Troisième partie III

Analyse LALR

Analyse ascendante LR

Exemple

Soit la grammaire

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

$\text{Valid}_0(\varepsilon)$
$S \text{ -> } .\ a\ A$ $S \text{ -> } .\ b\ B$

a c c d c c

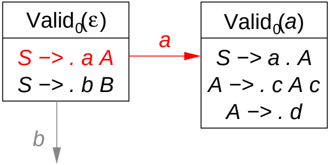
$$\$[\varepsilon]\|accdcc\$$$

Analyse ascendante LR

Exemple

Soit la grammaire

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



a c c d c c

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

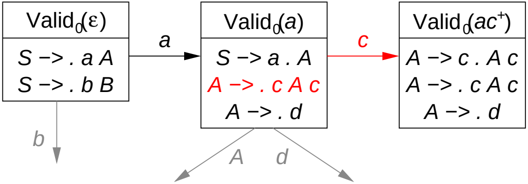
$$\models_{\text{shift}} \$[\varepsilon][a] \| ccdcc\$$$

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*

$$\models_{\text{shift}} \$[\varepsilon][a] \| ccdcc\$$$

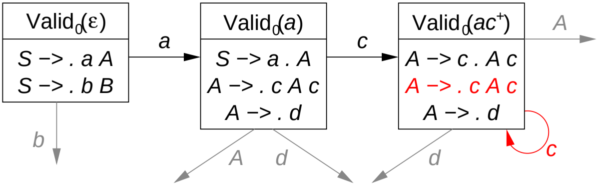
$$\models_{\text{shift}} \$[\varepsilon][a][ac] \| cdcc\$$$

Analyse ascendante LR

Exemple

Soit la grammaire

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



$a\ c\ c\ d\ c\ c$

$$\models_{shift} \$[\varepsilon][a][ac] \| cdcc\$$$

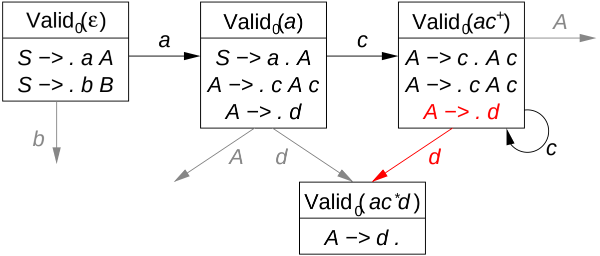
$$\models_{shift} \$[\varepsilon][a][ac][acc] \| dcc\$$$

Analyse ascendante LR

Exemple

Soit la grammaire

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



a c c d c c

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

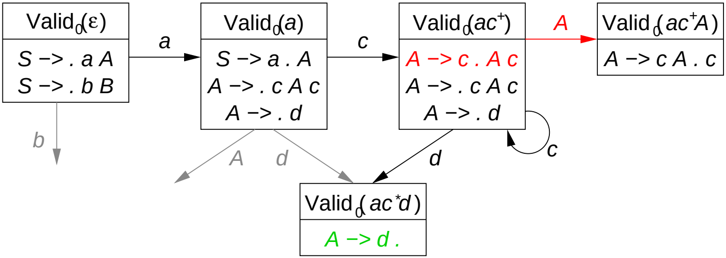
$$\models_{\text{shift}} \$[\varepsilon][a][ac][acc][acc\color{red}{d}] \| cc\$$$

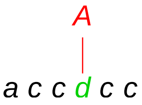
Analyse ascendante LR

Exemple

Soit la grammaire

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





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

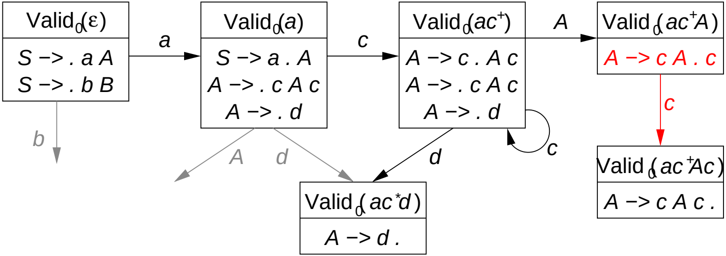
$$\models_{A \to d} \$[\varepsilon][a][ac][acc][accA]\|cc\$$$

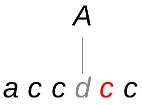
Analyse ascendante LR

Exemple

Soit la grammaire

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





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

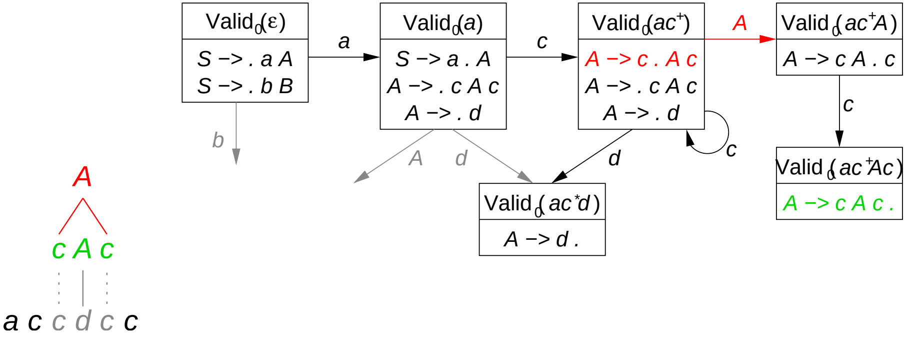
$$\models_{\text{shift}} \$[\varepsilon][a][ac][acc][accA][accAc] \| c\$$$

Analyse ascendante LR

Exemple

Soit la grammaire

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



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

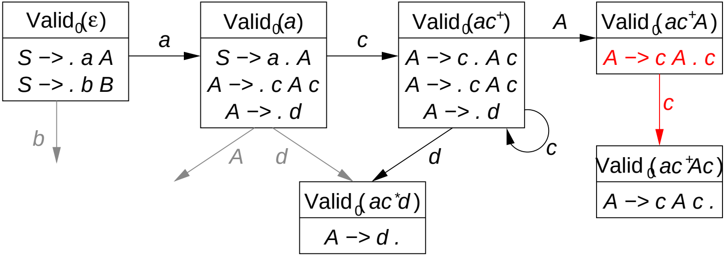
$$\models_{A\to cAc} \$[\varepsilon][a][ac][acA]\|c\$$$

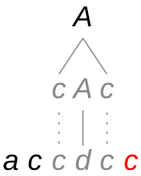
Analyse ascendante LR

Exemple

Soit la grammaire

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





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

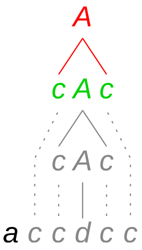
$$\models_{\text{shift}} \$[\varepsilon][a][ac][acA][acAc] \| \$$$

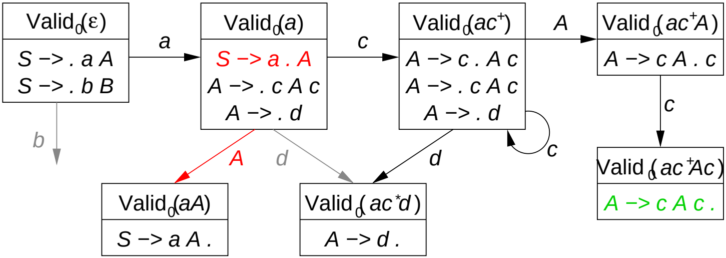
Analyse ascendante LR

Exemple

Soit la grammaire

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





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

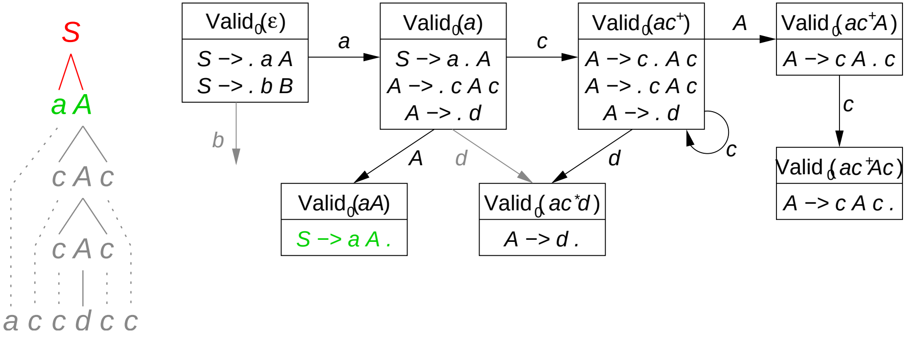
$$\models_{A \rightarrow cAc} \$[\varepsilon][a][aA]\|\$$$

Analyse ascendante LR

Exemple

Soit la grammaire

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



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

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

En résumé

- Le langage des contextes gauches des poignées est rationnel.
- Un langage rationnel définit des classes d'équivalence sur les phrases du langage.
- Ces classes d'équivalence peuvent être calculées directement en utilisant les items valides LR(k)
- Ces classes d'équivalence servent d'états pour l'automate à pile LR(k).
- Il existe $2^{|\mathcal{G}||\Sigma|^k}$ classes d'équivalence distinctes possibles.

En résumé

- Le langage des contextes gauches des poignées est rationnel, puisque généré par la grammaire caractéristique, qui est linéaire.
- Un langage rationnel définit des classes d'équivalence sur les phrases du langage.
- Ces classes d'équivalence peuvent être calculées directement en utilisant les items valides LR(k)
- Ces classes d'équivalence servent d'états pour l'automate à pile LR(k).
- Il existe $2^{|\mathcal{G}||\Sigma|^k}$ classes d'équivalence distinctes possibles.

En résumé

- Le langage des contextes gauches des poignées est rationnel.
- Un langage rationnel définit des classes d'équivalence sur les phrases du langage par le biais des états de son automate à états finis déterministe minimal.
- Ces classes d'équivalence peuvent être calculées directement en utilisant les items valides LR(k)
- Ces classes d'équivalence servent d'états pour l'automate à pile LR(k).
- Il existe $2^{|\mathcal{G}||\Sigma|^k}$ classes d'équivalence distinctes possibles.

En résumé

- Le langage des contextes gauches des poignées est rationnel.
- Un langage rationnel définit des classes d'équivalence sur les phrases du langage.
- Ces classes d'équivalence peuvent être calculées directement en utilisant les items valides 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 $\mathrm{Valid}_k(\gamma_1) = \mathrm{Valid}_k(\gamma_2)$

- Ces classes d'équivalence servent d'états pour l'automate à pile LR(k).
- Il existe $2^{|\mathcal{G}||\Sigma|^k}$ classes d'équivalence distinctes possibles.

En résumé

- Le langage des contextes gauches des poignées est rationnel.
- Un langage rationnel définit des classes d'équivalence sur les phrases du langage.
- Ces classes d'équivalence peuvent être calculées directement en utilisant les items valides LR(k)
- Ces classes d'équivalence servent d'états pour l'automate à pile LR(k).
- Il existe $2^{|\mathcal{G}||\Sigma|^k}$ classes d'équivalence distinctes possibles.

En résumé

- Le langage des contextes gauches des poignées est rationnel.
- Un langage rationnel définit des classes d'équivalence sur les phrases du langage.
- Ces classes d'équivalence peuvent être calculées directement en utilisant les items valides LR(k)
- Ces classes d'équivalence servent d'états pour l'automate à pile LR(k).
- Il existe $2^{|\mathcal{G}||\Sigma|^k}$ classes d'équivalence distinctes possibles.

En résumé

- Le langage des contextes gauches des poignées est rationnel.
- Un langage rationnel définit des classes d'équivalence sur les phrases du langage.
- Ces classes d'équivalence peuvent être calculées directement en utilisant les items valides LR(k)
- Ces classes d'équivalence servent d'états pour l'automate à pile LR(k).
- Il existe $2^{|\mathcal{G}||\Sigma|^k}$ classes d'équivalence distinctes possibles.

La classe de grammaires LR(k) est la classe analysable par automates à pile déterministes la plus large possible.

Le problème

Exercice

Donnez les ensembles Valid_0 pour la grammaire

$$S \longrightarrow aSa \mid bSb \mid c.$$

Combien y-a-t'il de classes d'équivalence LR(0) ? LR(1) ? LR(k) ?

Le problème

Exercice

Donnez les ensembles Valid_0 pour la grammaire

$$S \to aSa \mid bSb \mid c.$$

Combien y-a-t'il de classes d'équivalence LR(0) ? LR(1) ? LR(k) ?

Pour chacun des corps d'item $S \to a \bullet Sa$, $S \to aS \bullet a$, $S \to aSa \bullet$, $S \to b \bullet Sb$, $S \to bS \bullet b$, $S \to bSb \bullet$ et $S \to c \bullet$, n'importe quel ensemble de k symboles ou moins, pris parmi $\{a, b\}$, peut apparaître en fenêtre, complété par des \$.
Soit

$$2^k + 2^{k-1} + \cdots + 2 + 1 = 2^{k+1} - 1$$

fenêtres de k symboles possibles pour chaque corps. Soit au final $7(2^{k+1} - 1) + 1$ états LR(k).

Fusion d'états

Les trois états $\boxed{\begin{array}{l} S \to a \bullet Sa, \$ \\ S \to \bullet aSa, a \\ S \to \bullet bSb, a \\ S \to \bullet c, a \end{array}}$, $\boxed{\begin{array}{l} S \to a \bullet Sa, a \\ S \to \bullet aSa, a \\ S \to \bullet bSb, a \\ S \to \bullet c, a \end{array}}$ et $\boxed{\begin{array}{l} S \to a \bullet Sa, b \\ S \to \bullet aSa, a \\ S \to \bullet bSb, a \\ S \to \bullet c, a \end{array}}$

sont fusionnés en un seul $\boxed{\begin{array}{l} S \to a \bullet Sa, \{\$, a, b\} \\ S \to \bullet aSa, \{a\} \\ S \to \bullet bSb, \{a\} \\ S \to \bullet c, \{a\} \end{array}}$.

Fusion d'états

Les trois états $\boxed{\begin{array}{l} S \rightarrow a \bullet Sa, \$ \\ S \rightarrow \bullet aSa, a \\ S \rightarrow \bullet bSb, a \\ S \rightarrow \bullet c, a \end{array}}$, $\boxed{\begin{array}{l} S \rightarrow a \bullet Sa, a \\ S \rightarrow \bullet aSa, a \\ S \rightarrow \bullet bSb, a \\ S \rightarrow \bullet c, a \end{array}}$ et $\boxed{\begin{array}{l} S \rightarrow a \bullet Sa, b \\ S \rightarrow \bullet aSa, a \\ S \rightarrow \bullet bSb, a \\ S \rightarrow \bullet c, a \end{array}}$

sont fusionnés en un seul $\boxed{\begin{array}{l} S \rightarrow a \bullet Sa, \{\$, a, b\} \\ S \rightarrow \bullet aSa, \{a\} \\ S \rightarrow \bullet bSb, \{a\} \\ S \rightarrow \bullet c, \{a\} \end{array}}$.

Fusion d'états

Les trois états $\boxed{\begin{array}{l} S \rightarrow a \bullet Sa, \$ \\ S \rightarrow \bullet aSa, a \\ S \rightarrow \bullet bSb, a \\ S \rightarrow \bullet c, a \end{array}}$, $\boxed{\begin{array}{l} S \rightarrow a \bullet Sa, a \\ S \rightarrow \bullet aSa, a \\ S \rightarrow \bullet bSb, a \\ S \rightarrow \bullet c, a \end{array}}$ et $\boxed{\begin{array}{l} S \rightarrow a \bullet Sa, b \\ S \rightarrow \bullet aSa, a \\ S \rightarrow \bullet bSb, a \\ S \rightarrow \bullet c, a \end{array}}$

sont fusionnés en un seul $\boxed{\begin{array}{l} S \rightarrow a \bullet Sa, \{\$, a, b\} \\ S \rightarrow \bullet aSa, \{a\} \\ S \rightarrow \bullet bSb, \{a\} \\ S \rightarrow \bullet c, \{a\} \end{array}}$.

Fusion d'états

Les trois états $\boxed{\begin{array}{l} S \rightarrow a \bullet Sa, \$ \\ S \rightarrow \bullet aSa, a \\ S \rightarrow \bullet bSb, a \\ S \rightarrow \bullet c, a \end{array}}$, $\boxed{\begin{array}{l} S \rightarrow a \bullet Sa, a \\ S \rightarrow \bullet aSa, a \\ S \rightarrow \bullet bSb, a \\ S \rightarrow \bullet c, a \end{array}}$ et $\boxed{\begin{array}{l} S \rightarrow a \bullet Sa, b \\ S \rightarrow \bullet aSa, a \\ S \rightarrow \bullet bSb, a \\ S \rightarrow \bullet c, a \end{array}}$

sont fusionnés en un seul $\boxed{\begin{array}{l} S \rightarrow a \bullet Sa, \{\$, a, b\} \\ S \rightarrow \bullet aSa, \{a\} \\ S \rightarrow \bullet bSb, \{a\} \\ S \rightarrow \bullet c, \{a\} \end{array}}$.

Fusion d'états

Les trois états $\boxed{\begin{array}{l} S \to a \bullet Sa, \$ \\ S \to \bullet aSa, a \\ S \to \bullet bSb, a \\ S \to \bullet c, a \end{array}}$, $\boxed{\begin{array}{l} S \to a \bullet Sa, a \\ S \to \bullet aSa, a \\ S \to \bullet bSb, a \\ S \to \bullet c, a \end{array}}$ et $\boxed{\begin{array}{l} S \to a \bullet Sa, b \\ S \to \bullet aSa, a \\ S \to \bullet bSb, a \\ S \to \bullet c, a \end{array}}$

sont fusionnés en un seul $\boxed{\begin{array}{l} S \to a \bullet Sa, \{\$, a, b\} \\ S \to \bullet aSa, \{a\} \\ S \to \bullet bSb, \{a\} \\ S \to \bullet c, \{a\} \end{array}}$.

Fusion d'états

Les trois états $\boxed{\begin{array}{l} S \rightarrow a \bullet Sa, \$ \\ S \rightarrow \bullet aSa, a \\ S \rightarrow \bullet bSb, a \\ S \rightarrow \bullet c, a \end{array}}$, $\boxed{\begin{array}{l} S \rightarrow a \bullet Sa, a \\ S \rightarrow \bullet aSa, a \\ S \rightarrow \bullet bSb, a \\ S \rightarrow \bullet c, a \end{array}}$ et $\boxed{\begin{array}{l} S \rightarrow a \bullet Sa, b \\ S \rightarrow \bullet aSa, a \\ S \rightarrow \bullet bSb, a \\ S \rightarrow \bullet c, a \end{array}}$

sont fusionnés en un seul $\boxed{\begin{array}{l} S \rightarrow a \bullet Sa, \{\$, a, b\} \\ S \rightarrow \bullet aSa, \{a\} \\ S \rightarrow \bullet bSb, \{a\} \\ S \rightarrow \bullet c, \{a\} \end{array}}$.

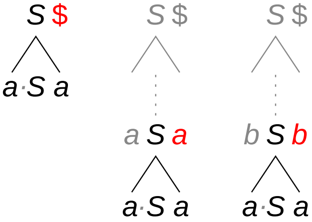
Définition ([KM81])

La fenêtre LALR(k) d'un corps valide pour un préfixe γ est

$$LA([\gamma], A \rightarrow \alpha \bullet \alpha') = \{k : z \mid S \underset{\mathrm{rm}}{\Rightarrow}^{*} \delta A x \underset{\mathrm{rm}}{\Rightarrow} \delta \alpha \alpha' x \underset{\mathrm{rm}}{\Rightarrow}^{*} \delta \alpha z \text{ et } \delta \alpha \equiv_{\mathsf{LR}(0)} \gamma\}.$$

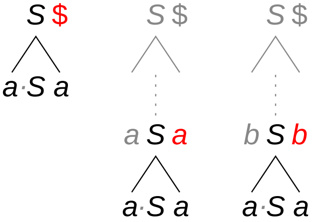
Équivalence d'arbres

Le corps d'item $S \rightarrow a \bullet Sa$ peut apparaître dans trois types d'arbres différents :



Équivalence d'arbres

Le corps d'item $S \to a \bullet Sa$ peut apparaître dans trois types d'arbres différents :



Les préfixes viables correspondants sont LR(0)-équivalents :

$$[a]_0 = \{wa \mid w \in \{a,b\}^*\} = (a|b)^* a.$$

Fenêtre sur un arbre

Le calcul de la fenêtre est compliqué par la présence de règles vides.

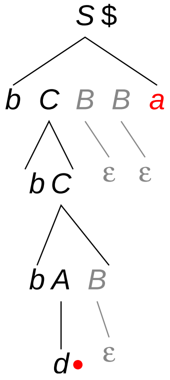
Exemple

$$S \rightarrow CBBa,\ A \rightarrow d,\ B \rightarrow \varepsilon,\ C \rightarrow bC \mid bAB \mid bDBb,\ D \rightarrow d.$$

Fenêtre sur un arbre

Exemple

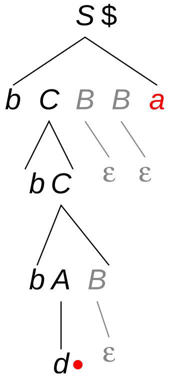
$$S \to CBBa,\ A \to d,\ B \to \varepsilon,\ C \to bC \mid bAB \mid bDBb,\ D \to d.$$



Fenêtre sur un arbre

Exemple

$$S\to CBBa,\ A\to d,\ B\to\varepsilon,\ C\to bC\,|\,bAB\,|\,bDBb,\ D\to d.$$



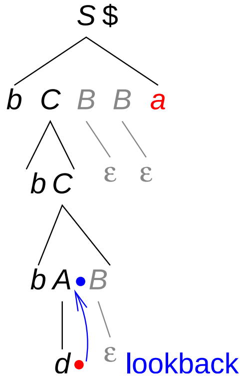
Une position dans un arbre syntaxique équivaut à une position dans une forme sententielle droite. On écrit une telle position $\delta[A\to\alpha\bullet\alpha']x$, où $\delta\alpha\alpha' x$ est une forme sententielle droite et $\alpha\alpha'$ une poignée dans cette forme sententielle droite.

Par exemple, on écrirait $bbb[A\to d\bullet]a$ pour la position dans l'arbre ci-contre.

Fenêtre sur un arbre

Exemple

$$S\to CBBa,\ A\to d,\ B\to\varepsilon,\ C\to bC\,|\,bAB\,|\,bDBb,\ D\to d.$$



Définition

Si $\delta Cy \underset{\mathrm{rm}}{\Rightarrow} \delta\gamma A\sigma y \underset{\mathrm{rm}}{\Rightarrow}^{*} \delta\gamma Axy \underset{\mathrm{rm}}{\Rightarrow} \delta\gamma\alpha xy$, alors

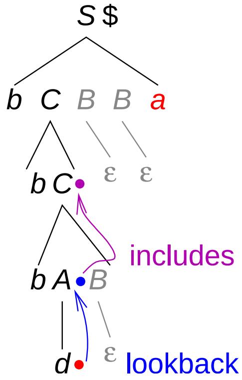
$$\delta\gamma[A\to\alpha\bullet]xy\ \textbf{lookback}\ \delta[C\to\gamma A\bullet\sigma]y.$$

Par exemple,
$bbb[A\to d\bullet]a$ **lookback** $bb[C\to bA\bullet B]a$.

Fenêtre sur un arbre

Exemple

$$S \to CBBa,\ A \to d,\ B \to \varepsilon,\ C \to bC \mid bAB \mid bDBb,\ D \to d.$$



Définition

Si $\delta Cy \underset{\mathrm{rm}}{\Rightarrow} \delta\gamma A\sigma y \underset{\mathrm{rm}}{\Rightarrow}^* \delta\gamma Axy \underset{\mathrm{rm}}{\Rightarrow} \delta\gamma\alpha\varphi xy \underset{\mathrm{rm}}{\Rightarrow}^* \delta\gamma\alpha xy$, alors

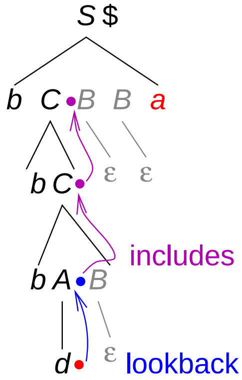
$$\delta\gamma[A \to \alpha \bullet \varphi]xy \ \textbf{includes}\ \delta[C \to \gamma A \bullet \sigma]y.$$

Par exemple, $bb[C \to bA \bullet B]a$ **includes** $b[C \to bC\bullet]a$.

Fenêtre sur un arbre

Exemple

$$S \to CBBa,\ A \to d,\ B \to \varepsilon,\ C \to bC \mid bAB \mid bDBb,\ D \to d.$$



Définition

Si $\delta C y \underset{\mathrm{rm}}{\Rightarrow} \delta\gamma A \sigma y \underset{\mathrm{rm}}{\Rightarrow}^* \delta\gamma A x y \underset{\mathrm{rm}}{\Rightarrow} \delta\gamma\alpha\varphi x y \underset{\mathrm{rm}}{\Rightarrow}^* \delta\gamma\alpha x y$, alors

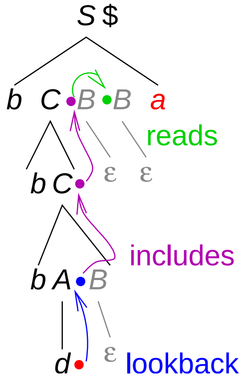
$$\delta\gamma[A \to \alpha \bullet \varphi]xy \textbf{ includes } \delta[C \to \gamma A \bullet \sigma]y.$$

Par exemple, $bb[C \to bA \bullet B]a$ **includes** $b[C \to bC \bullet]a$ et $b[C \to bC \bullet]a$ **includes** $[S \to bC \bullet BBa]$.

Fenêtre sur un arbre

Exemple

$$S \rightarrow CBBa,\ A \rightarrow d,\ B \rightarrow \varepsilon,\ C \rightarrow bC \mid bAB \mid bDBb,\ D \rightarrow d.$$



Définition

Si $\delta C y \underset{\mathrm{rm}}{\Rightarrow} \delta\gamma A\sigma y \underset{\mathrm{rm}}{\Rightarrow}^{*} \delta\gamma A x y \underset{\mathrm{rm}}{\Rightarrow} \delta\gamma x y$, alors

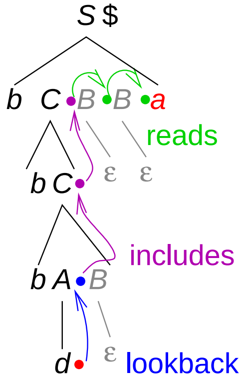
$$\delta[C \rightarrow \gamma \bullet A\sigma]y \textbf{ reads } \delta[C \rightarrow \gamma A \bullet \sigma]y.$$

Par exemple, $[S \rightarrow bC \bullet BBa]$ **reads** $[S \rightarrow bCB \bullet Ba]$.

Fenêtre sur un arbre

Exemple

$$S \rightarrow CBBa,\ A \rightarrow d,\ B \rightarrow \varepsilon,\ C \rightarrow bC \mid bAB \mid bDBb,\ D \rightarrow d.$$



Définition

Si $\delta C y \underset{\mathrm{rm}}{\Rightarrow} \delta\gamma A\sigma y \underset{\mathrm{rm}}{\Rightarrow}^{*} \delta\gamma A x y \underset{\mathrm{rm}}{\Rightarrow} \delta\gamma x y$, alors

$$\delta[C \rightarrow \gamma \bullet A\sigma]y \ \textbf{reads}\ \delta[C \rightarrow \gamma A \bullet \sigma]y.$$

Par exemple, $[S \rightarrow bC \bullet BBa]$ **reads** $[S \rightarrow bCB \bullet Ba]$ et $[S \rightarrow bCB \bullet Ba]$ **reads** $[S \rightarrow bCBB \bullet a]$.

Et LALR(1) ? [DP82]

Théorème

$S \underset{\mathrm{rm}}{\Rightarrow}^* \delta A a x \underset{\mathrm{rm}}{\Rightarrow} \delta \alpha a x$ si et seulement si

$$\delta[A \rightarrow \alpha \bullet] a x \ \textbf{lookback} \ \circ \ \textbf{includes}^* \circ (\ \textbf{reads}^* \circ \overset{\varepsilon}{\curvearrowright}{}^*)^* \gamma [C \rightarrow \beta \bullet a \beta'] y.$$

Et LALR(1) ? [DP82]

Théorème

$S \underset{\mathrm{rm}}{\Rightarrow}^* \delta A a x \underset{\mathrm{rm}}{\Rightarrow} \delta \alpha a x$ si et seulement si

$$\delta[A \to \alpha \bullet] ax \ \textbf{lookback} \ \circ \ \textbf{includes}^* \circ (\ \textbf{reads}^* \circ \overset{\varepsilon}{\curvearrowright}{}^*)^* \gamma [C \to \beta \bullet a \beta'] y.$$

Pour faire un calcul de fenêtre LALR(1), il suffit de faire ces opérations modulo l'équivalence $\equiv_{\mathrm{LR}(0)}$ sur le préfixe de la position.

Et LALR(1) ? [DP82]

Théorème

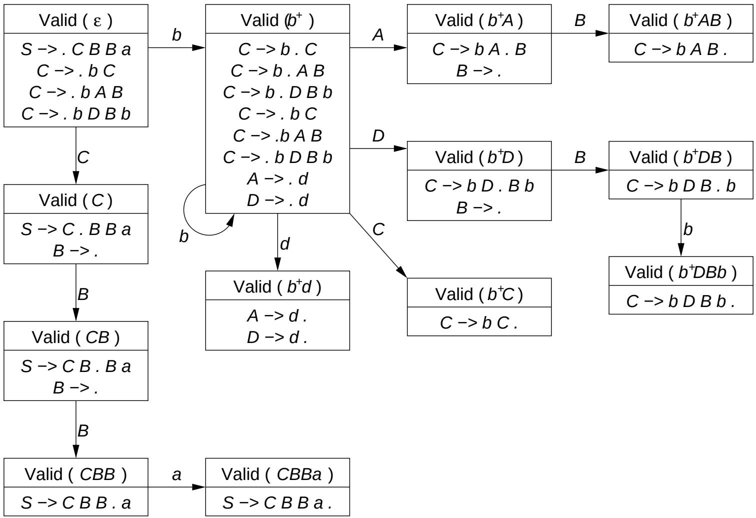
$S \underset{\text{rm}}{\Rightarrow}^* \delta A a x \underset{\text{rm}}{\Rightarrow} \delta \alpha a x$ *si et seulement si*

$$\delta[A \rightarrow \alpha \bullet] a x \ \textbf{lookback} \ \circ \ \textbf{includes}^* \circ (\ \textbf{reads}^* \circ \overset{\varepsilon}{\curvearrowright}^*)^* \gamma [C \rightarrow \beta \bullet a \beta'] y.$$

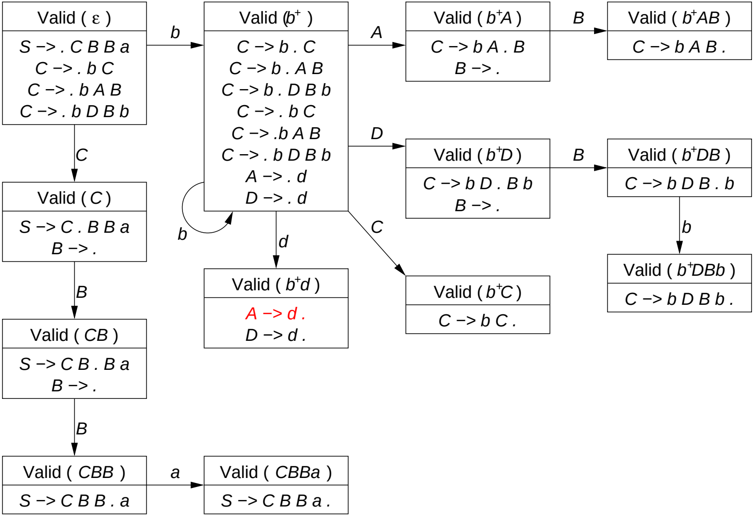
Pour faire un calcul de fenêtre LALR(1), il suffit de faire ces opérations modulo l'équivalence $\equiv_{\mathsf{LR}(0)}$ sur le préfixe de la position. Les relations définies en termes d'états et de transitions LR(0) deviennent :

$$\begin{aligned}
([\delta\alpha], A \rightarrow \alpha) \ &\textbf{lookback} \ ([\delta], A) \\
([\delta\beta], A) \ &\textbf{includes} \ ([\delta], B) \text{ ssi } B \rightarrow \beta A \gamma \text{ et } \gamma \Rightarrow^* \varepsilon, \\
([\delta], A) \ &\textbf{reads} \ ([\delta A], C) \text{ ssi } ([\delta A], C) \text{ et } C \Rightarrow^* \varepsilon.
\end{aligned}$$

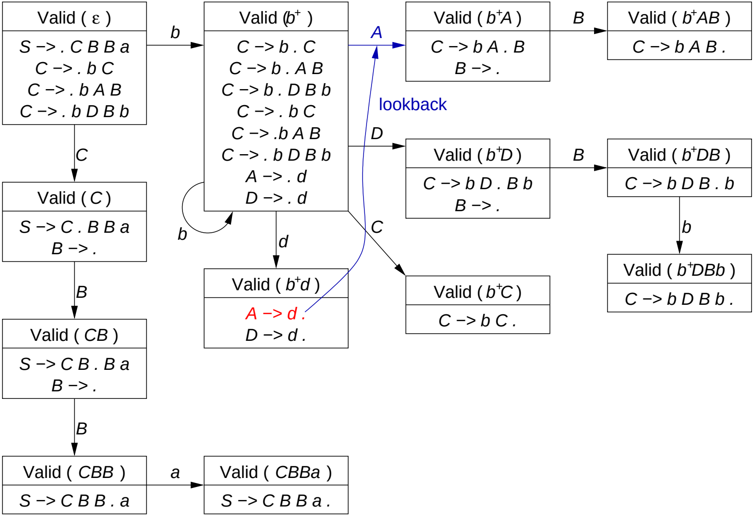
Exemple


Valid (ε)
S −> . C B B a
C −> . b C
C −> . b A B
C −> . b D B b
b
Valid (b+)
C −> b . C
C −> b . A B
C −> b . D B b
C −> . b C
C −> .b A B
C −> . b D B b
A −> . d
D −> . d
A
Valid (b+A)
C −> b A . B
B −> .
B
Valid (b+AB)
C −> b A B .
D
Valid (b+D)
C −> b D . B b
B −> .
B
Valid (b+DB)
C −> b D B . b
b
Valid (b+DBb)
C −> b D B b .
C
Valid (b+C)
C −> b C .
d
Valid (b+d)
A −> d .
D −> d .
b
C
Valid (C)
S −> C . B B a
B −> .
B
Valid (CB)
S −> C B . B a
B −> .
B
Valid (CBB)
S −> C B B . a
a
Valid (CBBa)
S −> C B B a .

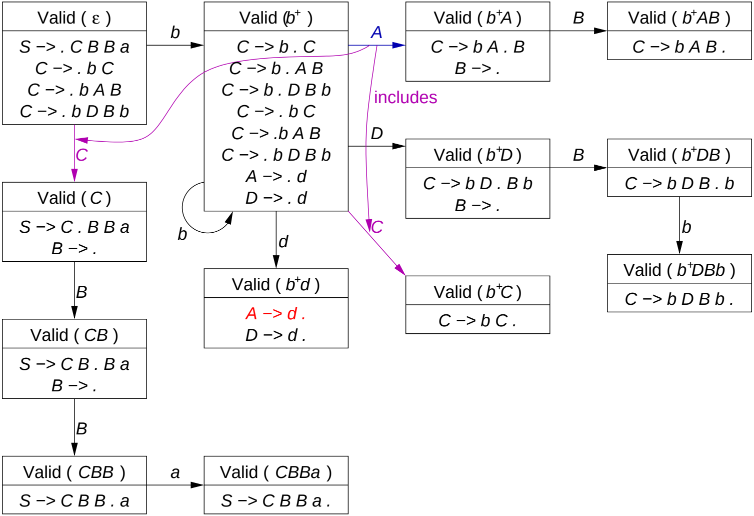
Exemple


Valid (ε)
S −> . C B B a
C −> . b C
C −> . b A B
C −> . b D B b
b
Valid (b⁺)
C −> b . C
C −> b . A B
C −> b . D B b
C −> . b C
C −> .b A B
C −> . b D B b
A −> . d
D −> . d
A
Valid (b⁺A)
C −> b A . B
B −> .
B
Valid (b⁺AB)
C −> b A B .
D
Valid (b⁺D)
C −> b D . B b
B −> .
B
Valid (b⁺DB)
C −> b D B . b
b
Valid (b⁺DBb)
C −> b D B b .
C
Valid (C)
S −> C . B B a
B −> .
b
d
Valid (b⁺d)
A −> d .
D −> d .
C
Valid (b⁺C)
C −> b C .
B
Valid (CB)
S −> C B . B a
B −> .
B
Valid (CBB)
S −> C B B . a
a
Valid (CBBa)
S −> C B B a .

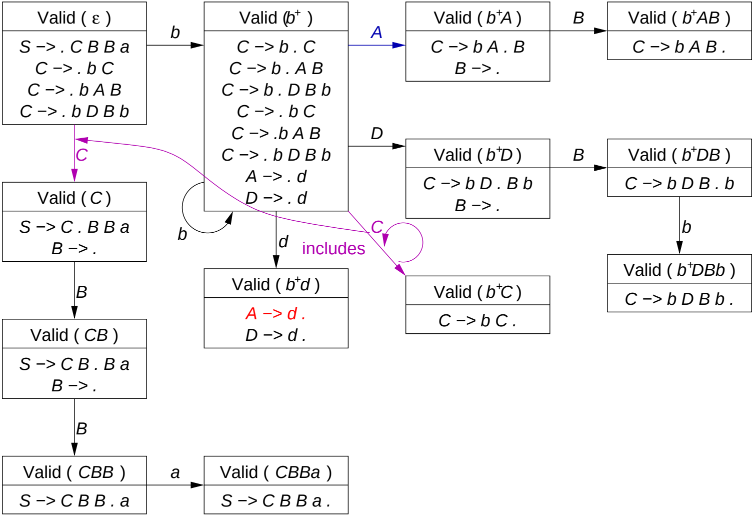
Exemple


Valid (ε)
S –> . C B B a
C –> . b C
C –> . b A B
C –> . b D B b
b
Valid (b+)
C –> b . C
C –> b . A B
C –> b . D B b
C –> . b C
C –> .b A B
C –> . b D B b
A –> . d
D –> . d
A
Valid (b+A)
C –> b A . B
B –> .
B
Valid (b+AB)
C –> b A B .
lookback
D
Valid (b+D)
C –> b D . B b
B –> .
B
Valid (b+DB)
C –> b D B . b
b
Valid (b+DBb)
C –> b D B b .
C
Valid (C)
S –> C . B B a
B –> .
b
d
C
Valid (b+C)
C –> b C .
Valid (b+d)
A –> d .
D –> d .
B
Valid (CB)
S –> C B . B a
B –> .
B
Valid (CBB)
S –> C B B . a
a
Valid (CBBa)
S –> C B B a .

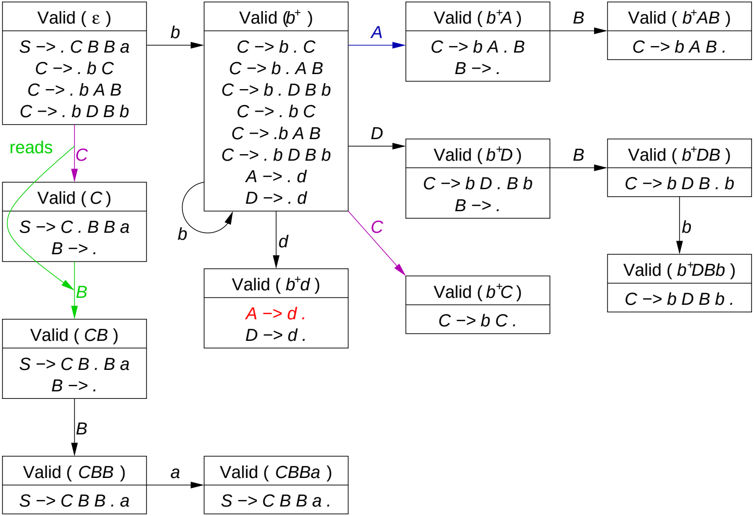
Exemple


Valid (ε)
S –> . C B B a
C –> . b C
C –> . b A B
C –> . b D B b
b
Valid (b+)
C –> b . C
C –> b . A B
C –> b . D B b
C –> . b C
C –> .b A B
C –> . b D B b
A –> . d
D –> . d
A
Valid (b+A)
C –> b A . B
B –> .
B
Valid (b+AB)
C –> b A B .
includes
D
Valid (b+D)
C –> b D . B b
B –> .
B
Valid (b+DB)
C –> b D B . b
b
Valid (b+DBb)
C –> b D B b .
C
Valid (C)
S –> C . B B a
B –> .
b
d
C
Valid (b+d)
A –> d .
D –> d .
Valid (b+C)
C –> b C .
B
Valid (CB)
S –> C B . B a
B –> .
B
Valid (CBB)
S –> C B B . a
a
Valid (CBBa)
S –> C B B a .

Exemple

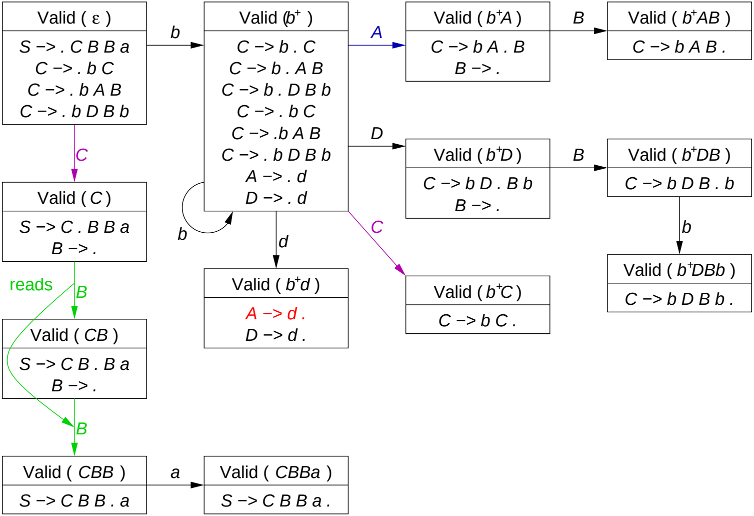

Valid (ε)
S –> . C B B a
C –> . b C
C –> . b A B
C –> . b D B b
b
Valid (b+)
C –> b . C
C –> b . A B
C –> b . D B b
C –> . b C
C –> .b A B
C –> . b D B b
A –> . d
D –> . d
A
Valid (b+A)
C –> b A . B
B –> .
B
Valid (b+AB)
C –> b A B .
D
Valid (b+D)
C –> b D . B b
B –> .
B
Valid (b+DB)
C –> b D B . b
b
Valid (b+DBb)
C –> b D B b .
C
Valid (C)
S –> C . B B a
B –> .
b
d
includes
C
Valid (b+d)
A –> d .
D –> d .
Valid (b+C)
C –> b C .
B
Valid (CB)
S –> C B . B a
B –> .
B
Valid (CBB)
S –> C B B . a
a
Valid (CBBa)
S –> C B B a .

Exemple

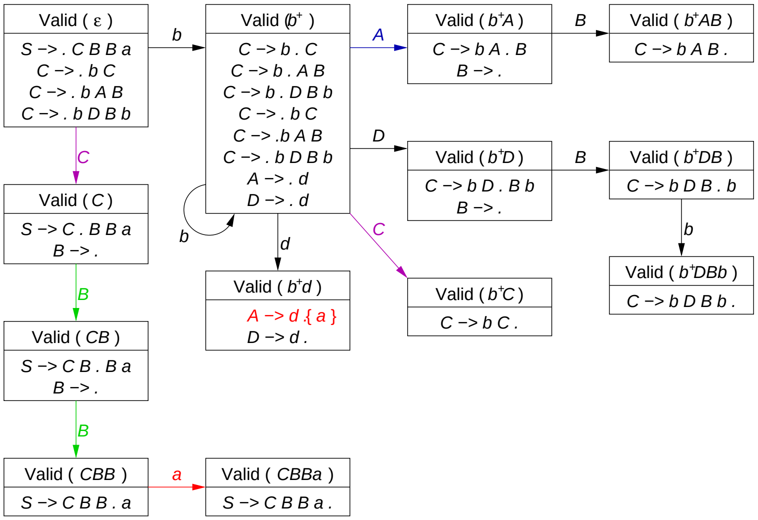


Valid (ε)
S –> . C B B a
C –> . b C
C –> . b A B
C –> . b D B b
b
Valid (b+)
C –> b . C
C –> b . A B
C –> b . D B b
C –> . b C
C –> .b A B
C –> . b D B b
A –> . d
D –> . d
A
Valid (b+A)
C –> b A . B
B –> .
B
Valid (b+AB)
C –> b A B .
D
Valid (b+D)
C –> b D . B b
B –> .
B
Valid (b+DB)
C –> b D B . b
b
Valid (b+DBb)
C –> b D B b .
reads
C
Valid (C)
S –> C . B B a
B –> .
B
Valid (CB)
S –> C B . B a
B –> .
B
Valid (CBB)
S –> C B B . a
a
Valid (CBBa)
S –> C B B a .
b
d
Valid (b+d)
A –> d .
D –> d .
C
Valid (b+C)
C –> b C .

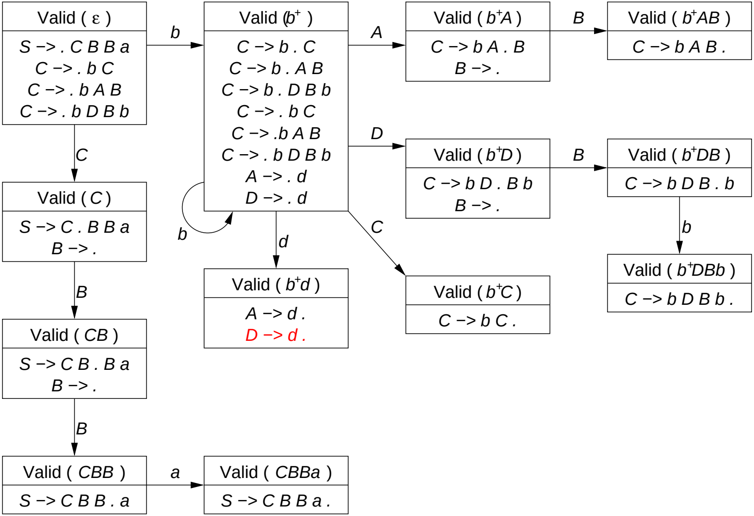
Exemple


Valid (ε)
S –> . C B B a
C –> . b C
C –> . b A B
C –> . b D B b
b
Valid (b+)
C –> b . C
C –> b . A B
C –> b . D B b
C –> . b C
C –> .b A B
C –> . b D B b
A –> . d
D –> . d
A
Valid (b+A)
C –> b A . B
B –> .
B
Valid (b+AB)
C –> b A B .
D
Valid (b+D)
C –> b D . B b
B –> .
B
Valid (b+DB)
C –> b D B . b
b
Valid (b+DBb)
C –> b D B b .
C
Valid (b+C)
C –> b C .
d
Valid (b+d)
A –> d .
D –> d .
b
C
Valid (C)
S –> C . B B a
B –> .
reads
B
Valid (CB)
S –> C B . B a
B –> .
B
Valid (CBB)
S –> C B B . a
a
Valid (CBBa)
S –> C B B a .

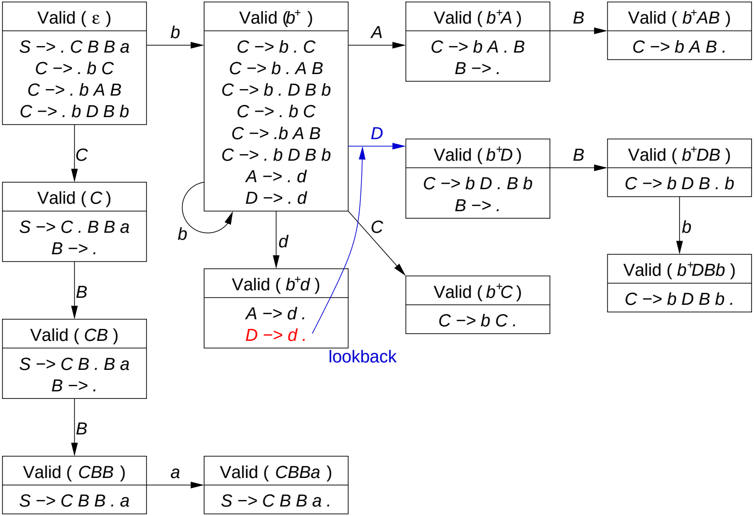
Exemple


Valid (ε)
S –> . C B B a
C –> . b C
C –> . b A B
C –> . b D B b
b
Valid (b⁺)
C –> b . C
C –> b . A B
C –> b . D B b
C –> . b C
C –> .b A B
C –> . b D B b
A –> . d
D –> . d
A
Valid (b⁺A)
C –> b A . B
B –> .
B
Valid (b⁺AB)
C –> b A B .
D
Valid (b⁺D)
C –> b D . B b
B –> .
B
Valid (b⁺DB)
C –> b D B . b
b
Valid (b⁺DBb)
C –> b D B b .
C
Valid (b⁺C)
C –> b C .
d
Valid (b⁺d)
A –> d .{ a }
D –> d .
b
C
Valid (C)
S –> C . B B a
B –> .
B
Valid (CB)
S –> C B . B a
B –> .
B
Valid (CBB)
S –> C B B . a
a
Valid (CBBa)
S –> C B B a .

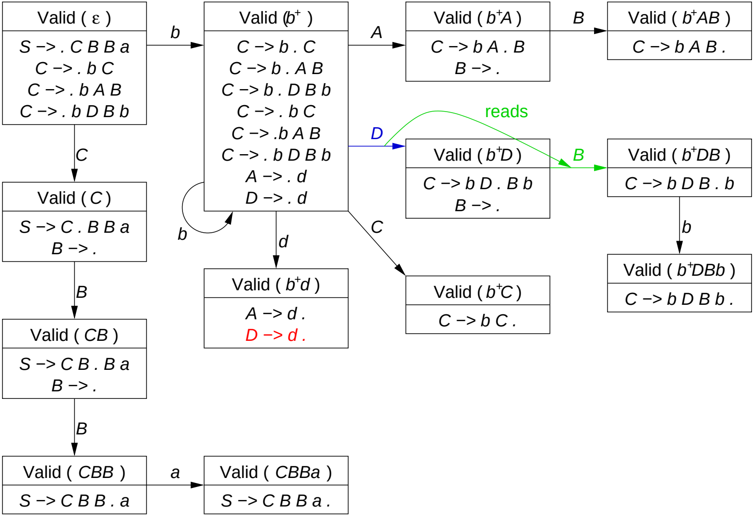
Exemple


Valid (ε)
S –> . C B B a
C –> . b C
C –> . b A B
C –> . b D B b
b
Valid (b+)
C –> b . C
C –> b . A B
C –> b . D B b
C –> . b C
C –> .b A B
C –> . b D B b
A –> . d
D –> . d
A
Valid (b+A)
C –> b A . B
B –> .
B
Valid (b+AB)
C –> b A B .
D
Valid (b+D)
C –> b D . B b
B –> .
B
Valid (b+DB)
C –> b D B . b
b
Valid (b+DBb)
C –> b D B b .
C
Valid (b+C)
C –> b C .
d
Valid (b+d)
A –> d .
D –> d .
C
Valid (C)
S –> C . B B a
B –> .
B
Valid (CB)
S –> C B . B a
B –> .
B
Valid (CBB)
S –> C B B . a
a
Valid (CBBa)
S –> C B B a .

Exemple

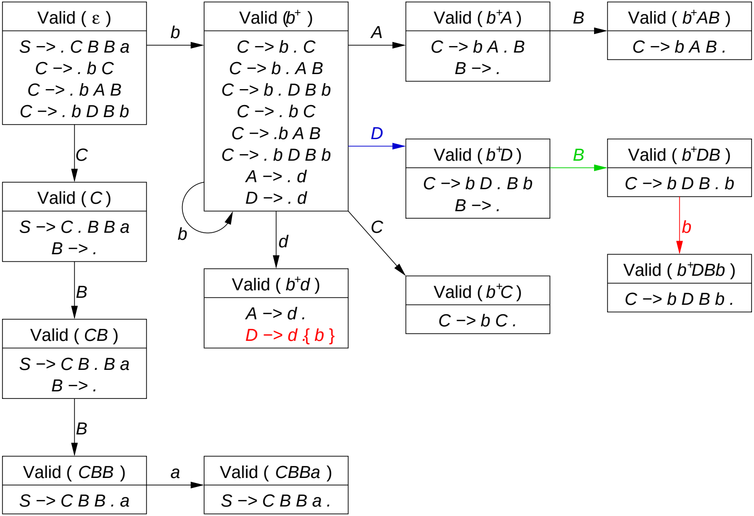

Valid (ε)
S −> . C B B a
C −> . b C
C −> . b A B
C −> . b D B b
b
Valid (b+)
C −> b . C
C −> b . A B
C −> b . D B b
C −> . b C
C −> .b A B
C −> . b D B b
A −> . d
D −> . d
A
Valid (b+A)
C −> b A . B
B −> .
B
Valid (b+AB)
C −> b A B .
D
Valid (b+D)
C −> b D . B b
B −> .
B
Valid (b+DB)
C −> b D B . b
b
Valid (b+DBb)
C −> b D B b .
C
Valid (C)
S −> C . B B a
B −> .
B
Valid (CB)
S −> C B . B a
B −> .
B
Valid (CBB)
S −> C B B . a
a
Valid (CBBa)
S −> C B B a .
d
Valid (b+d)
A −> d .
D −> d .
C
Valid (b+C)
C −> b C .
lookback

Exemple



Valid (ε)
S −> . C B B a
C −> . b C
C −> . b A B
C −> . b D B b
b
Valid (b+)
C −> b . C
C −> b . A B
C −> b . D B b
C −> . b C
C −> .b A B
C −> . b D B b
A −> . d
D −> . d
A
Valid (b+A)
C −> b A . B
B −> .
B
Valid (b+AB)
C −> b A B .
reads
D
Valid (b+D)
C −> b D . B b
B −> .
B
Valid (b+DB)
C −> b D B . b
b
Valid (b+DBb)
C −> b D B b .
C
Valid (C)
S −> C . B B a
B −> .
b
d
Valid (b+d)
A −> d .
D −> d .
C
Valid (b+C)
C −> b C .
B
Valid (CB)
S −> C B . B a
B −> .
B
Valid (CBB)
S −> C B B . a
a
Valid (CBBa)
S −> C B B a .

Exemple



Valid (ε)
S –> . C B B a
C –> . b C
C –> . b A B
C –> . b D B b
b
Valid (b+)
C –> b . C
C –> b . A B
C –> b . D B b
C –> . b C
C –> .b A B
C –> . b D B b
A –> . d
D –> . d
A
Valid (b+A)
C –> b A . B
B –> .
B
Valid (b+AB)
C –> b A B .
D
Valid (b+D)
C –> b D . B b
B –> .
B
Valid (b+DB)
C –> b D B . b
b
Valid (b+DBb)
C –> b D B b .
C
Valid (b+C)
C –> b C .
d
Valid (b+d)
A –> d .
D –> d .{ b }
C
Valid (C)
S –> C . B B a
B –> .
B
Valid (CB)
S –> C B . B a
B –> .
B
Valid (CBB)
S –> C B B . a
a
Valid (CBBa)
S –> C B B a .

En résumé

- Le calcul du contexte droit d'une position n'est pas forcément un calcul LR(k) précis, il peut utiliser une équivalence.
- Cette relation d'équivalence est la relation $\equiv_{\mathrm{LR}(0)}$ pour LALR(1).
- Cette approximation est assez précise pour analyser tous les langages déterministes [DeR69].
- On peut imaginer d'autres approximations [Kor69, And72]...

En résumé

- Le calcul du contexte droit d'une position n'est pas forcément un calcul LR(k) précis, il peut utiliser une équivalence.
- Cette relation d'équivalence est la relation $\equiv_{\mathrm{LR}(0)}$ pour LALR(1).
- Cette approximation est assez précise pour analyser tous les langages déterministes [DeR69].
- On peut imaginer d'autres approximations [Kor69, And72]...

En résumé

- Le calcul du contexte droit d'une position n'est pas forcément un calcul LR(k) précis, il peut utiliser une équivalence.
- Cette relation d'équivalence est la relation $\equiv_{\mathrm{LR}(0)}$ pour LALR(1).
- Cette approximation est assez précise pour analyser tous les langages déterministes [DeR69].
- On peut imaginer d'autres approximations [Kor69, And72]...

En résumé

- Le calcul du contexte droit d'une position n'est pas forcément un calcul LR(k) précis, il peut utiliser une équivalence.
- Cette relation d'équivalence est la relation $\equiv_{\mathsf{LR}(0)}$ pour LALR(1).
- Cette approximation est assez précise pour analyser tous les langages déterministes [DeR69].
- On peut imaginer d'autres approximations [Kor69, And72]...

En résumé

- Le calcul du contexte droit d'une position n'est pas forcément un calcul LR(k) précis, il peut utiliser une équivalence.
- Cette relation d'équivalence est la relation $\equiv_{\mathrm{LR}(0)}$ pour LALR(1).
- Cette approximation est assez précise pour analyser tous les langages déterministes [DeR69].
- On peut imaginer d'autres approximations [Kor69, And72]...

Limites de LALR

Les limites de LALR sont les limites de l'approximation par LR(0).

Exercice

Donner l'automate LALR(1) pour la grammaire

$$S \rightarrow aAa \,|\, bAb \,|\, aBb \,|\, bBa, \;\; A \rightarrow c, \;\; B \rightarrow c.$$

T. Anderson.
Syntactic analysis of LR(k) languages.
PhD thesis, Department of Computing Science, University of Newcastle upon Tyne, 1972.

Franklin Lewis DeRemer.
Practical Translators for LR(k) Languages.
PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1969.

Frank DeRemer and Thomas Pennello.
Efficient computation of LALR(1) look-ahead sets.
ACM Transactions on Programming Languages and Systems (TOPLAS), 4(4):615–649, 1982.

Bent Bruun Kristensen and Ole Lehrmann Madsen.
Methods for computing LALR(k) lookahead.
ACM Transactions on Programming Languages and Systems, 3(1):60–82, 1981.

A. J. Korenjak.