Analyse LALR

2005/2006

1 Rappels

1.1 Analyse LR

Analyse ascendante LR

Exemple 1.1. Soit la grammaire

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

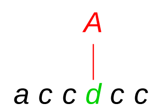
a c c d c c

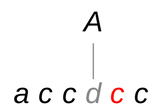
a c c d c c

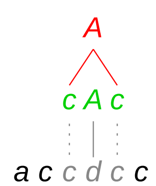
a c c d c c

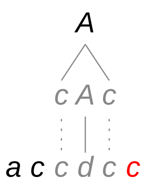
a c c d c c

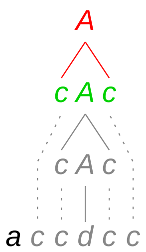
a c c d c c

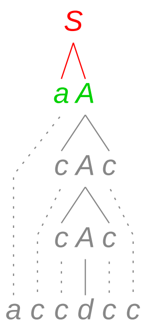


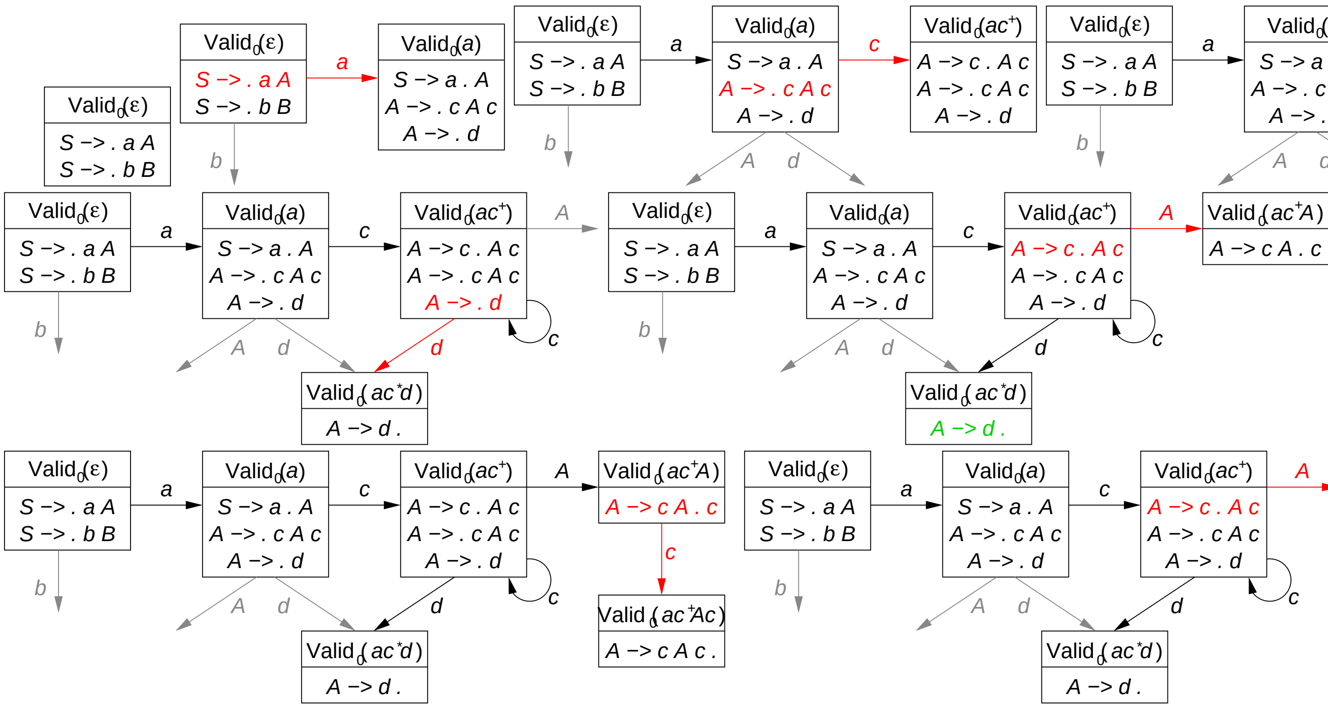


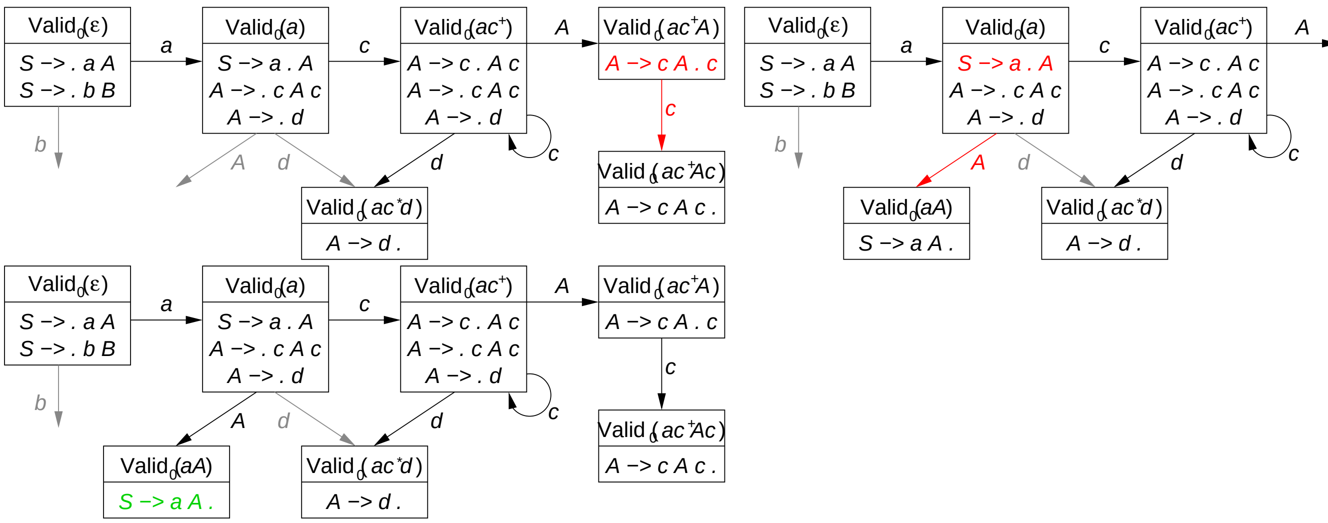



A
c A c
a c c d c c


A
c A c
c A c
a c c d c c


S
a A
c A c
c A c
a c c d c c


Valid0(ε)
S −> . a A
S −> . b B
Valid0(a)
S −> a . A
A −> . c A c
A −> . d
Valid0(ac+)
A −> c . A c
A −> . c A c
A −> . d
Valid0(ac*d)
A −> d .
Valid0(ac+A)
A −> c A . c
Valid0(ac+Ac)
A −> c A c .



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

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 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) :
Définition 1.2. 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.

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

1.2 Le problème

Le problème

Exercice 1.3. 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) ?

2 Analyse LALR

2.1 Principe de LALR

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

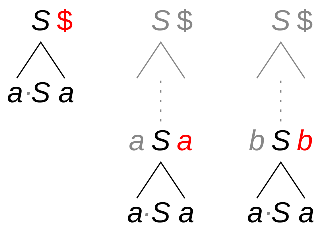
sont fusionnés en un seul $\boxed{\begin{array}{l} S\to a\bullet Sa,\{\$,a,b\} \\ S\to\bullet aSa,\{a\} \\ S\to\bullet bSb,\{a\} \\ S\to\bullet c,\{a\} \end{array}}$.

Définition 2.1 ([?])**.** La fenêtre LALR(k) d'un corps valide pour un préfixe γ est

$$LA([\gamma], A\to\alpha\bullet\alpha') = \{k : z \mid S \underset{\text{rm}}{\Rightarrow}^* \delta A x \underset{\text{rm}}{\Rightarrow} \delta\alpha\alpha' x \underset{\text{rm}}{\Rightarrow}^* \delta\alpha z \text{ et } \delta\alpha \equiv_{\text{LR}(0)} \gamma\}. \tag{1}$$

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* :

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

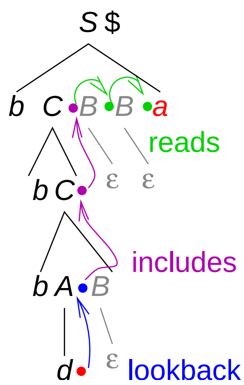
2.2 Calcul d'une fenêtre

Fenêtre sur un arbre

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

Exemple 2.2.

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



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

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

Définition 2.3. 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

$$\delta\gamma[A{\rightarrow}\alpha\bullet]xy \ \textbf{lookback}\ \delta[C{\rightarrow}\gamma A\bullet\sigma]y. \tag{2}$$

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

Définition 2.4. 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

$$\delta\gamma[A{\rightarrow}\alpha\bullet\varphi]xy \ \textbf{includes}\ \delta[C{\rightarrow}\gamma A\bullet\sigma]y. \tag{3}$$

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

Définition 2.5. Si $\delta Cy \underset{\mathrm{rm}}{\Rightarrow} \delta\gamma A\sigma y \underset{\mathrm{rm}}{\Rightarrow}^* \delta\gamma Axy \underset{\mathrm{rm}}{\Rightarrow} \delta\gamma xy$, alors

$$\delta[C{\rightarrow}\gamma\bullet A\sigma]y \ \textbf{reads}\ \delta[C{\rightarrow}\gamma A\bullet\sigma]y. \tag{4}$$

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) ? [?]

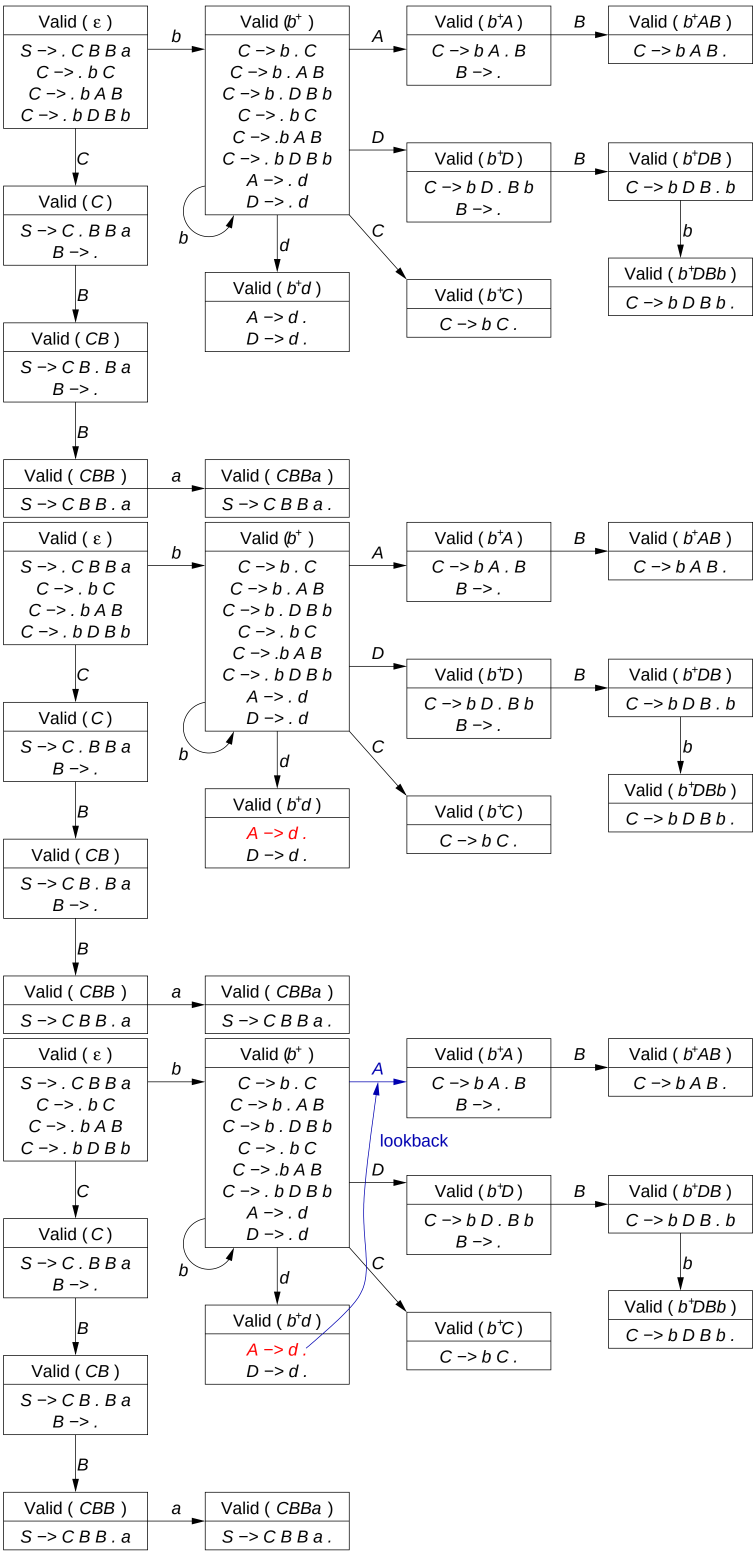
Théorème 2.6. *$S \underset{\mathrm{rm}}{\Rightarrow}^* \delta Aax \underset{\mathrm{rm}}{\Rightarrow} \delta\alpha ax$ si et seulement si*

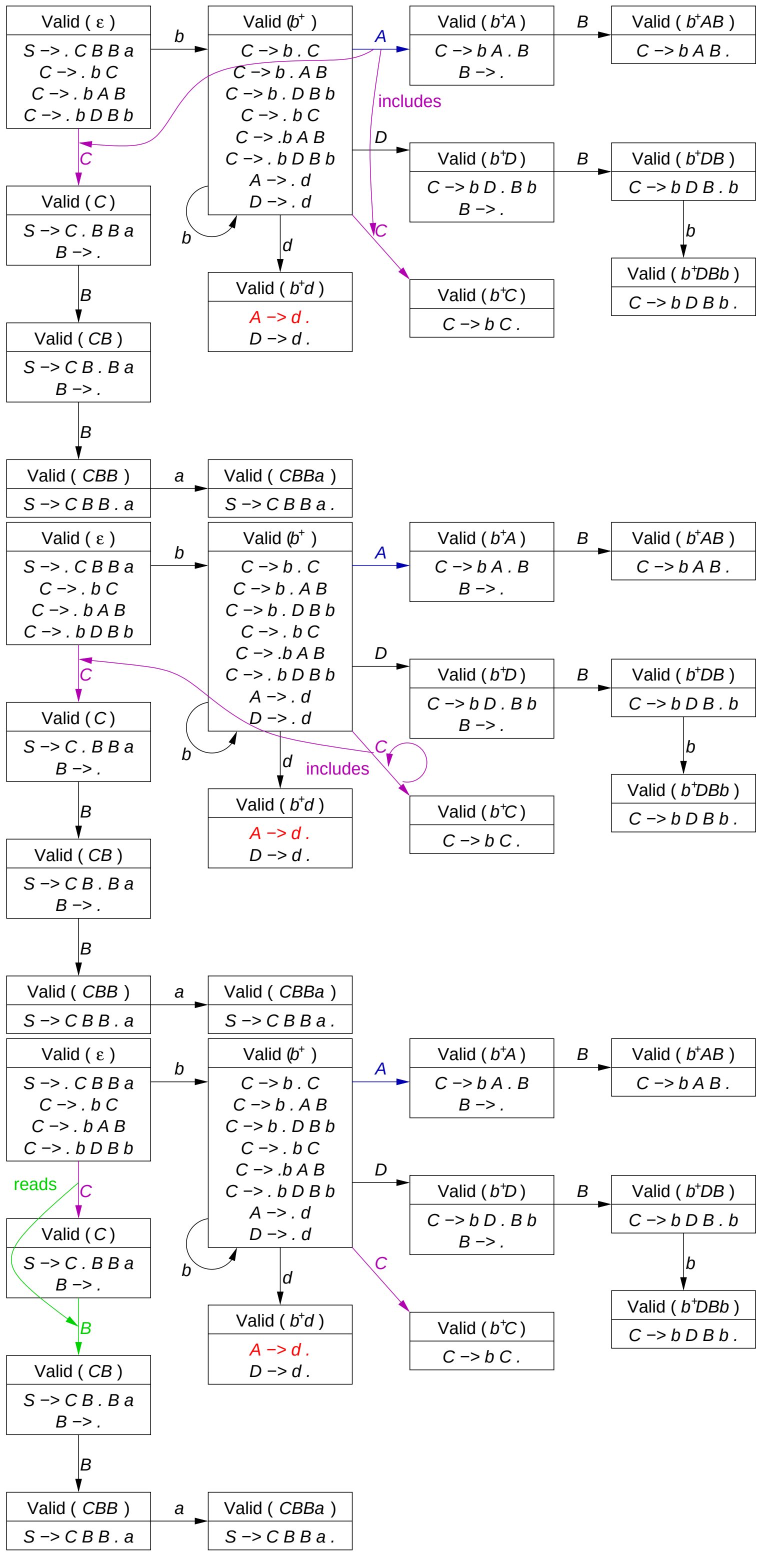
$$\delta[A{\rightarrow}\alpha\bullet]ax \ \textbf{\textit{lookback}} \circ \textbf{\textit{includes}}^* \circ (\ \textbf{\textit{reads}}^* \circ \overset{\varepsilon}{\curvearrowright}^*)^* \gamma[C{\rightarrow}\beta\bullet a\beta']y. \tag{5}$$

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. Les relations définies en termes d'états et de transitions LR(0) deviennent :

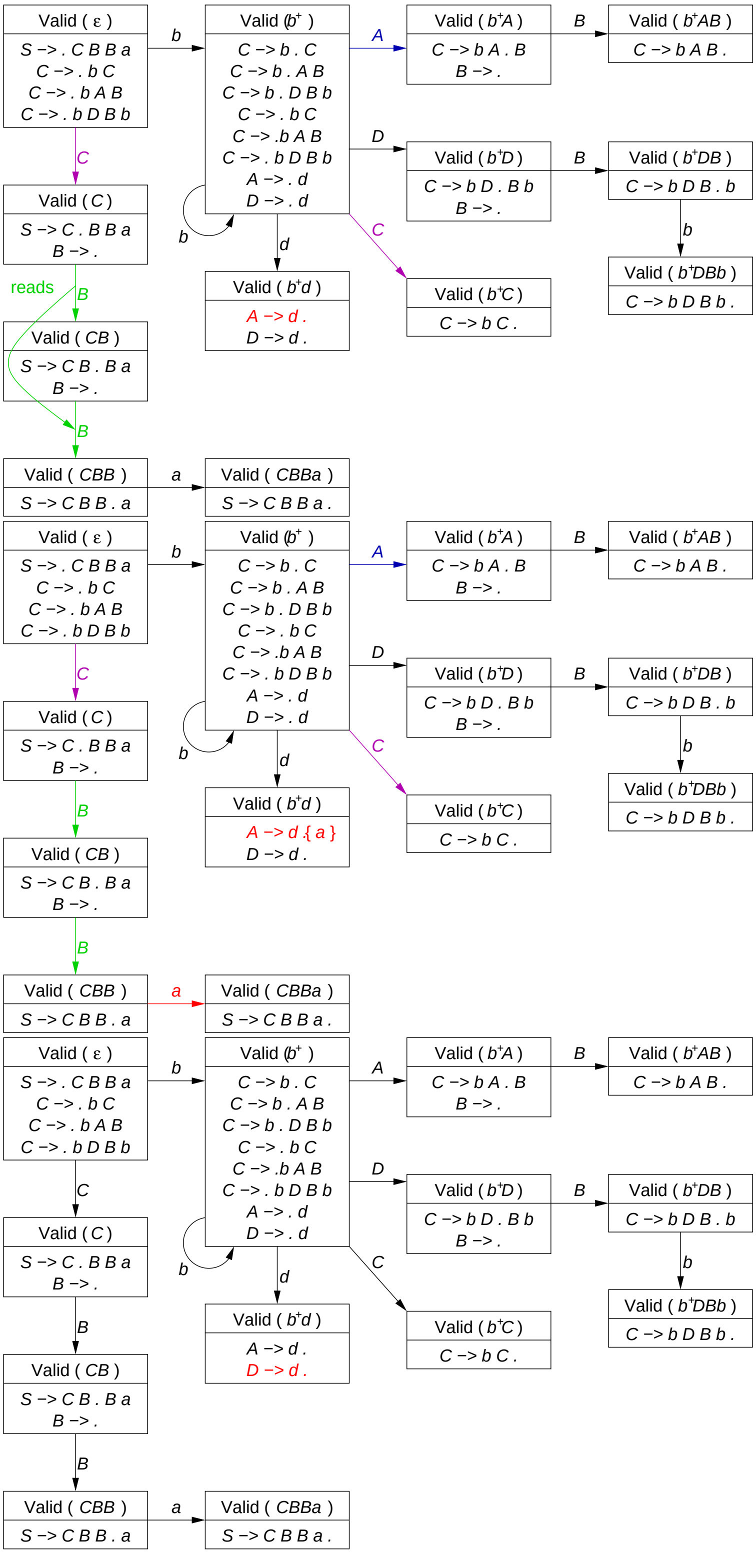
$$\begin{aligned}
([\delta\alpha], A{\rightarrow}\alpha)\ &\textbf{lookback}\ ([\delta], A)\\
([\delta\beta], A)\ &\textbf{includes}\ ([\delta], B) \text{ ssi } B{\rightarrow}\beta A\gamma \text{ et } \gamma\Rightarrow^*\varepsilon,\\
([\delta], A)\ &\textbf{reads}\ ([\delta A], C) \text{ ssi } ([\delta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

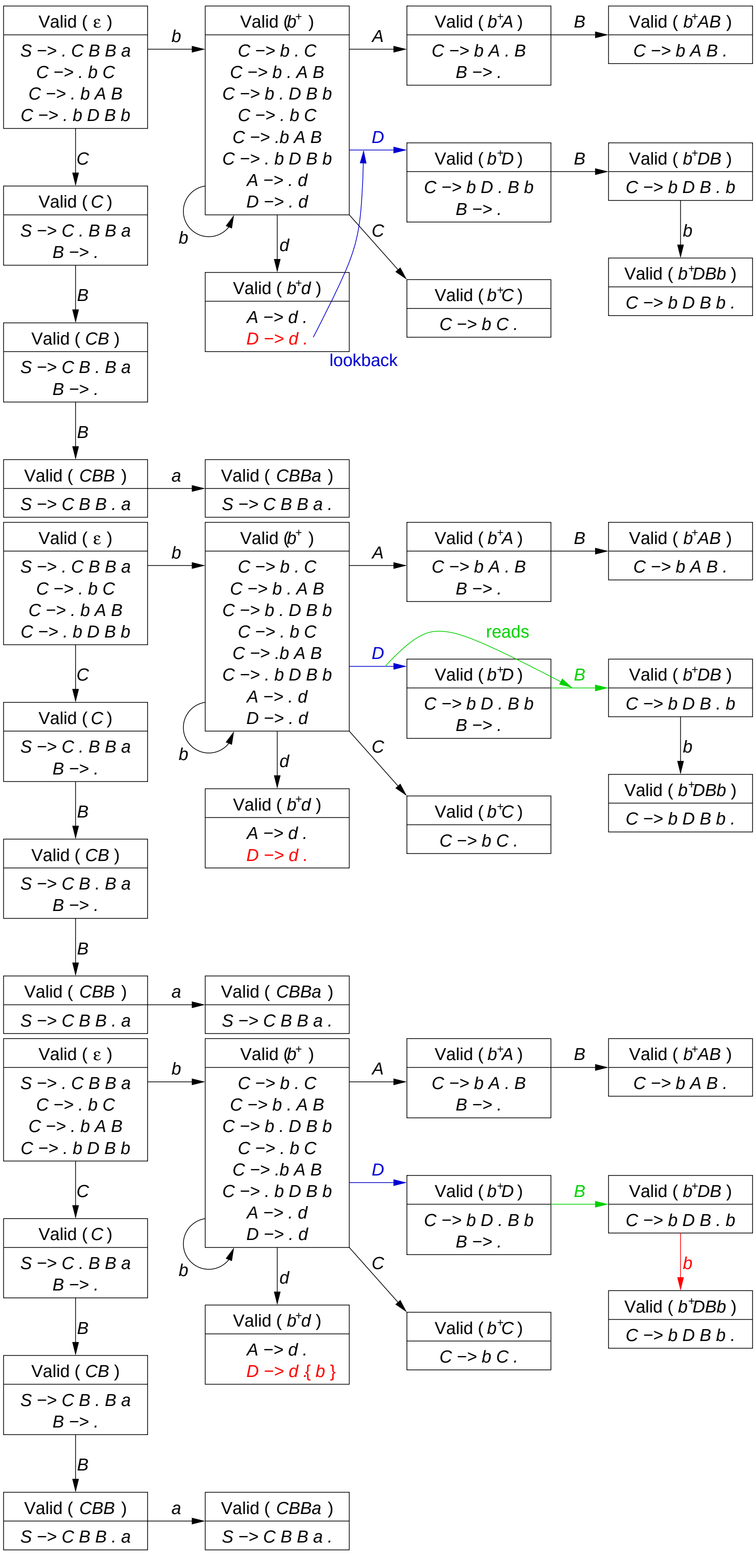
Exemple


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



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 (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 .
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
includes
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 .
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 .
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 .





3 Conclusion

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 [**?**].
– On peut imaginer d'autres approximations [**?**, **?**]...

Limites de LALR

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

Exercice 3.1. Donner l'automate LALR(1) pour la grammaire

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