Script **generated by TTT**

Title: Petter: Compilerbau (29.06.2017)

Date: Thu Jun 29 14:16:10 CEST 2017

Duration: 106:52 min

Pages: 34

LR(1)-Parsing

Idea: Let's equip items with 1-lookahead

Definition LR(1)-Item

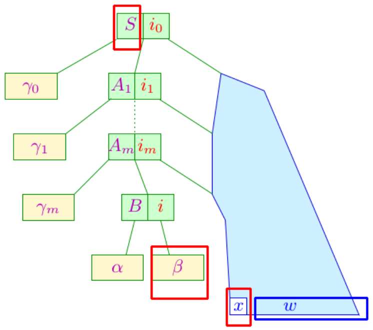
An $LR(1)$-item is a pair $[B \to \alpha \bullet \beta,\, x]$ with

$$x \in \text{Follow}_1(B) = \bigcup \{\text{First}_1(\nu) \mid S \to^* \mu\, B\, \nu\}$$

Admissible LR(1)-Items

The item $[B \to \alpha \bullet \beta,\, x]$ is *admissable* for $\gamma\,\alpha$ if:

$$S \to_R^* \gamma\, B\, w \qquad \text{with} \qquad \{x\} = \text{First}_1(w)$$



... with $\gamma_0 \ldots \gamma_m = \gamma$

The Characteristic LR(1)-Automaton

The set of admissible $LR(1)$-items for viable prefixes is again computed with the help of the finite automaton $c(G, 1)$.

The automaton $c(G, 1)$:

States: $LR(1)$-items

Start state: $[S' \to \bullet\, S,\, \epsilon]$

Final states: $\{[B \to \gamma \bullet,\, x] \mid B \to \gamma \in P, x \in \text{Follow}_1(B)\}$

Transitions:

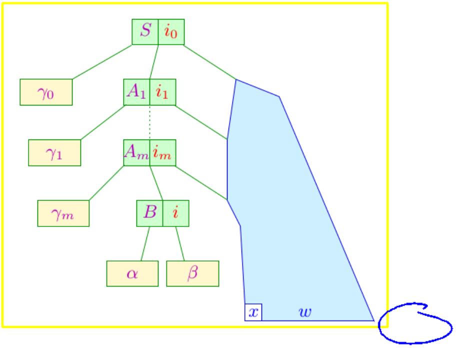
(1) $([A \to \alpha \bullet X\, \beta,\, x], X, [A \to \alpha\, X \bullet \beta,\, x]), \quad X \in (N \cup T)$

(2) $([A \to \alpha \bullet B\, \beta,\, x], \epsilon, [B \to \bullet\, \gamma,\, x']),$
$A \to \alpha\, B\, \beta\,,\ B \to \gamma \in P,$
$x' \in \text{First}_1(\beta) \odot_1 \{x\}$

Admissible LR(1)-Items

The item $[B \to \alpha \bullet \beta,\, x]$ is *admissable* for $\gamma\,\alpha$ if:

$S \to^*_R \gamma\, B\, w \quad$ with $\quad \{x\} = \text{First}_1(w)$



... with $\gamma_0 \ldots \gamma_m = \gamma$

The Characteristic LR(1)-Automaton

The set of admissible $LR(1)$-items for viable prefixes is again computed with the help of the finite automaton $c(G, 1)$.

The automaton $c(G, 1)$:

States: $LR(1)$-items

Start state: $[S' \to \bullet\, S,\, \epsilon]$

Final states: $\{[B \to \gamma \bullet,\, x] \mid B \to \gamma \in P, x \in \text{Follow}_1(B)\}$

Transitions:

(1) $([A \to \alpha \bullet X\,\beta,\, x], X, [A \to \alpha\, X \bullet \beta,\, x]), \quad X \in (N \cup T)$

(2) $([A \to \alpha \bullet B\,\beta,\, x], \epsilon, [B \to \bullet\,\gamma,\, x']),$
$A \to \alpha\, B\, \beta\,,\; B \to \gamma \in P,$
$x' \in \text{First}_1(\beta) \odot_1 \{x\}$

The Canonical LR(1)-Automaton

The canonical $LR(1)$-automaton $LR(G, 1)$ is created from $c(G, 1)$, by performing arbitrarily many ϵ-transitions and then making the resulting automaton deterministic ...

The Canonical LR(1)-Automaton

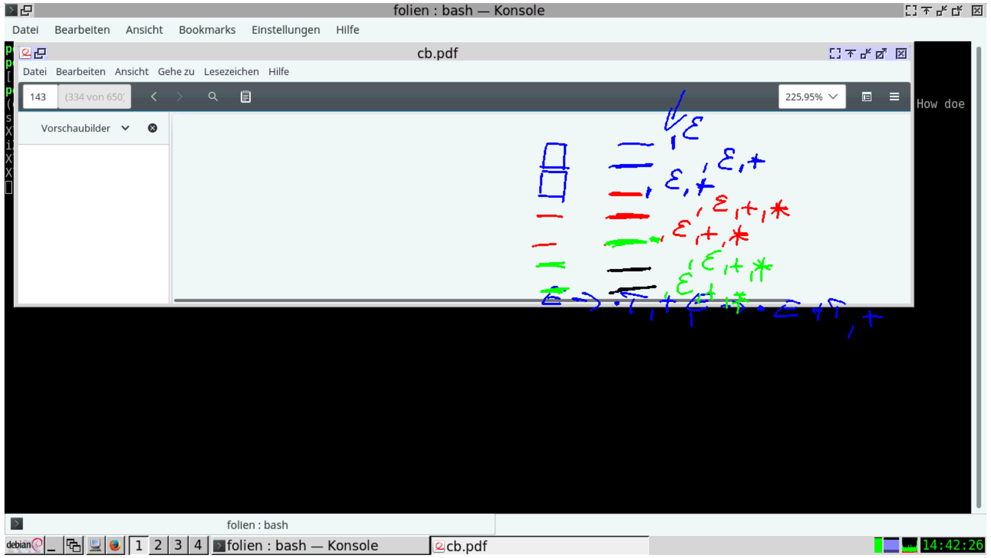
For example:

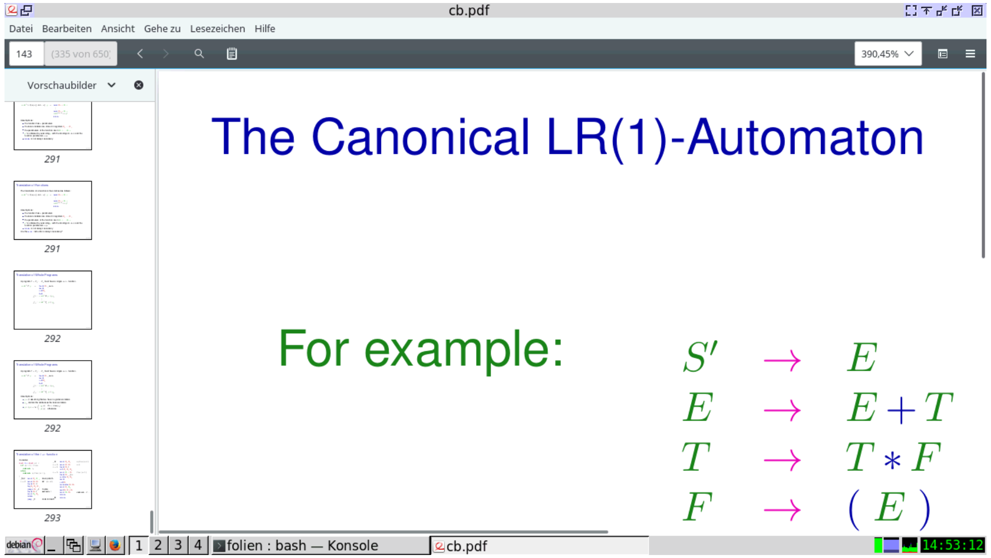
$$
\begin{array}{lcll}
S' & \to & E & \\
E & \to & E + T & \mid\; T \\
T & \to & T * F & \mid\; F \\
F & \to & (\,E\,) & \mid\; \text{int}
\end{array}
$$

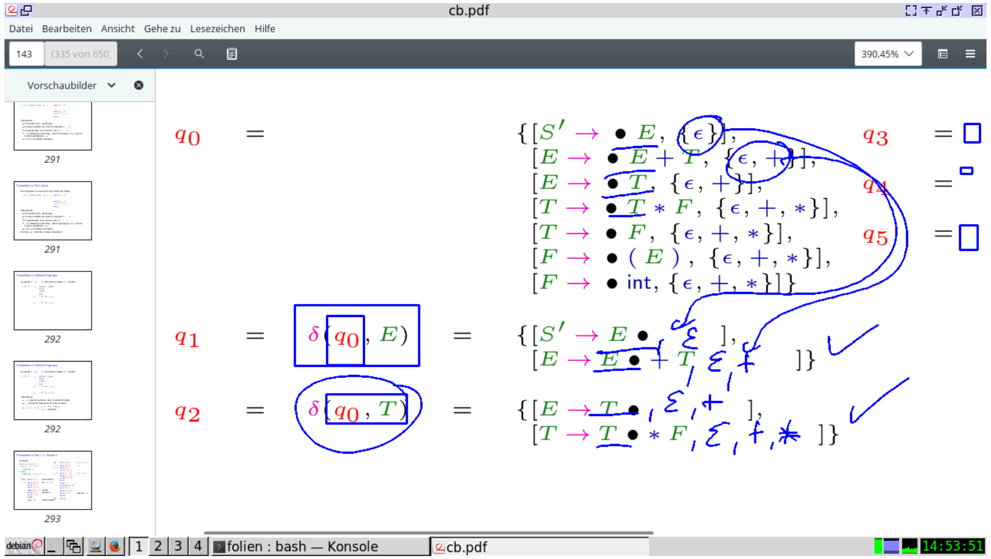
$\text{First}_1(S') = \text{First}_1(E) = \text{First}_1(T) = \text{First}_1(F) = \text{name}, \text{int}, ($

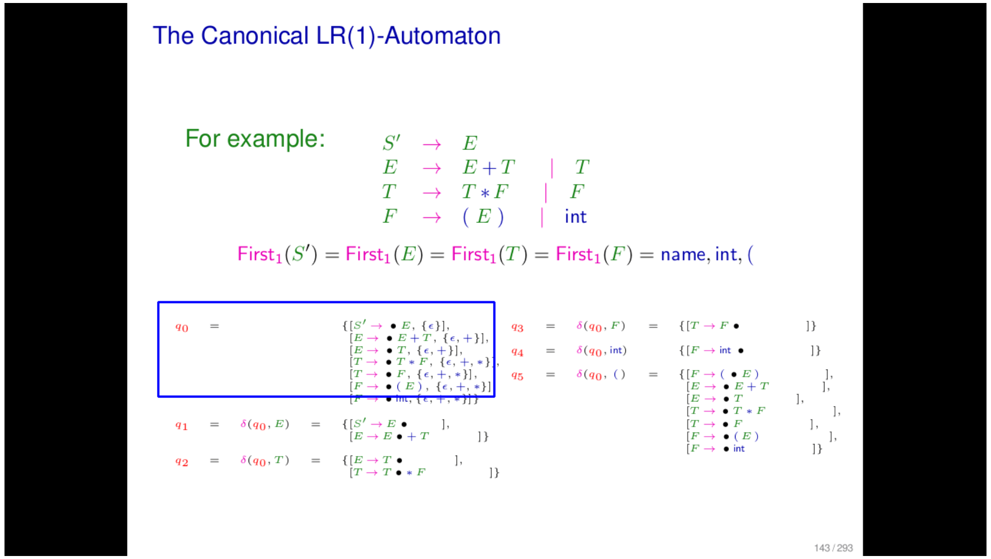
$$
\begin{array}{lllll}
q_0 & = & & & \{[S' \to \bullet\, E \quad], \\
& & & & \;[E \to \bullet\, E + T \quad], \\
& & & & \;[E \to \bullet\, T \quad], \\
& & & & \;[T \to \bullet\, T * F \quad], \\
& & & & \;[T \to \bullet\, F \quad], \\
& & & & \;[F \to \bullet\, (\,E\,) \quad], \\
& & & & \;[F \to \bullet\, \text{int} \quad]\} \\
q_1 & = & \delta(q_0, E) & = & \{[S' \to E\, \bullet \quad], \\
& & & & \;[E \to E \bullet + T \quad]\} \\
q_2 & = & \delta(q_0, T) & = & \{[E \to T\, \bullet \quad], \\
& & & & \;[T \to T \bullet * F \quad]\}
\end{array}
$$

$$
\begin{array}{lllll}
q_3 & = & \delta(q_0, F) & = & \{[T \to F\, \bullet \quad]\} \\
q_4 & = & \delta(q_0, \text{int}) & & \{[F \to \text{int}\, \bullet \quad]\} \\
q_5 & = & \delta(q_0, (\,) & = & \{[F \to (\, \bullet\, E\,) \quad], \\
& & & & \;[E \to \bullet\, E + T \quad], \\
& & & & \;[E \to \bullet\, T \quad], \\
& & & & \;[T \to \bullet\, T * F \quad], \\
& & & & \;[T \to \bullet\, F \quad], \\
& & & & \;[F \to \bullet\, (\,E\,) \quad], \\
& & & & \;[F \to \bullet\, \text{int} \quad]\}
\end{array}
$$


folien : bash — Konsole
Datei Bearbeiten Ansicht Bookmarks Einstellungen Hilfe
cb.pdf
Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe
143 (334 von 650)
225,95%
Vorschaubilder
How doe
folien : bash
1 2 3 4 folien : bash — Konsole cb.pdf
14:42:26


cb.pdf
Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe
143 (335 von 650)
390,45%
Vorschaubilder
291
291
292
292
293
The Canonical LR(1)-Automaton
For example:
S' → E
E → E + T
T → T ∗ F
F → (E)
1 2 3 4 folien : bash — Konsole cb.pdf
14:53:12


cb.pdf
Datei Bearbeiten Ansicht Gehe zu Lesezeichen Hilfe
143 (335 von 650)
390,45%
Vorschaubilder
291
291
292
292
293
q0 = {[S' → • E, {ε}],
[E → • E + T, {ε, +}],
[E → • T, {ε, +}],
[T → • T ∗ F, {ε, +, ∗}],
[T → • F, {ε, +, ∗}],
[F → • (E), {ε, +, ∗}],
[F → • int, {ε, +, ∗}]}
q3 =
q4 =
q5 =
q1 = δ(q0, E) = {[S' → E •,],
[E → E • + T,]}
q2 = δ(q0, T) = {[E → T •,],
[T → T • ∗ F,]}
1 2 3 4 folien : bash — Konsole cb.pdf
14:53:51


The Canonical LR(1)-Automaton
For example:
S' → E
E → E + T | T
T → T ∗ F | F
F → (E) | int
First1(S') = First1(E) = First1(T) = First1(F) = name, int, (
q0 = {[S' → • E, {ε}],
[E → • E + T, {ε, +}],
[E → • T, {ε, +}],
[T → • T ∗ F, {ε, +, ∗}],
[T → • F, {ε, +, ∗}],
[F → • (E), {ε, +, ∗}],
[F → • int, {ε, +, ∗}]}
q1 = δ(q0, E) = {[S' → E •,],
[E → E • + T]}
q2 = δ(q0, T) = {[E → T •,],
[T → T • ∗ F]}
q3 = δ(q0, F) = {[T → F •]}
q4 = δ(q0, int) = {[F → int •]}
q5 = δ(q0, ( ) = {[F → (• E)],
[E → • E + T],
[E → • T],
[T → • T ∗ F],
[T → • F],
[F → • (E)],
[F → • int]}
143 / 293

The Canonical LR(1)-Automaton

For example:

$$\begin{array}{lcll} S' & \to & E \\ E & \to & E+T & \mid \; T \\ T & \to & T*F & \mid \; F \\ F & \to & (\,E\,) & \mid \; \text{int} \end{array}$$

$\text{First}_1(S') = \text{First}_1(E) = \text{First}_1(T) = \text{First}_1(F) = \text{name}, \text{int}, ($

$q'_2 = \delta(q'_5, T) = \{[E \to T\bullet, \{\,)\,, +\}], [T \to T \bullet * F, \{\,)\,, +, *\}]\}$

$q'_3 = \delta(q'_5, F) = \{[T \to F\bullet, \{\,)\,, +, *\}]\}$

$q'_4 = \delta(q'_5, \text{int}) = \{[F \to \text{int}\bullet, \{\,)\,, +, *\}]\}$

$q'_6 = \delta(q_8, +) = \{[E \to E + \bullet T, \{\,)\,, +\}], [T \to \bullet T * F, \{\,)\,, +, *\}], [T \to \bullet F, \{\,)\,, +, *\}], [F \to \bullet (\,E\,), \{\,)\,, +, *\}], [F \to \bullet \text{int}, \{\,)\,, +, *\}]\}$

$q'_7 = \delta(q_9, *) = \{[T \to T * \bullet F, \{\,)\,, +, *\}], [F \to \bullet (\,E\,), \{\,)\,, +, *\}], [F \to \bullet \text{int}, \{\,)\,, +, *\}]\}$

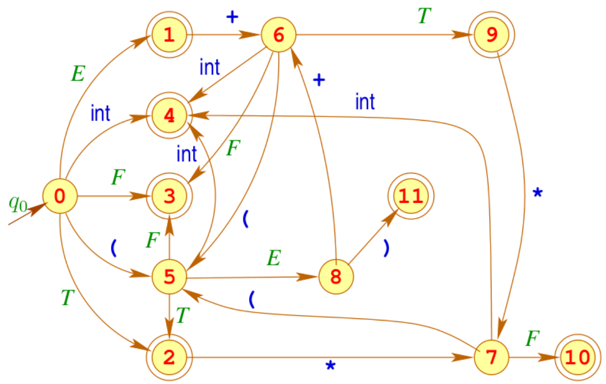
$q'_8 = \delta(q'_5, E) = \{[F \to (\,E \bullet\,), \{\,)\,, +, *\}]\} \; [E \to E \bullet + T, \{\,)\,, +\}]\}$

$q'_9 = \delta(q'_6, T) = \{[E \to E + T\bullet, \{\,)\,, +\}], [T \to T \bullet * F, \{\,)\,, +, *\}]\}$

$q'_{10} = \delta(q'_7, F) = \{[T \to T * F\bullet, \{\,)\,, +, *\}]\}$

$q'_{11} = \delta(q'_8, \,)\,) = \{[F \to (\,E\,)\bullet, \{\,)\,, +, *\}]\}$

The Canonical LR(1)-Automaton



The Canonical LR(1)-Automaton

Discussion:

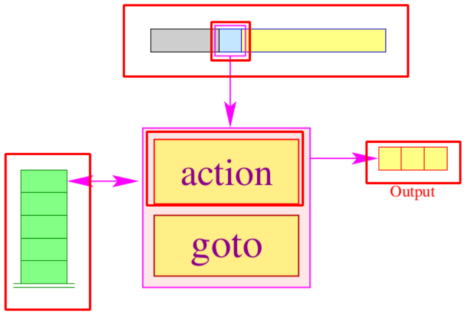
- In the example, the number of states was almost doubled
  ... and it can become even worse
- The conflicts in states q_1, q_2, q_9 are now resolved !
  e.g. we have for:

$$q_9 = \{[E \to E + T\bullet, \{\epsilon, +\}], [T \to T \bullet * F, \{\epsilon, +, *\}]\}$$

with:

$$\{\epsilon, +\} \cap (\text{First}_1(*F) \odot_1 \{\epsilon, +, *\}) = \{\epsilon, +\} \cap \{*\} = \emptyset$$

The LR(1)-Parser:



- The goto-table encodes the transitions:

$$\text{goto}[q, X] = \delta(q, X) \quad \in \quad Q$$

- The action-table describes for every state q and possible lookahead w the necessary action.

The LR(1)-Parser:

The construction of the $LR(1)$-parser:

States: $Q \cup \{f\}$ (f fresh)
Start state: q_0
Final state: f

Transitions:

Shift: $(p, a, p\,q)$ if $q = \text{goto}[q, a]$, $s = \text{action}[p, w]$

Reduce: $(p\,q_1 \ldots q_{|\beta|}, \epsilon, p\,q)$ if $[A \to \beta\,\bullet] \in q_{|\beta|}$, $q = \text{goto}(p, A)$, $[A \to \beta\,\bullet] = \text{action}[q_{|\beta|}, w]$

Finish: $(q_0\,p, \epsilon, f)$ if $[S' \to S\bullet] \in p$

with $LR(G, 1) = (Q, T, \delta, q_0, F)$.

The LR(1)-Parser:

Possible actions are:

- **shift** // Shift-operation
- **reduce** $(A \to \gamma)$ // Reduction with callback/output
- **error** // Error

... for example:

$S' \to E$
$E \to E + T^{\,0} \mid T^{\,1}$
$T \to T * F^{\,0} \mid F^{\,1}$
$F \to (\,E\,)^{\,0} \mid \text{int}^{\,1}$

action	$	int	(	)	+	*
q_1	$S', 0$				s	
q_2	$E, 1$				$E, 1$	s
q_2'				$E, 1$	$E, 1$	s
q_3	$T, 1$				$T, 1$	$T, 1$
q_3'				$T, 1$	$T, 1$	$T, 1$
q_4	$F, 1$				$F, 1$	$F, 1$
q_4'				$F, 1$	$F, 1$	$F, 1$
q_9	$E, 0$				$E, 0$	s
q_9'				$E, 0$	$E, 0$	s
q_{10}	$T, 0$				$T, 0$	$T, 0$
q_{10}'				$T, 0$	$T, 0$	$T, 0$
q_{11}	$F, 0$				$F, 0$	$F, 0$
q_{11}'				$F, 0$	$F, 0$	$F, 0$

The Canonical LR(1)-Automaton

In general: We identify two conflicts:

Reduce-Reduce-Conflict:
$[A \to \gamma\,\bullet, x], \; [A' \to \gamma'\,\bullet, x] \in q$ with $A \neq A' \vee \gamma \neq \gamma'$

Shift-Reduce-Conflict:
$[A \to \gamma\,\bullet, x], \; [A' \to \alpha\,\bullet\,a\,\beta, y] \in q$
with $a \in T$ und $x \in \{a\}$.

for a state $q \in Q$.

Such states are now called $LR(1)$-unsuited

The Canonical LR(1)-Automaton

In general: We identify two conflicts:

Reduce-Reduce-Conflict:
$[A \to \gamma\,\bullet, x], \; [A' \to \gamma'\,\bullet, x] \in q$ with $A \neq A' \vee \gamma \neq \gamma'$

Shift-Reduce-Conflict:
$[A \to \gamma\,\bullet, x], \; [A' \to \alpha\,\bullet\,a\,\beta, y] \in q$
with $a \in T$ und $x \in \{a\} \odot_k \text{First}_k(\beta) \odot_k \{y\}$.

for a state $q \in Q$.

Such states are now called $LR(k)$-unsuited

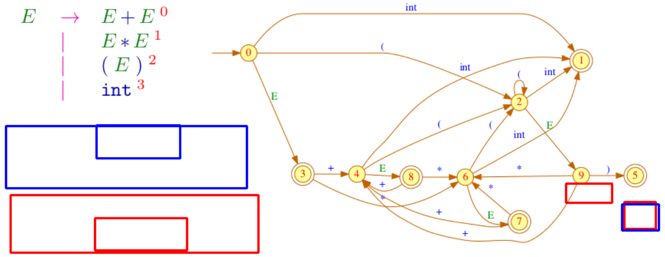
Theorem:
A reduced contextfree grammar G is called $LR(k)$ iff the canonical $LR(k)$-automaton $LR(G, k)$ has no $LR(k)$-unsuited states.

Precedences

Many parser generators give the chance to fix Shift-/Reduce-Conflicts by patching the action table either by hand or with *token precedences*.

... for example:

$$
\begin{array}{lll}
S' & \to & E^{\,0} \\
E & \to & E+E^{\,0} \\
 & | & E*E^{\,1} \\
 & | & (\,E\,)^{\,2} \\
 & | & \texttt{int}^{\,3}
\end{array}
$$



What if precedences are not enough?

Example (very simplified lambda expressions):

$$
\begin{array}{lll}
E & \to & (\,E\,)^{0} \mid \text{ident}^{1} \mid L^{2} \\
L & \to & \langle\text{args}\rangle \Rightarrow E^{0} \\
\langle\text{args}\rangle & \to & (\,\langle\text{idlist}\rangle\,)^{0} \mid \text{ident}^{1} \\
\langle\text{idlist}\rangle & \to & \langle\text{idlist}\rangle\ \text{ident}^{0} \mid \text{ident}^{1}
\end{array}
$$

What if precedences are not enough?

In practice, $LR(k)$-parser generators working with the lookahead sets of sizes larger then $k = 1$ are not common, since computing lookahead sets with $k > 1$ blows up exponentially. However,

1. there exist several practical $LR(k)$ grammars of $k > 1$, e.g. Java 1.6+ ($LR(2)$), ANSI C, etc.
2. often, more lookahead is only exhausted locally
3. should we really give up, whenever we are confronted with a Shift-/Reduce-Conflict?

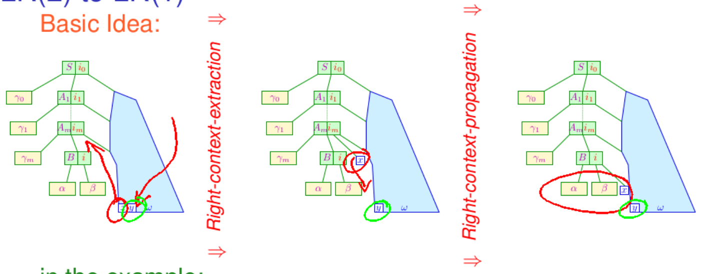
LR(2) to LR(1)

... Example:

$$
\begin{array}{lll}
S & \to & A\,b\,b^{0} \mid B\,b\,c^{1} \\
A & \to & a\,A^{0} \mid a^{1} \\
B & \to & a\,B^{0} \mid a^{1}
\end{array}
$$

LR(2) to LR(1)

Basic Idea:



in the example:

Right-context is already extracted, so we only perform *Right-context-propagation*:

$$S \rightarrow A\,b\,b^0 \mid B\,b\,c^1$$
$$A \rightarrow a\,A^0 \mid a^1$$
$$B \rightarrow a\,B^0 \mid a^1$$

⇒

LR(2) to LR(1)

... Example:

$$S \rightarrow A\,b\,b^0 \mid B\,b\,c^1$$
$$A \rightarrow a\,A^0 \mid a^1$$
$$B \rightarrow a\,B^0 \mid a^1$$

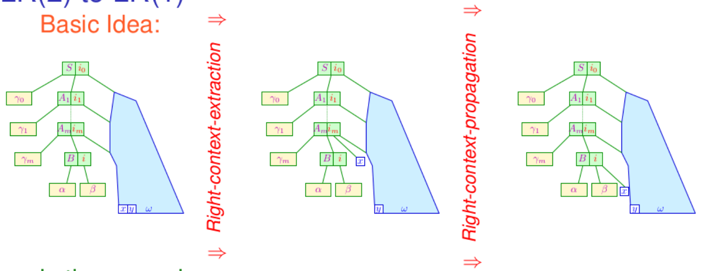
S rightmost-derives one of these forms:

$$a^n\underline{a}bb\;,\;a^n\underline{a}bc\;,\;a^n\underline{a\,A}bb\;,\;a^n\underline{a\,B}bc,\;\underline{A}bb\;,\;\underline{B}bc \quad\Rightarrow\quad LR(2)$$

in $LR(1)$, you will have Reduce-/Reduce-Conflicts between the productions $A, 1$ and $B, 1$ as well as $A, 0$ and $B, 0$ under lookahead b

LR(2) to LR(1)

Basic Idea:



in the example:

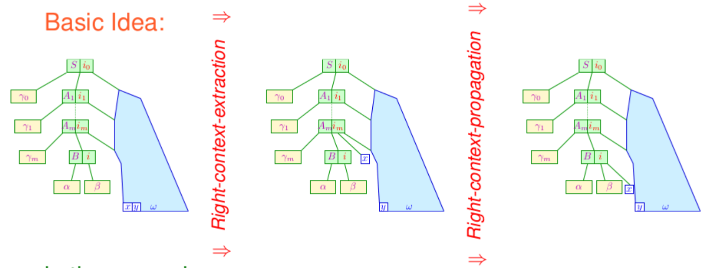
Right-context is already extracted, so we only perform *Right-context-propagation*:

$$S \rightarrow A\,b\,b^0 \mid B\,b\,c^1$$
$$A \rightarrow a\,A^0 \mid a^1$$
$$B \rightarrow a\,B^0 \mid a^1$$

⇒

LR(2) to LR(1)

Basic Idea:



in the example:

Right-context is already extracted, so we only perform *Right-context-propagation*:

$$S \rightarrow A\,b\,b^0 \mid B\,b\,c^1$$
$$A \rightarrow a\,A^0 \mid a^1$$
$$B \rightarrow a\,B^0 \mid a^1$$

⇒

$$S \rightarrow \langle A\,b\rangle\,b^0 \mid \langle B\,b\rangle\,c^1$$
$$\langle A\,b\rangle \rightarrow a\,\langle A\,b\rangle^0 \mid a\,b^1$$
$$\langle B\,b\rangle \rightarrow a\,\langle B\,b\rangle^0 \mid a\,b^1$$

unreachable

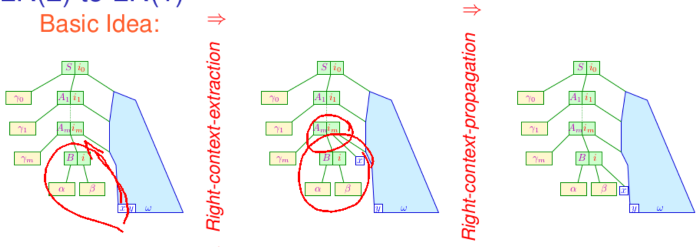
LR(2) to LR(1)

Example cont'd:

$$
\begin{array}{lcl}
S & \to & A'\,b^0 \mid B'\,c^1 \\
A' & \to & a\,A'^0 \mid a\,b^1 \\
B' & \to & a\,B'^0 \mid a\,b^1
\end{array}
$$

LR(2) to LR(1)

Basic Idea:



in the example:

Right-context is already extracted, so we only perform *Right-context-propagation*:

$$
\begin{array}{lcl}
S & \to & A\,b\,b^0 \mid B\,b\,c^1 \\
A & \to & a\,A^0 \mid a^1 \\
B & \to & a\,B^0 \mid a^1
\end{array}
\quad \Rightarrow \quad
\begin{array}{lcl}
S & \to & \langle A\,b\rangle\,b^0 \mid \langle B\,b\rangle\,c^1 \\
\langle A\,b\rangle & \to & a\,\langle A\,b\rangle^0 \mid a\,b^1 \\
\langle B\,b\rangle & \to & a\,\langle B\,b\rangle^0 \mid a\,b^1 \\
A & \to & a\,A^0 \mid a^1 \\
B & \to & a\,B^0 \mid a^1
\end{array}
$$

LR(2) to LR(1)

Example cont'd:

$$
\begin{array}{lcl}
S & \to & A'\,b^0 \mid B'\,c^1 \\
A' & \to & a\,A'^0 \mid a\,b^1 \\
B' & \to & a\,B'^0 \mid a\,b^1
\end{array}
$$

LR(2) to LR(1)

Example 2 cont'd:

$[S \to a]$'s right context is now terminal a ⇒ perform *Right-context-propagation*

LR(2) to LR(1)

Example 2 finished:

With fresh nonterminals we get the final grammar

$$S \to b\,S\,S^0 \mid a^1 \mid a\,a\,c^2 \quad \Rightarrow \quad \begin{array}{lll} S & \to & b\,C\,A,^0 \mid b\,S\,b\,B,^1 \mid a^2 \mid a\,a\,c^3 \\ A & \to & \epsilon^0 \mid a\,c^1 \\ B & \to & C\,A^0 \mid S\,b\,B^1 \\ C & \to & b\,C\,D^0 \mid b\,S\,b\,E^1 \mid a\,a^2 \mid a\,a\,c\,a^3 \\ D & \to & a^0 \mid a\,c\,a^1 \\ E & \to & C\,D^0 \mid S\,b\,E^1 \end{array}$$

LR(2) to LR(1)

Example 2 finished:

With fresh nonterminals we get the final grammar

$$S \to b\,S\,S^0 \mid a^1 \mid a\,a\,c^2 \quad \Rightarrow \quad \begin{array}{lll} S & \to & b\,C\,A,^0 \mid b\,S\,b\,B,^1 \mid a^2 \mid a\,a\,c^3 \\ A & \to & \epsilon^0 \mid a\,c^1 \\ B & \to & C\,A^0 \mid S\,b\,B^1 \\ C & \to & b\,C\,D^0 \mid b\,S\,b\,E^1 \mid a\,a^2 \mid a\,a\,c\,a^3 \\ D & \to & a^0 \mid a\,c\,a^1 \\ E & \to & C\,D^0 \mid S\,b\,E^1 \end{array}$$

LR(2) to LR(1)

Algorithm:

For a Rule $A \to \alpha$, which is *reduce-conflicting* under terminal x

- $B \to \beta\,A$ is also considered *reduce-conflicting* under terminal x
- $B \to \beta\,A\,C\,\gamma$ is transformed by *right-context-extraction* on C:

$$B \to \beta\,A\,C\,\gamma \quad \Rightarrow \quad B \to \beta\,A\,x\,\langle x/C\rangle\,\gamma \;\Big|_{y \in \mathsf{First}_1(C) \setminus x}\; \beta\,A\,y\,\langle y/C\rangle\,\gamma$$

  if $\epsilon \in \mathsf{First}_1(C)$ then consider $B \to \beta\,A\,\gamma$ for r.-c.-extraction

- $B \to \beta\,A\,x\,\gamma$ is transformed by *right-context-propagation* on A:

$$B \to \beta\,A\,x\,\gamma \quad \Rightarrow \quad B \to \beta\,\langle Ax\rangle\,\gamma$$

- The appropriate rules, created from introducing $\langle Ax\rangle \to \delta$ and $\langle x/B\rangle \to \eta$ are added to the grammar