Script generated by TTT

Title: Petter: Compilerbau (06.06.2016)

Date: Mon Jun 06 14:29:05 CEST 2016

Duration: 85:35 min

Pages: 47

LR(k)-Grammars

Idea: Consider k-lookahead in conflict situations.

Definition:

The reduced contextfree grammar G is called $LR(k)$-grammar, if for $\text{First}_{\boxed{k}}(w) = \text{First}_{\boxed{k}}(x)$ with:

$$\left.\begin{array}{llllll} S & \rightarrow^*_R & \boxed{\alpha\, A}\, w & \rightarrow & \boxed{\alpha\,\beta}\,\boxed{w} \\ S & \rightarrow^*_R & \boxed{\alpha'\, A'}\, w' & \rightarrow & \boxed{\alpha\,\beta}\,\boxed{x} \end{array}\right\} \text{ follows: } \alpha = \alpha' \wedge A = A' \wedge w' = x$$

LR(k)-Grammars

Idea: Consider k-lookahead in conflict situations.

Definition:

The reduced contextfree grammar G is called $LR(k)$-grammar, if for $\text{First}_k(w) = \text{First}_k(x)$ with:

$$\left.\begin{array}{llllll} S & \rightarrow^*_R & \alpha\, A\, w & \rightarrow & \alpha\,\beta\, w \\ S & \rightarrow^*_R & \alpha'\, A'\, w' & \rightarrow & \alpha\,\beta\, x \end{array}\right\} \text{ follows: } \alpha = \alpha' \wedge A = A' \wedge w' = x$$

$X \to \beta$

Strategy for testing Grammars for $LR(k)$-property

1. Focus iteratively on all rightmost derivations $S \rightarrow^*_R \alpha\, X\, w \rightarrow \boxed{\alpha\,\beta\, w}$
2. Identify handle $\boxed{\alpha\,\beta}$ in s. forms $\alpha\,\beta\, w$ ($w \in T^*$, $\alpha, \beta \in (N \cup T)^*$)
3. Determine minimal k, such that $\text{First}_k(w)$ associates β with a unique $X \to \beta$ for non-prefixing $\alpha\,\beta$s

LR(k)-Grammars

for example:

(1) $S \to A \mid B \quad A \to a\, A\, b \mid 0 \quad B \to a\, B\, b\, b \mid 1$

$A \quad B \quad aAb \quad aBbb$

$a\,aAbb \quad a\,aBbbbb$

$a^k\,aAb\,b^k \quad a^k\,aBb^2\,b^{2k}$

$a^k 0\, b^k \quad a^k 1\, b^{2k}$

LR(k)-Grammars

for example:

(1) $S \to A \mid B \qquad A \to a\,A\,b \mid 0 \qquad B \to a\,B\,b\,b \mid 1$

... is not $LL(k)$ for any k :

Let $S \to_R^* \alpha\,X\,w \to \alpha\,\beta\,w$. Then $\alpha\,\underline{\beta}$ is of one of these forms:

$$\underline{A}\,,\ \underline{B}\,,\ a^n\,\underline{a\,A\,b}\,,\ a^n\,\underline{a\,B\,b\,b}\,,\ a^n\,\underline{0}\,,\ a^n\,\underline{1} \qquad (n \geq 0)$$

LR(k)-Grammars

for example:

(1) $S \to A \mid B \qquad A \to a\,A\,b \mid 0 \qquad B \to a\,B\,b\,b \mid 1$

... is not $LL(k)$ for any k — but $LR(0)$:

Let $S \to_R^* \alpha\,X\,w \to \alpha\,\beta\,w$. Then $\alpha\,\underline{\beta}$ is of one of these forms:

$$\underline{A}\,,\ \underline{B}\,,\ a^n\,\underline{a\,A\,b}\,,\ a^n\,\underline{a\,B\,b\,b}\,,\ a^n\,\underline{0}\,,\ a^n\,\underline{1} \qquad (n \geq 0)$$

LR(k)-Grammars

for example:

(1) $S \to A \mid B \qquad A \to a\,A\,b \mid 0 \qquad B \to a\,B\,b\,b \mid 1$

... is not $LL(k)$ for any k — but $LR(0)$:

Let $S \to_R^* \alpha\,X\,w \to \alpha\,\beta\,w$. Then $\alpha\,\underline{\beta}$ is of one of these forms:

$$\underline{A}\,,\ \underline{B}\,,\ a^n\,\underline{a\,A\,b}\,,\ a^n\,\underline{a\,B\,b\,b}\,,\ a^n\,\underline{0}\,,\ a^n\,\underline{1} \qquad (n \geq 0)$$

(2) $S \to a\,A\,c \qquad A \to A\,b\,b \mid b$

$\underline{a\,A\,c}$

$a\,\underline{A\,b\,b}^2\,c$

$a\,\underline{A\,b\,b}^2\,b\,b^{2k}\,c$

$a\,\underline{b}\,b^{2k}\,c$

LR(k)-Grammars

for example:

(1) $S \to A \mid B \qquad A \to a\,A\,b \mid 0 \qquad B \to a\,B\,b\,b \mid 1$

... is not $LL(k)$ for any k — but $LR(0)$:

Let $S \to_R^* \alpha\,X\,w \to \alpha\,\beta\,w$. Then $\alpha\,\underline{\beta}$ is of one of these forms:

$$\underline{A}\,,\ \underline{B}\,,\ a^n\,\underline{a\,A\,b}\,,\ a^n\,\underline{a\,B\,b\,b}\,,\ a^n\,\underline{0}\,,\ a^n\,\underline{1} \qquad (n \geq 0)$$

(2) $S \to a\,A\,c \qquad A \to A\,b\,b \mid b$

... is also not $LL(k)$ for any k :

Let $S \to_R^* \alpha\,X\,w \to \alpha\,\beta\,w$. Then $\alpha\,\underline{\beta}$ is of one of these forms:

$$a\,\underline{b}\,,\ a\,\underline{A\,b\,b}\,,\ \underline{a\,A\,c}$$

LR(k)-Grammars

1

$a\;b\;c$

for example:

(3) $\quad S \to a\,A\,c \qquad A \to b\,b\,A \;|\; b$

$a\,A\,c \qquad a\;b\;b\;A\;c$

$a\;b^{2k}\;b^{2}\;A\;c$

$a\;\cancel{b^{2k}}\;b\;c$

LR(k)-Grammars

for example:

(3) $\quad S \to a\,A\,c \qquad A \to b\,b\,A \;|\; b$

Let $\; S \to_R^* \alpha\,X\,w \to \alpha\,\beta\,w \;$ with $\; \{y\} = \text{First}_k(w) \;$ then $\alpha\,\underline{\beta}\,y$ is of one of these forms:

$$a\,b^{2n}\,\underline{b\,c}\;,\; a\,b^{2n}\,\underline{b\,b\,A}\,c\;,\; \underline{a\,A\,c}$$

LR(k)-Grammars

for example:

(3) $\quad S \to a\,A\,c \qquad A \to b\,b\,A \;|\; b \qquad$ ... is not $LR(0)$, but $LR(1)$:

Let $\; S \to_R^* \alpha\,X\,w \to \alpha\,\beta\,w \;$ with $\; \{y\} = \text{First}_k(w) \;$ then $\alpha\,\underline{\beta}\,y$ is of one of these forms:

$$a\,b^{2n}\,\underline{b}\,c\;,\; a\,b^{2n}\,\underline{b\,b\,A}\,c\;,\; \underline{a\,A\,c}$$

LR(k)-Grammars

for example:

(3) $\quad S \to a\,A\,c \qquad A \to b\,b\,A \;|\; b \qquad$ ... is not $LR(0)$, but $LR(1)$:

Let $\; S \to_R^* \alpha\,X\,w \to \alpha\,\beta\,w \;$ with $\; \{y\} = \text{First}_k(w) \;$ then $\alpha\,\underline{\beta}\,y$ is of one of these forms:

$$a\,b^{2n}\,\underline{b}\,c\;,\; a\,b^{2n}\,\underline{b\,b\,A}\,c\;,\; \underline{a\,A\,c}$$

(4) $\quad S \to a\,A\,c \qquad A \to b\,A\,b \;|\; b$

LR(k)-Grammars

for example:

(3) $S \to a\,A\,c \qquad A \to b\,b\,A \mid b$ ... is not $LR(0)$, but $LR(1)$:

Let $S \to_R^* \alpha\,X\,w \to \alpha\,\beta\,w$ with $\{y\} = \text{First}_k(w)$ then $\alpha\,\underline{\beta}\,y$ is of one of these forms:

$$a\,b^{2n}\,\underline{b\,c}\;,\;\; a\,b^{2n}\,\underline{b\,b\,A}\,c\;,\;\; \underline{a\,A\,c}$$

(4) $S \to a\,A\,c \qquad A \to b\,A\,b \mid b$

Consider the rightmost derivations:

$$S \to_R^* \boxed{a\,b^n\,A\,b^n\,c} \to \boxed{a\,b^n\,\underline{b}}\,b^n\,c$$

LR(k)-Grammars

for example:

(3) $S \to a\,A\,c \qquad A \to b\,b\,A \mid b$ ... is not $LR(0)$, but $LR(1)$:

Let $S \to_R^* \alpha\,X\,w \to \alpha\,\beta\,w$ with $\{y\} = \text{First}_k(w)$ then $\alpha\,\underline{\beta}\,y$ is of one of these forms:

$$a\,b^{2n}\,\underline{b\,c}\;,\;\; a\,b^{2n}\,\underline{b\,b\,A}\,c\;,\;\; \underline{a\,A\,c}$$

(4) $S \to a\,A\,c \qquad A \to b\,A\,b \mid b$ ... is not $LR(k)$ for any $k \geq 0$:

Consider the rightmost derivations:

$$S \to_R^* a\,b^n\,A\,b^n\,c \to a\,b^n\,\underline{b}\,b^n\,c$$

LR(1)-Parsing

Idea: Let's equip items with 1-lookahead

Definition LR(1)-Item

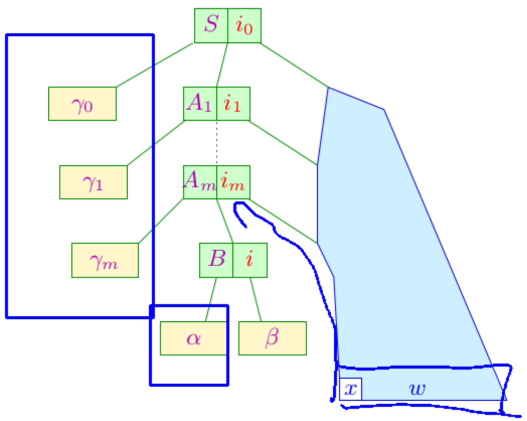
An $LR(1)$-item is a pair $\boxed{B \to \alpha \bullet \beta}, \boxed{x}$ with

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

Admissible LR(1)-Items

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

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



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

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

This automaton works like $c(G)$ — but additionally manages a 1-prefix from Follow_1 of the left-hand sides.

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

This automaton works like $c(G)$ — but additionally manages a 1-prefix from Follow_1 of the left-hand sides.

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

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 ...

But again, it can be constructed directly from the grammar; analoguously to $LR(0)$, we need the ϵ-closure δ_ϵ^* as a helper function:

$$\delta_\epsilon^*(q) = q \cup \{[C \to \bullet\gamma, x] \mid \exists\, [A \to \alpha \bullet B\,\beta', x'] \in q, \beta \in (N \cup T)^* : B \to^* C\,\beta \wedge x \in \text{First}_1(\beta\,\beta') \odot_1 \{x'\}\}$$

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

This automaton works like $c(G)$ — but additionally manages a 1-prefix from Follow_1 of the left-hand sides.

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 ...

But again, it can be constructed directly from the grammar; analoguously to $LR(0)$, we need the ϵ-closure δ_ϵ^* as a helper function:

$$\delta_\epsilon^*(q) = q \cup \{[\boxed{C \to \bullet\gamma}, \boxed{x}] \mid \exists\, [A \to \alpha \bullet \boxed{B}\,\beta', x'] \in q, \beta \in (N \cup T)^* : B \to^* C\,\beta \wedge x \in \boxed{\text{First}_1(\beta\,\beta') \odot_1 \{x'\}}\}$$

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 ...

But again, it can be constructed directly from the grammar; analoguously to $LR(0)$, we need the ϵ-closure δ^*_ϵ as a helper function:

$$\delta^*_\epsilon(q) = q \cup \{[C \to \bullet\gamma, x] \mid \exists\, [A \to \alpha \bullet B\,\beta', x'] \in q, \beta \in (N \cup T)^* : B \to^* C\,\beta \;\wedge\; x \in \mathsf{First}_1(\beta\,\beta') \odot_1 \{x'\}\}$$

Then, we define:

States: Sets of $LR(1)$-items;

Start state: $\delta^*_\epsilon\,\{[S' \to \bullet\, S, \epsilon]\}$

Final states: $\{q \mid \exists\, A \to \alpha \in P : [A \to \alpha\,\bullet, x] \in q\}$

Transitions: $\delta(q, X) = \delta^*_\epsilon\,\{[A \to \alpha\, X \bullet \beta, \boxed{x}] \mid [A \to \alpha \bullet X\,\beta, \boxed{x}] \in q\}$

The Canonical LR(1)-Automaton

For example:

$$\begin{array}{lcll} E & \to & E + T & \mid\; T \\ T & \to & T * F & \mid\; F \\ F & \to & (\,E\,) & \mid\; \mathsf{int} \end{array}$$

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

$$\begin{array}{lclcl}
q_0 & = & & & \{[S' \to \bullet\, E \quad], [E \to \bullet\, E + T \quad], [E \to \bullet\, T \quad], [T \to \bullet\, T * F \quad], [T \to \bullet\, F \quad], [F \to \bullet\, (\,E\,) \quad], [F \to \bullet\, \mathsf{int} \quad]\} \\
q_1 & = & \delta(q_0, E) & = & \{[S' \to E\,\bullet \quad], [E \to E\,\bullet + T \quad]\} \\
q_2 & = & \delta(q_0, T) & = & \{[E \to T\,\bullet \quad], [T \to T\,\bullet * F \quad]\} \\
q_3 & = & \delta(q_0, F) & = & \{[T \to F\,\bullet \quad]\} \\
q_4 & = & \delta(q_0, \mathsf{int}) & & \{[F \to \mathsf{int}\,\bullet \quad]\} \\
q_5 & = & \delta(q_0, () & = & \{[F \to (\,\bullet\, E\,) \quad], [E \to \bullet\, E + T \quad], [E \to \bullet\, T \quad], [T \to \bullet\, T * F \quad], [T \to \bullet\, F \quad], [F \to \bullet\, (\,E\,) \quad], [F \to \bullet\, \mathsf{int} \quad]\}
\end{array}$$

The Canonical LR(1)-Automaton

For example:

$$\begin{array}{lcll} E & \to & E + T & \mid\; T \\ T & \to & T * F & \mid\; F \\ F & \to & (\,E\,) & \mid\; \mathsf{int} \end{array}$$

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

$$\begin{array}{lclcl}
q_0 & = & & & \{[S' \to \bullet\, E, \{\epsilon\}], [E \to \bullet\, E + T, \{\epsilon, +\}], [E \to \bullet\, T, \{\epsilon, +\}], [T \to \bullet\, T * F, \{\epsilon, +, *\}], [T \to \bullet\, F, \{\epsilon, +, *\}], [F \to \bullet\, (\,E\,), \{\epsilon, +, *\}], [F \to \bullet\, \mathsf{int}, \{\epsilon, +, *\}]\} \\
q_1 & = & \delta(q_0, E) & = & \{[S' \to E\,\bullet, \{\epsilon\}], [E \to E\,\bullet + T, \{\epsilon, +\}]\} \\
q_2 & = & \delta(q_0, T) & = & \{[E \to T\,\bullet, \{\epsilon, +\}], [T \to T\,\bullet * F, \{\epsilon, +, *\}]\} \\
q_3 & = & \delta(q_0, F) & = & \{[T \to F\,\bullet, \{\epsilon, +, *\}]\} \\
q_4 & = & \delta(q_0, \mathsf{int}) & & \{[F \to \mathsf{int}\,\bullet, \{\epsilon, +, *\}]\} \\
q_5 & = & \delta(q_0, () & = & \{[F \to (\,\bullet\, E\,) \quad], [E \to \bullet\, E + T \quad], [E \to \bullet\, T \quad], [T \to \bullet\, T * F \quad], [T \to \bullet\, F \quad], [F \to \bullet\, (\,E\,) \quad], [F \to \bullet\, \mathsf{int} \quad]\}
\end{array}$$

$\epsilon + *$ $\epsilon + *$ $\epsilon + *$

The Canonical LR(1)-Automaton

For example:

$$\begin{array}{lcll} E & \to & E + T & \mid\; T \\ T & \to & T * F & \mid\; F \\ F & \to & (\,E\,) & \mid\; \mathsf{int} \end{array}$$

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

$$\begin{array}{lclcl}
q_0 & = & & & \{[S' \to \bullet\, E, \{\epsilon\}], [E \to \bullet\, E + T, \{\epsilon, +\}], [E \to \bullet\, T, \{\epsilon, +\}], [T \to \bullet\, T * F, \{\epsilon, +, *\}], [T \to \bullet\, F, \{\epsilon, +, *\}], [F \to \bullet\, (\,E\,), \{\epsilon, +, *\}], [F \to \bullet\, \mathsf{int}, \{\epsilon, +, *\}]\} \\
q_1 & = & \delta(q_0, E) & = & \{[S' \to E\,\bullet, \{\epsilon\}], [E \to E\,\bullet + T, \{\epsilon, +\}]\} \\
q_2 & = & \delta(q_0, T) & = & \{[E \to T\,\bullet, \{\epsilon, +\}], [T \to T\,\bullet * F, \{\epsilon, +, *\}]\} \\
q_3 & = & \delta(q_0, F) & = & \{[T \to F\,\bullet, \{\epsilon, +, *\}]\} \\
q_4 & = & \delta(q_0, \mathsf{int}) & & \{[F \to \mathsf{int}\,\bullet, \{\epsilon, +, *\}]\} \\
q_5 & = & \delta(q_0, () & = & \{[F \to (\,\bullet\, E\,\boxed{)}, \{\epsilon, +, *\}], [E \to \bullet\, E + T, \{\,),+\}], [E \to \bullet\, T, \{\,),+\}], [T \to \bullet\, T * F, \{\,),+,*\}], [T \to \bullet\, F, \{\,),+,*\}], [F \to \bullet\, (\,E\,), \{\,),+,*\}], [F \to \bullet\, \mathsf{int}, \{\,),+,*\}]\}
\end{array}$$

The Canonical LR(1)-Automaton

For example:

$$\begin{array}{lcll} E & \to & E+T & \mid\ T \\ T & \to & T*F & \mid\ F \\ F & \to & (\,E\,) & \mid\ \text{int} \end{array}$$

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

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

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

The Canonical LR(1)-Automaton

For example:

$$\begin{array}{lcll} E & \to & E+T & \mid\ T \\ T & \to & T*F & \mid\ F \\ F & \to & (\,E\,) & \mid\ \text{int} \end{array}$$

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

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

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

The Canonical LR(1)-Automaton

For example:

$$\begin{array}{lcll} E & \to & E+T & \mid\ T \\ T & \to & T*F & \mid\ F \\ F & \to & (\,E\,) & \mid\ \text{int} \end{array}$$

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

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

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

144 / 291

The Canonical LR(1)-Automaton

For example:

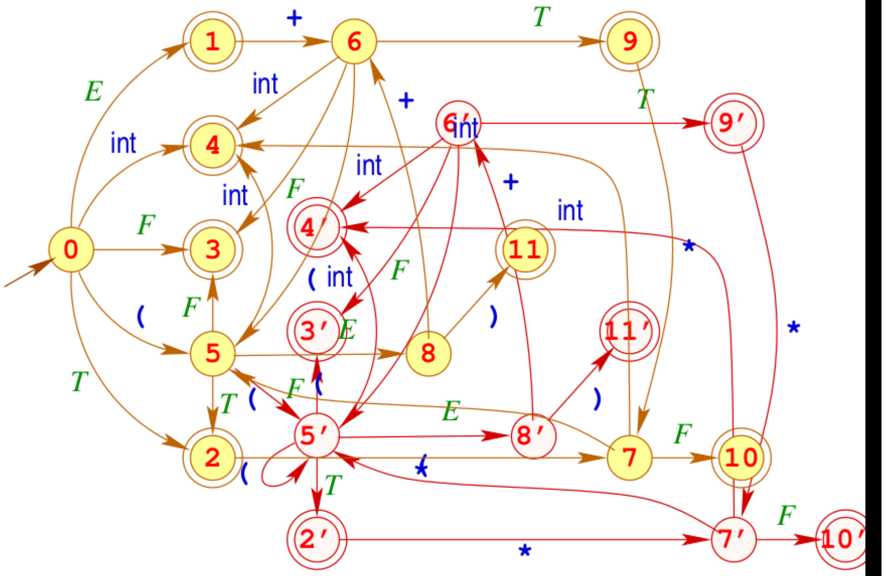
$$\begin{array}{lcll} E & \to & E+T & \mid\ T \\ T & \to & T*F & \mid\ F \\ F & \to & (\,E\,) & \mid\ \text{int} \end{array}$$

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

$$\begin{array}{lclcl}
q'_2 & = & \delta(q'_5, T) & = & \{[E \to T\bullet,\ \{\,),+\}], \\
 & & & & \ [T \to T\, \bullet * F,\ \{\,),+,*\}]\} \\
q'_3 & = & \delta(q'_5, F) & = & \{[F \to F\bullet,\ \{\,),+,*\}]\} \\
q'_4 & = & \delta(q'_5, \text{int}) & = & \{[F \to \text{int}\bullet,\ \{\,),+,*\}]\} \\
q'_6 & = & \delta(q_8, +) & = & \{[E \to E + \bullet\, T,\ \{\,),+\}], \\
 & & & & \ [T \to \bullet\, T*F,\ \{\,),+,*\}], \\
 & & & & \ [T \to \bullet\, F,\ \{\,),+,*\}], \\
 & & & & \ [F \to \bullet\, (\,E\,),\ \{\,),+,*\}], \\
 & & & & \ [F \to \bullet\, \text{int},\ \{\,),+,*\}]\}
\end{array}$$

$$\begin{array}{lclcl}
q'_7 & = & \delta(q_9, *) & = & \{[T \to T * \bullet\, F,\ \{\,),+,*\}], \\
 & & & & \ [F \to \bullet\, (\,E\,),\ \{\,),+,*\}], \\
 & & & & \ [F \to \bullet\, \text{int},\ \{\,),+,*\}]\} \\
q'_8 & = & \delta(q'_5, E) & = & \{[F \to (\,E\, \bullet\,),\ \{\,),+,*\}]\} \\
 & & & & \ [E \to E\, \bullet + T,\ \{\,),+\}]\} \\
q'_9 & = & \delta(q'_6, T) & = & \{[E \to E+T\bullet,\ \{\,),+\}], \\
 & & & & \ [T \to T\, \bullet * F,\ \{\,),+,*\}]\} \\
q'_{10} & = & \delta(q'_7, F) & = & \{[T \to T*F\bullet,\ \{\,),+,*\}]\} \\
q'_{11} & = & \delta(q'_8,)\,) & = & \{[F \to (\,E\,)\bullet,\ \{\,),+,*\}]\}
\end{array}$$

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

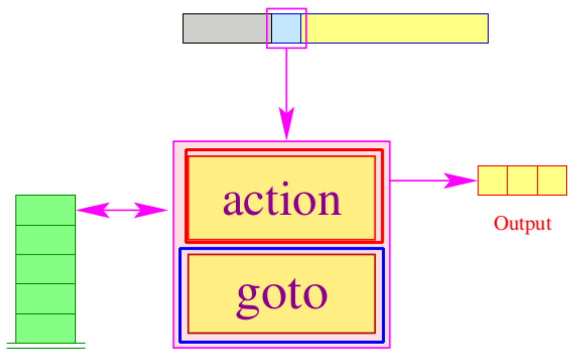
- The conflicts in states q_1, q_2, q_9 are now resolved !
  e.g. we have for:

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

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:

$$E \to E + T^{\,0} \mid T^{\,1}$$
$$T \to T * F^{\,0} \mid F^{\,1}$$
$$F \to (\,E\,)^{\,0} \mid \text{int}^{\,1}$$

action	ϵ	int	(	)	+	*
q_1	$S',0$					s
q_2	$E,1$					s
q_2'				$E,1$		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

Special LR(k)-Subclasses

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.

Special LR(k)-Subclasses

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.

Discussion:

- Our example apparently is $LR(1)$
- In general, the canonical $LR(k)$-automaton has much more states then $LR(G) = LR(G,0)$
- Therefore in practice, subclasses of $LR(k)$-grammars are often considered, which only use $LR(G)$ ...

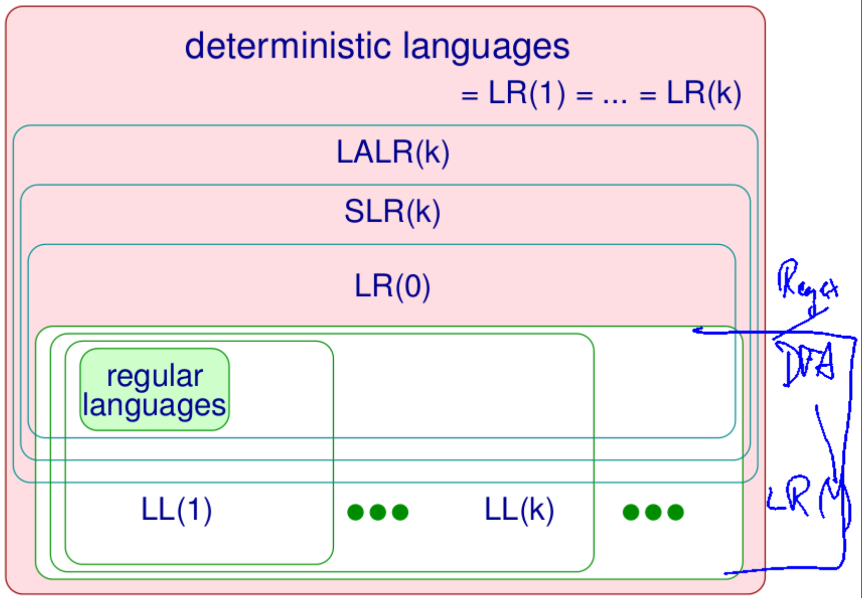
Special LR(k)-Subclasses

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.

Discussion:

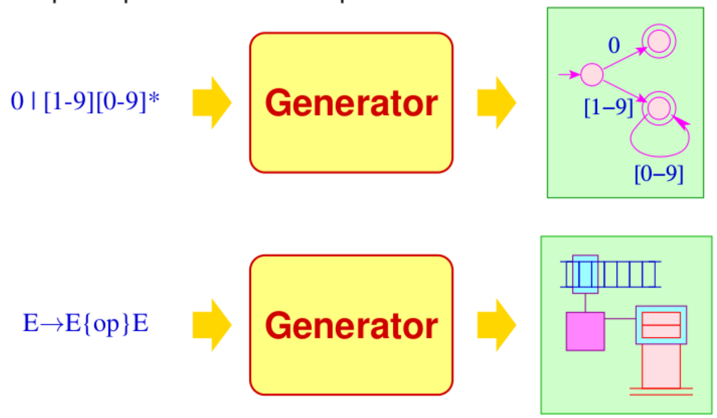
- Our example apparently is $LR(1)$
- In general, the canonical $LR(k)$-automaton has much more states then $LR(G) = LR(G,0)$
- Therefore in practice, subclasses of $LR(k)$-grammars are often considered, which only use $LR(G)$ ...
- For resolving conflicts, the items are assigned special lookahead-sets:
  1. independently on the state itself $\Longrightarrow$ Simple $LR(k)$
  2. dependent on the state itself $\Longrightarrow$ $LALR(k)$

Parsing Methods



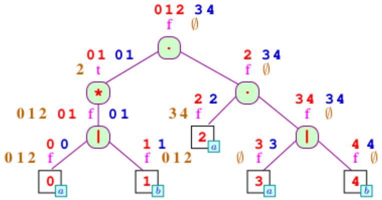
Lexical and Syntactical Analysis:

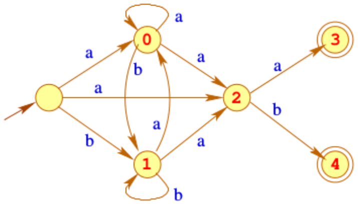
Concept of specification and implementation:



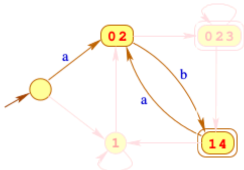
Lexical and Syntactical Analysis:

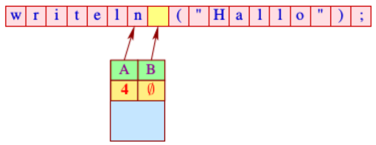
From Regular Expressions to Finite Automata





From Finite Automata to Scanners

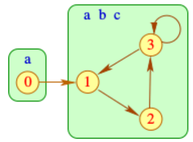


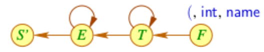


Lexical and Syntactical Analysis:

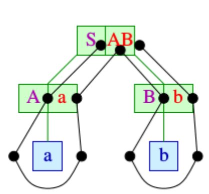
Computation of lookahead sets:

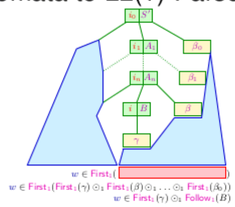
$$\begin{array}{lllllll} F_\epsilon(S') & \supseteq & F_\epsilon(E) & \quad & F_\epsilon(E) & \supseteq & F_\epsilon(E) \\ F_\epsilon(E) & \supseteq & F_\epsilon(T) & & F_\epsilon(T) & \supseteq & F_\epsilon(T) \\ F_\epsilon(T) & \supseteq & F_\epsilon(F) & & F_\epsilon(F) & \supseteq & \{(,\text{name},\text{int}\} \end{array}$$

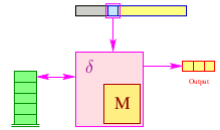




From Item-Pushdown Automata to LL(1)-Parsers:

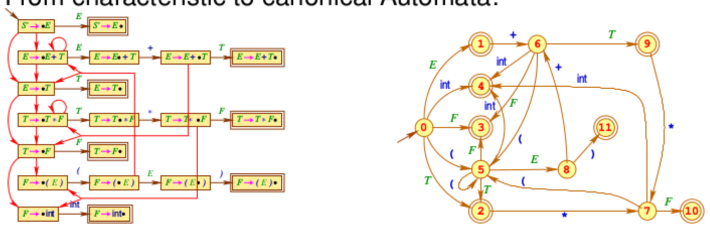




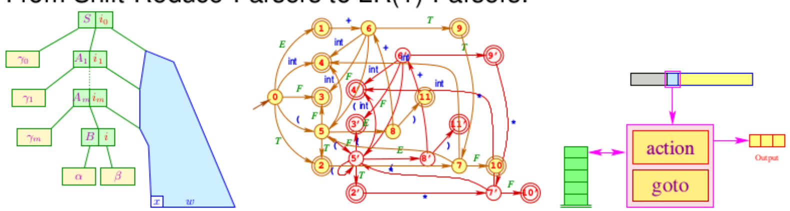


Lexical and Syntactical Analysis:

From characteristic to canonical Automata:



From Shift-Reduce-Parsers to LR(1)-Parsers:



Mini-Projects

1. parse Regular Expressions with PEGs and Recursive Descent Parsing
2. parse Regular Expressions with CUP
3. parse Regular Expressions with ANTLR
4. generate NFA-Transition table from Regex Trees via Thompson Construction
5. generate NFA-Transition table from Regex Trees via Berry-Sethi Construction
6. generate NFA-Transition table from Regex Trees via Antimirov Automaton Construction
7. generate NFA-Transition table from Regex Trees via Follow Automaton Construction
8. implement flex based on Regex -> NFA Module
9. extend simpleC by ellipsis, enums and unions
10. extend simpleC by typecasts and type checking
11. parsing BNF grammars with CUP and computing First/Follow$_1$
12. parsing BNF grammars with ANTLR and computing First/Follow$_k$
13. parsing BNF grammars with JavaCC and computing the canonical-LR(0) Automaton
14. given each symbols First/Follow-Set construct the SLR(1) Automaton
15. computing example traces for reaching S/R and R/R conflicts in the LR(0) Automaton
16. transforming (in-) direct left recursive grammars into (potentially) right recursive grammars
17. semi-deciding k-Ambiguity
18. C4Script (Generating LLVM from a script language)