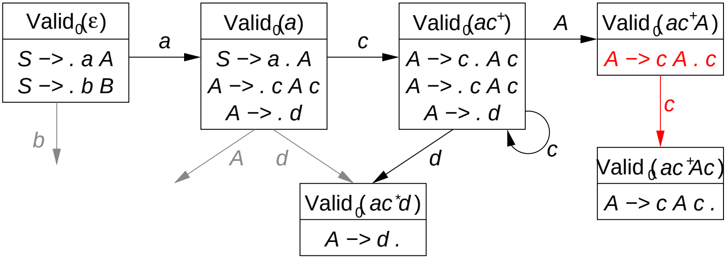
$$\models_{A\rightarrow d} \$[\varepsilon][a][ac][acc][accA]\|cc\$$$

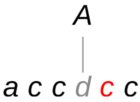
Analyse ascendante LR

Exemple

Soit la grammaire

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





$$\models_{A\rightarrow d} \$[\varepsilon][a][ac][acc][accA] \| cc\$$$

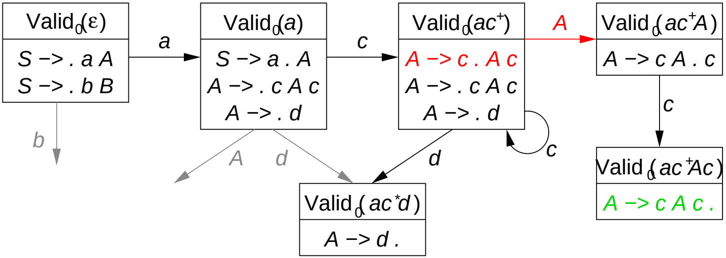
$$\models_{\text{shift}} \$[\varepsilon][a][ac][acc][accA][accAc] \| c\$$$

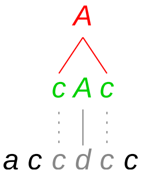
Analyse ascendante LR

Exemple

Soit la grammaire

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





$$\models_{\text{shift}} \$[\varepsilon][a][ac][acc][accA][accAc] \| c\$$$

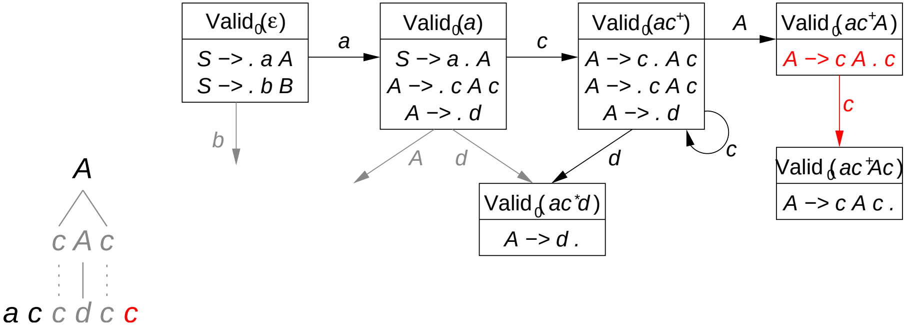
$$\models_{A \to cAc} \$[\varepsilon][a][ac][acA] \| c\$$$

Analyse ascendante LR

Exemple

Soit la grammaire

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



$$\models_{A \rightarrow cAc} \$[\varepsilon][a][ac][acA] \| c\$$$

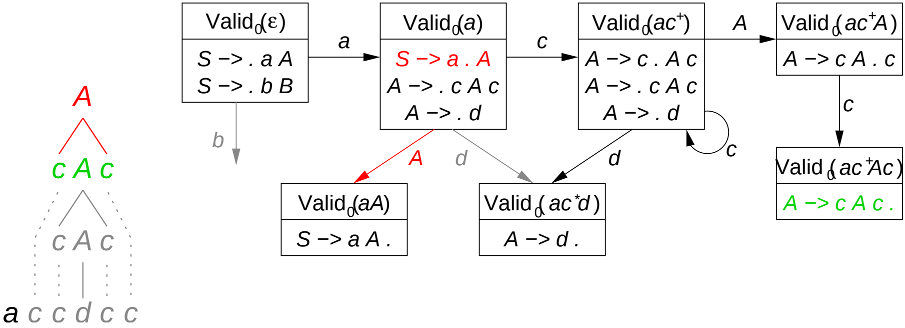
$$\models_{\text{shift}} \$[\varepsilon][a][ac][acA][acAc] \| \$$$

Analyse ascendante LR

Exemple

Soit la grammaire

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



$$\models_{\text{shift}} \$[\varepsilon][a][ac][acA][acAc]\|\$$$

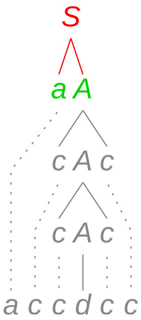
$$\models_{A\to cAc} \$[\varepsilon][a][aA]\|\$$$

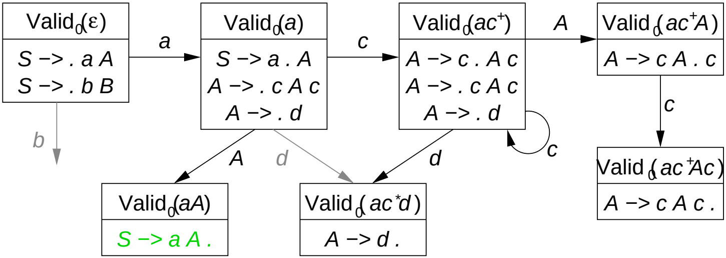
Analyse ascendante LR

Exemple

Soit la grammaire

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





$$\models_{A \rightarrow cAc} \$[\varepsilon][a][aA] \| \$$$

$$\models_{S \rightarrow aA} \$[\varepsilon][S] \| \$$$

Grammaires LR(k)

Définition

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

Grammaires LR(k)

Définition

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

Grammaires LR(k)

Définition

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

Grammaires LR(k)

Définition

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

Grammaires LR(k)

Définition

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

Grammaires LR(k)

Définition

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

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

Théorème

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

Langages déterministes

Théorème

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.

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.

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

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

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.

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