6.C Deterministic Right Parsers

Let G = (N, T, P, S) be a **context-free** *grammar.*

Def. *[$A \rightarrow \alpha\bullet\beta$] is called as an LR(0) item, if $A \rightarrow \alpha\beta \in P$.*
Let $I_G = \{[A \rightarrow \alpha\bullet\beta] \mid A \rightarrow \alpha\beta \in P\}$ be set of LR(0) items of G.

Def. *An* ***augmented*** *grammar of G = (N, T, P, S) is*

$$G' = (N \cup \{S'\}, T, P \cup \{S' \rightarrow S\}, S') \text{ where } S' \notin N.$$

Fact, *L(G) = L(G')*

Def. *The* ***rule automaton*** *for an augmented cfg G = (N, T, P, S) is an ε-NFA $RA_G = (I_G, N \cup T, \rightarrow_R, [S' \rightarrow \bullet S], \emptyset)$ where*

$([A \rightarrow \alpha\bullet X\gamma], X) \rightarrow_R ([A \rightarrow \alpha X\bullet\gamma], \varepsilon)$ for $X \in N \cup T$, $\alpha, \gamma \in (N \cup T)^*$,

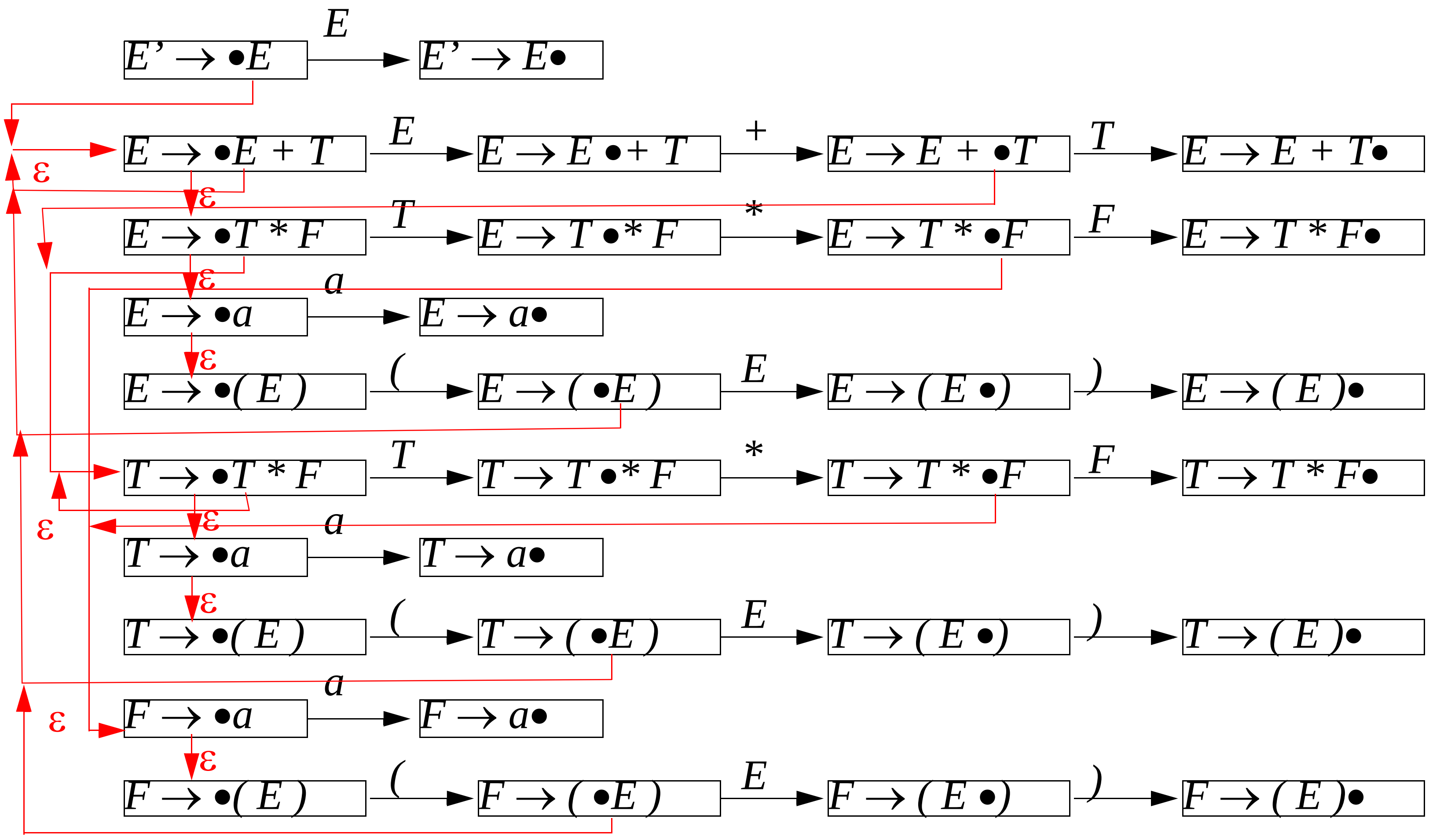
$([A \rightarrow \alpha\bullet B\gamma], \varepsilon) \rightarrow_R ([B \rightarrow \bullet\beta], \varepsilon)$ for $B \in N$, $\alpha, \gamma \in (N \cup T)^*$ $B \rightarrow \beta \in P$.

Example: RA_G: $E' \rightarrow E$

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

$T \rightarrow T * F \mid a \mid (E)$

$F \rightarrow a \mid (E)$



Def. *A DFA of the* ε*-NFA* RA_G *is* $D_G = (2^{I_G}, N \cup T, \rightarrow_D, \varepsilon^*([S' \rightarrow \bullet S]), \varepsilon^*([S' \rightarrow S\bullet])$ *where* $(\varepsilon^*(\{[A \rightarrow \alpha\bullet X\gamma]\}), X) \rightarrow_D (\varepsilon^*(\{[A \rightarrow \alpha X\bullet\gamma]\}), \varepsilon)$

***Alg*. *Collection* of *set of* LR(0) *items*(LR(0) *states*):** $C_0 \subseteq 2^{I_G}$.

$C_0 := \{\varepsilon^*([S' \rightarrow \bullet S])\}$; /* $= [\varepsilon]$ */

repeat for $q \in C_0$ ***do***

  for $[\{[A \rightarrow \alpha\bullet X\gamma]\} \in q$ ***do*** *where* $X \in N \cup T$ /* $q = [\alpha^R\delta^R]$ */

   $p_X := \varepsilon^*(\{[A \rightarrow \alpha X\bullet\gamma]\})$; /* $p_X = [X\alpha^R\delta^R]$ */

   $(X, q) \rightarrow^X (\varepsilon, p_X q)$; /* $(X, [\alpha^R\delta^R]) \rightarrow^X (\varepsilon, [X\alpha^R\delta^R][\alpha^R\delta^R])$ */

   $C_0 := C_0 \cup \{q\}$

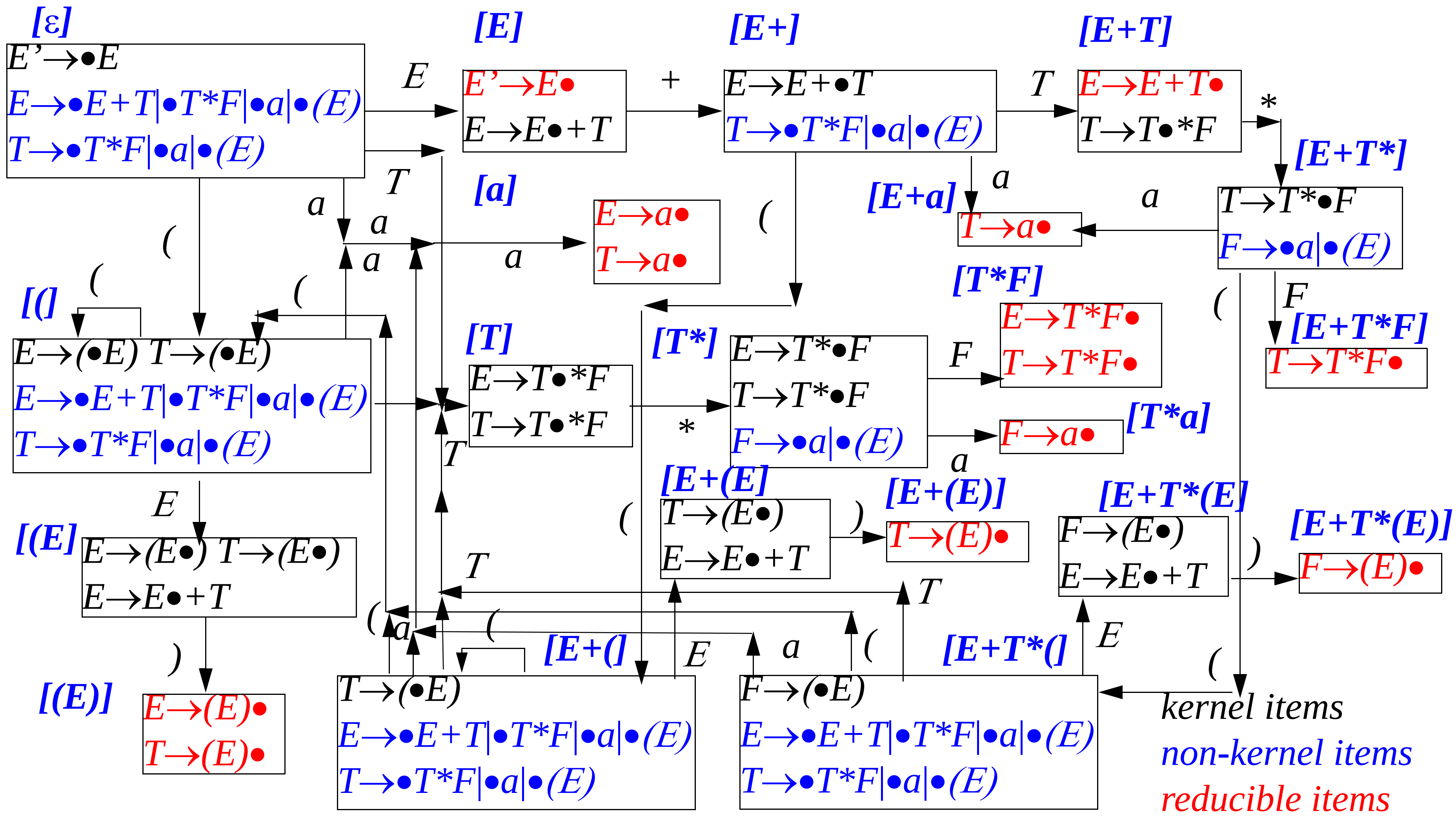
  od od until C_0 *does not change*

Kernel and non kernel items of in LR(0) states $p \in C_0$

  K_P*: Kernel items* $[A \rightarrow \alpha\bullet\beta] \in p$, *if* $(A = S') \vee (\alpha \neq \varepsilon)$.

  $p = \varepsilon^*(K_P) = K_P \cup \varepsilon^+(K_P)$.   *Kernel items* $\cup$ *Non kernel items*

C_0 for G_{Exp}:


[ε]
E'→•E
E→•E+T|•T*F|•a|•(E)
T→•T*F|•a|•(E)
[E]
E'→E•
E→E•+T
[E+]
E→E+•T
T→•T*F|•a|•(E)
[E+T]
E→E+T•
T→T•*F
[E+T*]
T→T*•F
F→•a|•(E)
[a]
E→a•
T→a•
[E+a]
T→a•
[T*F]
E→T*F•
T→T*F•
[E+T*F]
T→T*F•
[(]
E→(•E) T→(•E)
E→•E+T|•T*F|•a|•(E)
T→•T*F|•a|•(E)
[T]
E→T•*F
T→T•*F
[T*]
E→T*•F
T→T*•F
F→•a|•(E)
[T*a]
F→a•
[(E]
E→(E•) T→(E•)
E→E•+T
[E+(E]
T→(E•)
E→E•+T
[E+(E)]
T→(E)•
[E+T*(E]
F→(E•)
E→E•+T
[E+T*(E)]
F→(E)•
[(E)]
E→(E)•
T→(E)•
[E+(]
T→(•E)
E→•E+T|•T*F|•a|•(E)
T→•T*F|•a|•(E)
[E+T*(]
F→(•E)
E→•E+T|•T*F|•a|•(E)
T→•T*F|•a|•(E)
kernel items
non-kernel items
reducible items

*Two **actions** in LR(0) right parser(PDA) with **configuration** $T^* \times C_0^*$.*

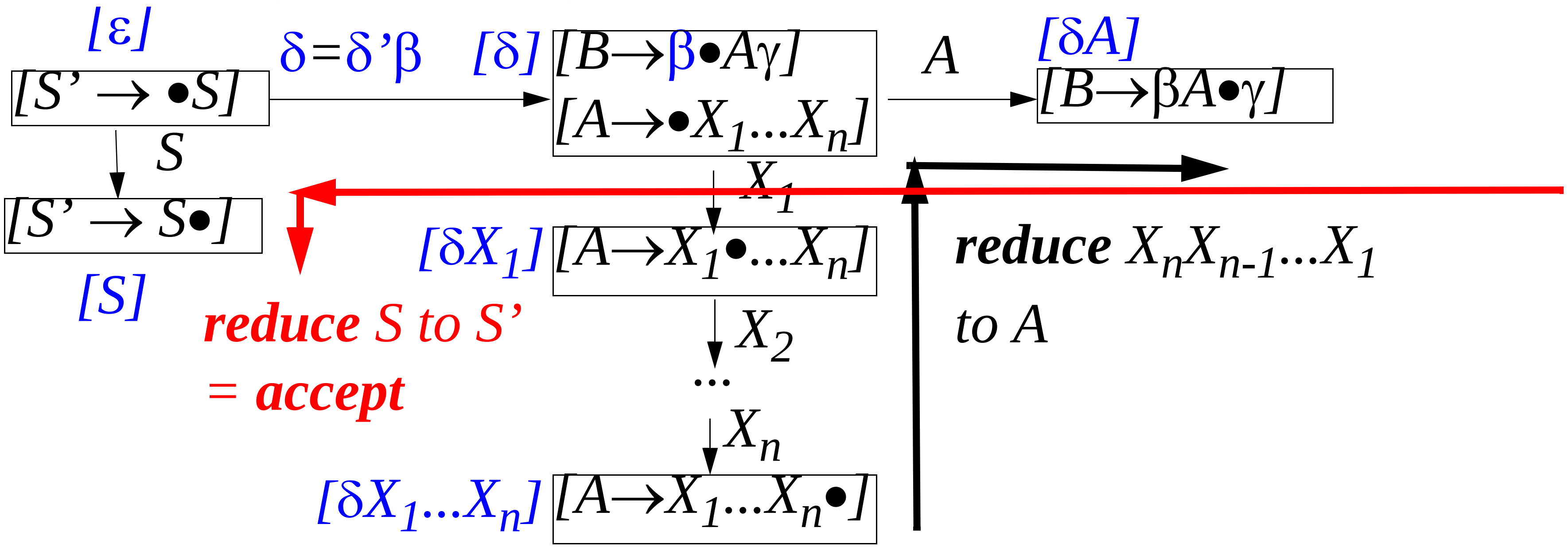
*1) **shift** a:* $a \in T$ $\quad [A \to \alpha \bullet a\beta] \in [\alpha^R\delta^R]$.

$$(a, [\delta^R\alpha^R]) \to (a, [a\alpha^R\delta^R][\alpha^R\delta^R])$$

*2) **reduce** α to A:* $A \to \alpha \in P$ $[A \to \alpha\bullet] \in [\alpha^R\delta^R]$ *where* $\alpha = X_1X_2 \ldots X_n$.

$$(\varepsilon, [X_nX_{n-1}\ldots X_1\delta^R][X_{n-1}\ldots X_1\delta^R] \ldots [X_2X_1\delta^R][X_1\delta^R][\delta^R])$$

$$\to (\varepsilon, [A\delta^R][\delta^R])$$



Let $G = (N, T, P, S)$ be a ***context-free*** *grammar.*

LR(0) parser *for G is* $P_{LR(0)} = (T^* \times C_0{}^*, \rightarrow_{LR(0)}, (x, [\varepsilon]), (\varepsilon, [S][\varepsilon]))$

1) ***shift*** *a:* $a \in T$

If $[A \rightarrow \alpha \bullet a\beta] \in [\alpha^R\delta^R]$, *then*

$(a, [\delta^R\alpha^R]) \rightarrow_{LR(0)}{}^a (a, [a\alpha^R\delta^R][\alpha^R\delta^R])$ ***shift*** *a* $\in T$.

no shift-shift conflict!

2) ***reduce*** α *to A:* $A \rightarrow \alpha \in P$

If $[A \rightarrow \alpha \bullet] \in [\alpha^R\delta^R]$ *where* $\alpha = X_1X_2 \ldots X_n$, *then*

$(\varepsilon, [X_nX_{n-1}\ldots X_1\delta^R][X_{n-1}\ldots X_1\delta^R] \ldots [X_2X_1\delta^R][X_1\delta^R][\delta^R])$

$\rightarrow_{LR(0)}{}^{A\rightarrow\alpha} (\varepsilon, [A\delta^R][\delta^R])$ ***reduce*** $A \rightarrow \alpha \in P$

shift-reduce *and/or* ***reduce-reduce conflict!***

If $P_{LR(0)}$ *is* ***deterministic***, *then G is LR(0) grammar.*

G_{exp} *is not LR(0).*

Right parser *for G is* $P_R = (T^* \times (T \cup N)^*, \rightarrow_R, (x, \varepsilon), (\varepsilon, S))$ *where*

1) ***shift*** *a:* $a \in T$ ***shift*** $a \in T$.

$$(a, \delta^R\alpha^R) \rightarrow_R^a (a, a\alpha^R\delta^R).$$

2) ***reduce*** *α to A:* $A \rightarrow \alpha \in P$ ***reduce*** $A \rightarrow \alpha \in P$.

$$(\varepsilon, \alpha^R\delta^R) \rightarrow_R^{A\rightarrow\alpha} (\varepsilon, A\delta^R).$$

Compare stack C_0^* *in LR(0) parser and* $(T \cup N)^*$ *in right parser!*

$[X\delta^R\alpha^R] \in C_0^*$ *in LR(0) vs* $X\delta^R\alpha^R \in (T \cup N)^*$ *in right parser.*

the equivalent class $[X\delta^R\alpha^R]$ *is the* ***refinement*** *of* $X\delta^R\alpha^R$.

$\iota_{LR(0)} = [\varepsilon]$ $|\iota_{LR(0)}| = 1$ but $\iota_R = \varepsilon$ $|\iota_R| = 0$,

$\phi_{LR(0)} = [S][\varepsilon]$ $|\phi_{LR(0)}| = 2$ but $\phi_R = S$ $|\phi_R| = 1$.

Depth *of the* ***stack*** *in LR parser is one larger than that of the right parser.*

Strong** LR(k) (SLL(k)) **parser

SLR(k) parser $P_{SLR(k)} = (T^* \times C_0^*, \rightarrow_{SLR(k)}, (x, [\varepsilon]), (\varepsilon, [S][\varepsilon]))$ *is*

1) ***shift** a:* $a \in T$

If $[A \rightarrow \alpha \bullet a\beta] \in [\alpha^R\delta^R]$*, then*

$(a, [\delta^R\alpha^R]) \rightarrow_{SLR(k)}^{a} (a, [a\alpha^R\delta^R][\alpha^R\delta^R])$ ***shift** a* $\in T$.

2) ***reduce** α to A:* $A \rightarrow \alpha \in P$

If $[A \rightarrow \alpha \bullet] \in [\alpha^R\delta^R]$ *where* $\alpha = X_1X_2 \ldots X_n$*, then*

$(x, [X_nX_{n-1}\ldots X_1\delta^R][X_{n-1}\ldots X_1\delta^R] \ldots [X_2X_1\delta^R][X_1\delta^R][\delta^R])$

$\rightarrow_{SLR(0)}^{A\rightarrow\alpha} (x, [A\delta^R][\delta^R])$ $x \in Follow(A)$ ***reduce*** $A \rightarrow \alpha \in P$

SLR(k) parser is called Strong LR(k) parser, since

$x \in Follow(A)$.

If $P_{SLR(k)}$ *is **deterministic**, the G is SLR(k) grammar.*

G_{exp} *is not SLR(1).*

Example *Parsing a*(a+a) in G_{exp}.*

$(a*(a+a), [\varepsilon]) \Rightarrow_{SLR(1)}^{a} (*(a+a), [a][\varepsilon]) \Rightarrow_{SLR(1)}^{T\to a} (*(a+a), [T][\varepsilon])$

$\Rightarrow_{SLR(1)}^{*} ((a+a), [*T][T][\varepsilon]) \Rightarrow_{SLR(1)}^{(} (a+a), [(*T][*T][T][\varepsilon])$

$\Rightarrow_{SLR(1)}^{a} (+a), [a(*T][(*T][*T][T][\varepsilon])$

$\Rightarrow_{SLR(1))}^{E\to a} (+a), [E(*T][(*T][*T][T][\varepsilon])$

$\Rightarrow_{SLR(1)}^{+} (a), [+E(*T][E(*T][(*T][*T][T][\varepsilon])$

$\Rightarrow_{SLR(1)}^{a} (), [a+E(*T][+E(*T][E(*T][(*T][*T][T][\varepsilon])$

$\Rightarrow_{SLR(1)}^{T\to a} (), [T+E(*T][+E(*T][E(*T][(*T][*T][T][\varepsilon])$

$\Rightarrow_{SLR(1)}^{E\to E+T} (), [E(*T][(*T][(*T][*T][\varepsilon])$

$\Rightarrow_{SLR(1)}^{)} (\varepsilon, [)E(*T][E(*T][(*T][*T][T][\varepsilon])$

$\Rightarrow_{SLR(1)}^{F\to (E)} (\varepsilon, [F*T][*T][T][\varepsilon])$

$\Rightarrow_{SLR(1)}^{E\to T*F} (\varepsilon, [E][\varepsilon])$ ***accept*** *a*(a+a).*

Def. *LR(k) item:* $[A \to \alpha\bullet\beta, x]$, *if* $A \to \alpha\beta \in P$ *and* $x \in T^{\leq k}$.

x *is called the* ***lookahead*** *of the LR(k) item* $[A \to \alpha\bullet\beta, x]$.

Alg. *Collection of* ***set of*** *LR(k)* ***items***(*LR(k)* ***states***): $C_k \subseteq 2^{I_G}$.

$C_k := \{\varepsilon_k^*([S' \to \bullet S, \varepsilon])\}$ /* $[\varepsilon]$ */

where $\varepsilon_k([A \to \alpha'\bullet B\gamma', x]) = [B \to \bullet\beta, y], y \in First_k(\gamma' x) \in Follow(B)$

repaeat

for $q \in C_k$ ***do***

for $[\{[A \to \alpha\bullet X\gamma, x]\} \in q$ ***do*** *where* $X \in N \cup T$ /* $q = [\alpha^R\delta^R]$ */

$p := \varepsilon_k^*(\{[A \to \alpha X\bullet\gamma, x]\}$; /* $p = [X\alpha^R\delta^R]$ */

$(X, q) \to^X (\varepsilon, p)$; /* $(X, [\alpha^R\delta^R]) \to^X (\varepsilon, [X\alpha^R\delta^R])$ */

$C_k := C_k \cup \{q\}$

od od

until C_k *does not increase*

LR(k) ***parser*** *for G = (N, T, P, S) is*

$P_{LR(k)} = (T^{*} \times C_k^{*}, \rightarrow_{LR(k)}, (x, [\varepsilon]_k), (\varepsilon, [\varepsilon]_k[S]_k))$ *where*

1) ***shift*** *a:* $a \in T$

If $[A \rightarrow \alpha \bullet a\beta, ax] \in [\alpha^R\delta^R]$, $ax \in T^{\leq k}$, *then*

$(a, [\delta^R\alpha^R]) \rightarrow_{LR(k)}^{a} (a, [a\alpha^R\delta^R][\alpha^R\delta^R])$

2) ***reduce*** α *to A:* $A \rightarrow \alpha \in P$

If $[A \rightarrow \alpha \bullet, x] \in [\alpha^R\delta^R]$ *where* $\alpha = X_1X_2 \ldots X_n$, *then*

$(x, [X_nX_{n-1}\ldots X_1\delta^R][X_{n-1}\ldots X_1\delta^R] \ldots [X_2X_1\delta^R][X_1\delta^R][\delta^R])$

$\rightarrow_{LR(k)}^{A\rightarrow\alpha} (x, [\delta^R A[\delta^R]])$

$x \in Follow(A)$

If LR(k) parser for G, $P_{LR(k)}$, *is deterministic G is called LR(k)* ***grammar***.

LR(k) ***parser*** *(= LR(k, k)* ***parser****; LR(k) Lookahead)*

state: LR(k) states

reduce: LR(k) lookahead

LALR(k) ***parser*** *(= LR(0, k)* ***parser****; LR(k) Lookahead)*

state: LR(0) states

reduce: LR(k) lookahead $\subset Follow_k(A)$

SLR(k) ***parser*** *(= SLR(0, k)* ***parser****;* $Follow_k(A)$*)*

state: LR(0) states

reduce: $Follow_k(A)$

SLR(k) grammars ⊂ *LALR(k) grammars* ⊂ *LR(k) grammars*

SLR(k) ⊂ *SLR(k+1) and LALR(k)* ⊂ *LALR(k+1)* **but** *LR(1) = LR(k)!*

In yacc(Yet Another Compiler-Compiler)

Try LR(0); ***if not****, try SLR(1);* ***if not****, try LALR(1)!* ***If not****, try LR(1)!*