Alg goto 1 $\mathcal{P}(\mathcal{E}_1) \times (N \cup \Sigma) \to \mathcal{P}(\mathcal{E}_1)$.

$goto1(D, X) = closure1(\{[A \to \alpha X . \beta, u] \mid [A \to \alpha . X\beta, u] \in D\})$

$[A \to \alpha . B\beta, u]$ - valid for $\gamma\alpha$, $u \in FIRST(w)$.

$S \underset{r}{\overset{*}{\Rightarrow}} \gamma A w \underset{r}{\Rightarrow} \underbrace{\gamma\alpha} B\beta w \underset{r}{\overset{*}{\Rightarrow}} \gamma\alpha B w' w \Rightarrow$

$\underset{r}{\overset{B \to \delta}{\Rightarrow}} \underbrace{\gamma\alpha\delta} w' w$.

$[B \to . \delta, \boxed{v}]$ $\quad v \in FIRST(w'w)$.

$v \in F(\beta u)$ $\quad FIRST(w'w) = F(w') \oplus F(w) =$

$= FIRST(\beta) \oplus u = F(\beta u)$.

Alg closure1

input G', FIRST, I

output C_1

$C_1 = \{I\}$

Repeat.

for $\forall [A \to \alpha . B\beta, u] \in C_1$ do.

for $\forall B \to \delta \in P$ do.

for $\forall v \in FIRST(\beta u)$ do

C_1.append($[B \to . \delta, v]$).

until C_1 is not changed.

Example. $S' \to S$.

1) $S \to AA$

$A \to aA$

$A \to b$

(date) 15.12.2015

$D_0 = closure(\{[S' \to . S, \$]\}) = \{[S' \to . S, \$], [S \to . AA, \$],$
$[A \to . aA, a], [A \to . aA, b], [A \to . b, a], [A \to . b, b]\}$ $\quad First(A) = \{a, b\}$

LR(1) table

	action $\Sigma \cup \{\$\}$	goto N
D_0		

in SLR

Rules for LR(1) table.

1. if $[A \to \alpha . \beta, u] \in D_i$, $\beta \neq \varepsilon$, $goto(D_i, v) = D_j$ $\Rightarrow$ $action(D_i, v) = shift\ D_j$
2. if $[A \to \alpha\beta ., u] \in D_i$, $A \neq S'$, $A \to \alpha\beta \in P(m)$ $\Rightarrow$ $action(D_i, u) = reduce(m)$.
3. if $[S' \to S ., \$] \in D_i \Rightarrow action(D_i, \$) = accept$
4. $goto(D_i, A) = D_j$; 5 otherwise error; 6. initial state $\ni [S' \to . S, \$]$

Remark: cfg is LR(1) if the table does not contain conflicts.

3) config + moves - SLR $\quad l: A \to \alpha$

$[A \to \alpha ., u] \in D_i \xrightarrow{reduce\ l} D_m$

$[A \to \alpha ., v] \in D_j \xrightarrow{reduce\ l} D_m$

Look Ahead. LR(1) $\Rightarrow$ LALR

to action before prediction.

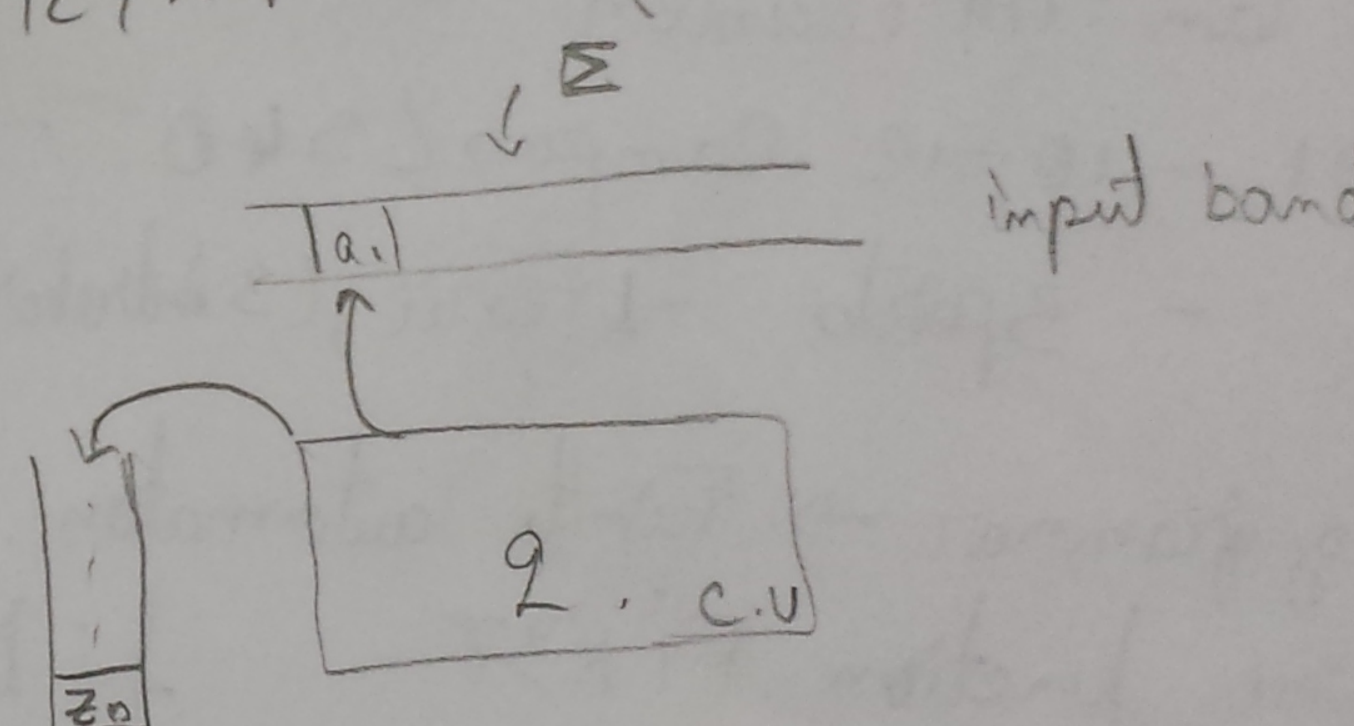
LALR Principle.
Merge states with the same kernel $s_i: [A \to \alpha., u]$ and $[A \to \alpha., v] : s_j$ in
$s_{i,j} \ni [A \to \alpha., u|v]$ only if no conflicts are created.

$\boxed{\text{no of states in LALR} = \text{no of states SLR / LR(0)}}$

Recap

cfg $\Longleftrightarrow$ PDA (push down automata)

Def. A push down automaton (PDA) is a 7-tuple: $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$
where: Q - finite set of states ($|Q| < \infty$).
Σ - input band alphabet ($|\Sigma| < \infty$).
Γ - stack alphabet. ($|\Gamma| < \infty$).
δ - transition func. $\delta : Q \times \Sigma \cup \{\varepsilon\} \times \Gamma \to \mathcal{P}(Q \times \Gamma^*)$
$q_0 \in Q$, initial state.
$Z_0 \in \Gamma$, initial stack symbol.
$F \subseteq Q$ set of final states.



Config and moves

(q, w, γ), $q \in Q$, $w \in \Sigma^*$, $\gamma \in \Gamma^*$

Direct move $(q, aw, Z\gamma) \vdash (p, w, \alpha\gamma)$ iff $\delta(q, a, Z) = (p, \alpha)$

$\overset{k}{\vdash}, \overset{+}{\vdash}, \overset{*}{\vdash}$

Def: Language accepted by a PDA M:
a) final state principle: $L(M) = \{ w \in \Sigma^* \mid (q_0, w, Z_0) \overset{*}{\vdash} (q_f, \varepsilon, \gamma), q_f \in F \}$
b) empty stack principle: $L(M) = \{ w \in \Sigma^* \mid (q_0, w, Z_0) \overset{*}{\vdash} (q, \varepsilon, \varepsilon) \}$.