

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 .

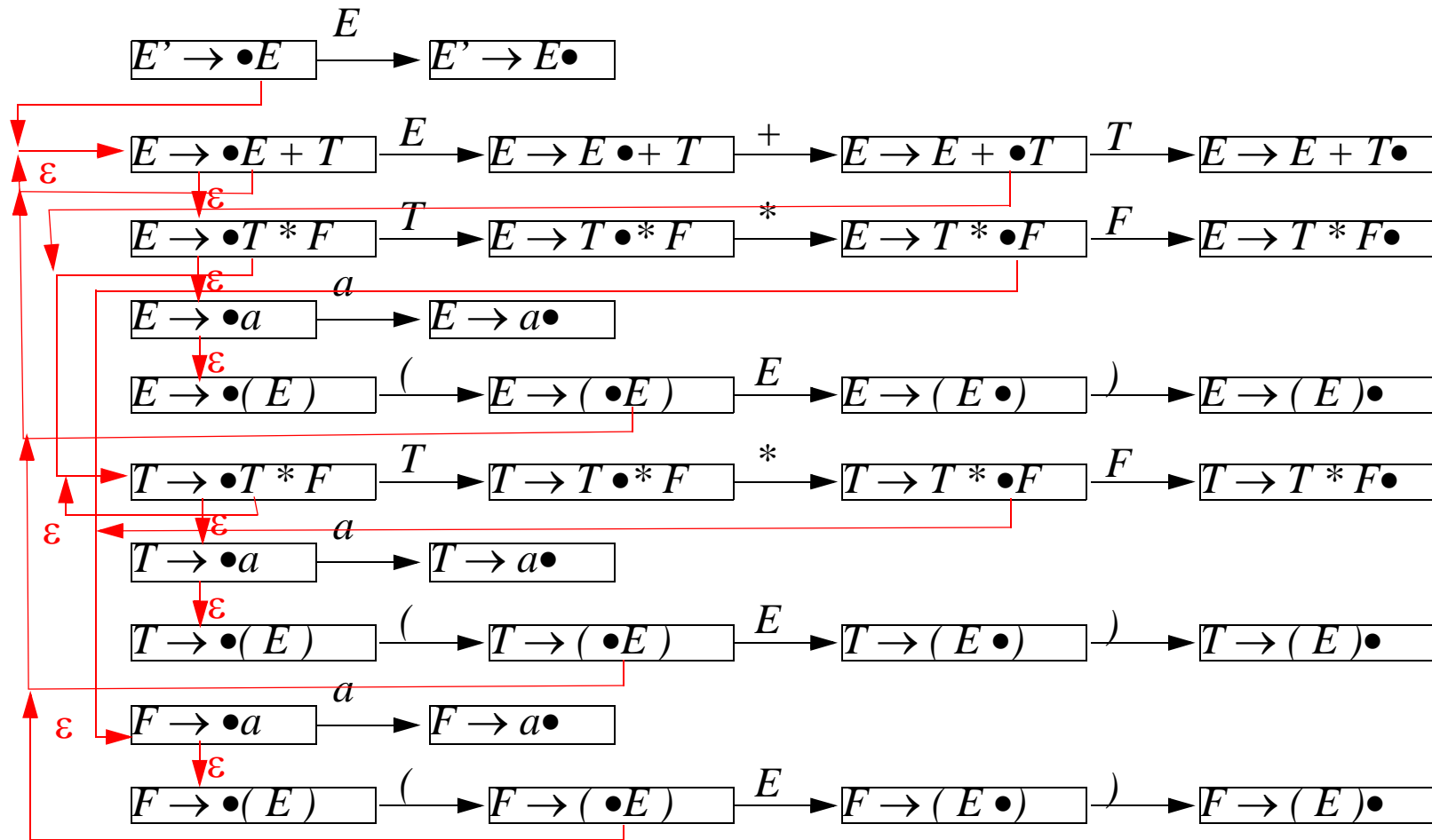
Def. An **augmented** grammar of $G = (N, T, P, S)$ is
 $G' = (N \cup \{S'\}, T, P \cup \{S' \rightarrow S\}, S')$ where $S' \notin N$.

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 $X \in N \cup T /* q = [\alpha^R \delta^R] */$

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

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

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

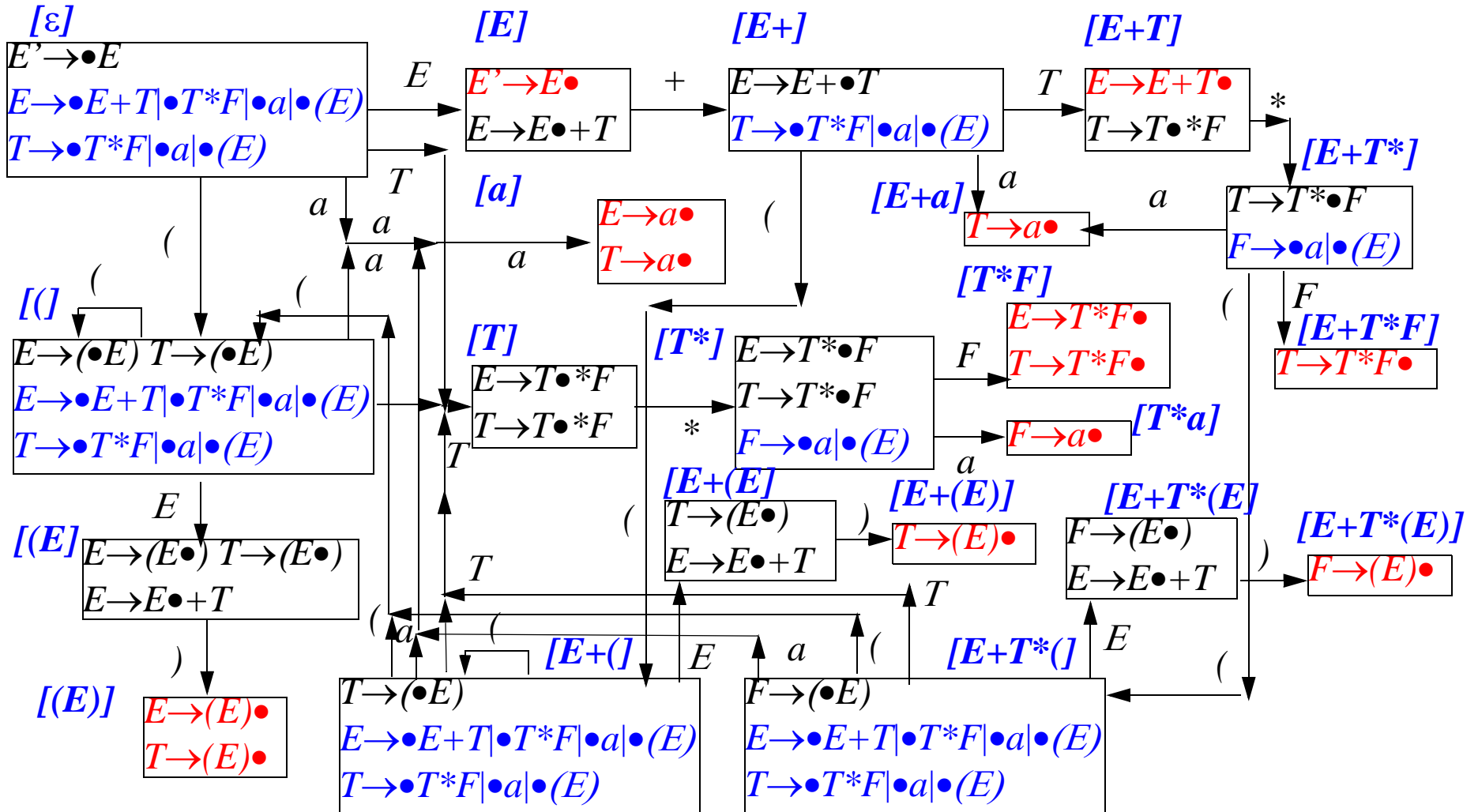
od od until C_0 does not change

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} :



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

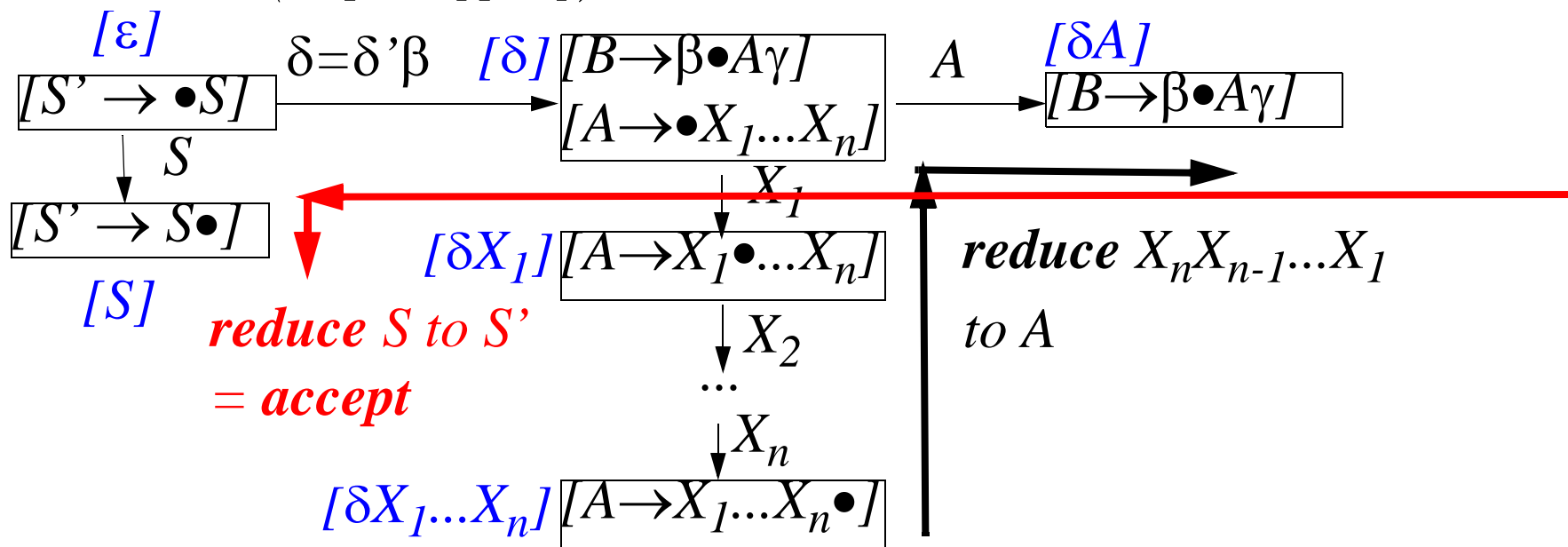
1) **shift** a : $a \in T$ $[A \rightarrow \alpha \bullet a \beta] \in [\alpha^R \delta^R]$.

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

2) **reduce** α to A : $A \rightarrow \alpha \in P$ $[A \rightarrow \alpha \bullet] \in [\alpha^R \delta^R]$ where $\alpha = X_1 X_2 \dots X_n$.

$$(\epsilon, [X_n X_{n-1} \dots X_1 \delta^R][X_{n-1} \dots X_1 \delta^R] \dots [X_2 X_1 \delta^R][X_1 \delta^R][\delta^R])$$

$$\rightarrow (\epsilon, [A \delta^R][\delta^R])$$



initial configuration $\iota = (x, [\epsilon])$

final configuration $\Phi = \{(\epsilon, [S][\epsilon])\}$

LR(0) parser for $G = (N, T, P, S)$. If

$$P_{LR(0)} = (T^* \times C_0^*, \rightarrow_{LR(0)}, (x, [\epsilon]), (\epsilon, [S][\epsilon]))$$

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

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

no shift-shift conflict!

2) *reduce* α to A : $A \rightarrow \alpha \in P$ $[A \rightarrow \alpha \bullet] \in [\alpha^R \delta^R]$ where $\alpha = X_1 X_2 \dots X_n$

$$(\epsilon, [X_n X_{n-1} \dots X_1 \delta^R][X_{n-1} \dots X_1 \delta^R] \dots [X_2 X_1 \delta^R][X_1 \delta^R][\delta^R])$$

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

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

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

G_{exp} is not LR(0).

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

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

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

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

2) **reduce** α to A : $A \rightarrow \alpha \in P \quad [A \rightarrow \alpha \bullet] \in [\alpha^R \delta^R]$ where $\alpha = X_1 X_2 \dots X_n$

$(x, [X_n X_{n-1} \dots X_1 \delta^R][X_{n-1} \dots X_1 \delta^R] \dots [X_2 X_1 \delta^R][X_1 \delta^R][\delta^R])$

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

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} .

$$\begin{aligned}
& (a^*(a+a), [\varepsilon]) \Rightarrow_{SLR(1)}^a (* (a+a), [a][\varepsilon]) \Rightarrow_{SLR(1)}^{T \rightarrow a} (* (a+a), [T][\varepsilon]) \\
& \Rightarrow_{SLR(1)}^* ((a+a), [*T][T][\varepsilon]) \Rightarrow_{SLR(1)}^{} (a+a, [(*T)[*T][T][\varepsilon]) \\
& \Rightarrow_{SLR(1)}^a (+a), [a(*T)[(*T)[*T][T][\varepsilon]) \\
& \Rightarrow_{SLR(1)}^{E \rightarrow a} (+a), [E(*T)[(*T)[*T][T][\varepsilon]) \\
& \Rightarrow_{SLR(1)}^+ (a), [+E(*T)[E(*T)[(*T)[*T][T][\varepsilon]) \\
& \Rightarrow_{SLR(1)}^a (), [a+E(*T)[+E(*T)[E(*T)[(*T)[*T][T][\varepsilon]) \\
& \Rightarrow_{SLR(1)}^{T \rightarrow a} (), [T+E(*T)[+E(*T)[E(*T)[(*T)[*T][T][\varepsilon]) \\
& \Rightarrow_{SLR(1)}^{E \rightarrow E+T} (), [E(*T)[(*T)[(*T)[*T][\varepsilon]) \\
& \Rightarrow_{SLR(1)}^{} (\varepsilon, [)E(*T)[E(*T)[(*T)[*T][T][\varepsilon]) \\
& \Rightarrow_{SLR(1)}^{F \rightarrow (E)} (\varepsilon, [F* T][* T][T][\varepsilon]) \\
& \Rightarrow_{SLR(1)}^{E \rightarrow T^* F} (\varepsilon, [E][\varepsilon]) \quad \text{accept } a^*(a+a).
\end{aligned}$$

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

\mathbf{x} is called the lookahead of the LR(k) item $[A \rightarrow \alpha \bullet \beta, \mathbf{x}]$.

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

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

where $\varepsilon_k([A \rightarrow \alpha' \bullet B \gamma', \mathbf{x}]) = [B \rightarrow \bullet \beta, \mathbf{y}]$, $\mathbf{y} \in \text{First}_k(\gamma' \mathbf{x})$

repeat

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

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

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

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

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

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

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

2) **reduce** α to A : $A \rightarrow \alpha \in P$ $[A \rightarrow \alpha \bullet, x] \in [\alpha^R \delta^R]$ where $\alpha = X_1 X_2 \dots X_n$.

$(x, [X_n X_{n-1} \dots X_1 \delta^R][X_{n-1} \dots X_1 \delta^R] \dots [X_2 X_1 \delta^R][X_1 \delta^R][\delta^R])$

$\rightarrow (x, [\delta^R A [\delta^R]])$

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

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

state: LR(0) states

reduce: LR(k) lookahead

In yacc(Yet Another Compiler-Compiler)

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