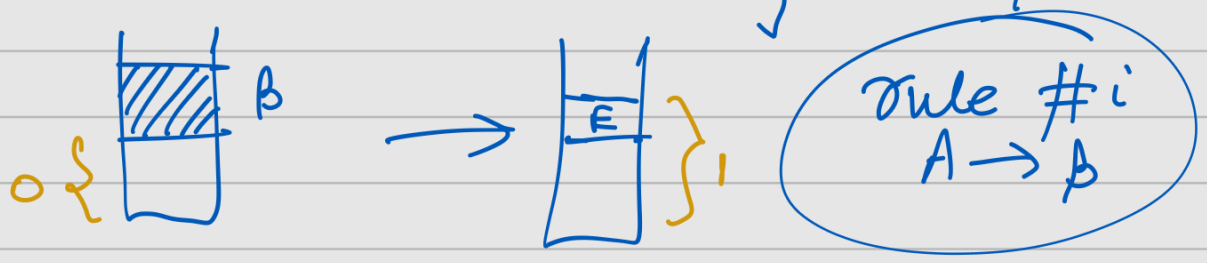
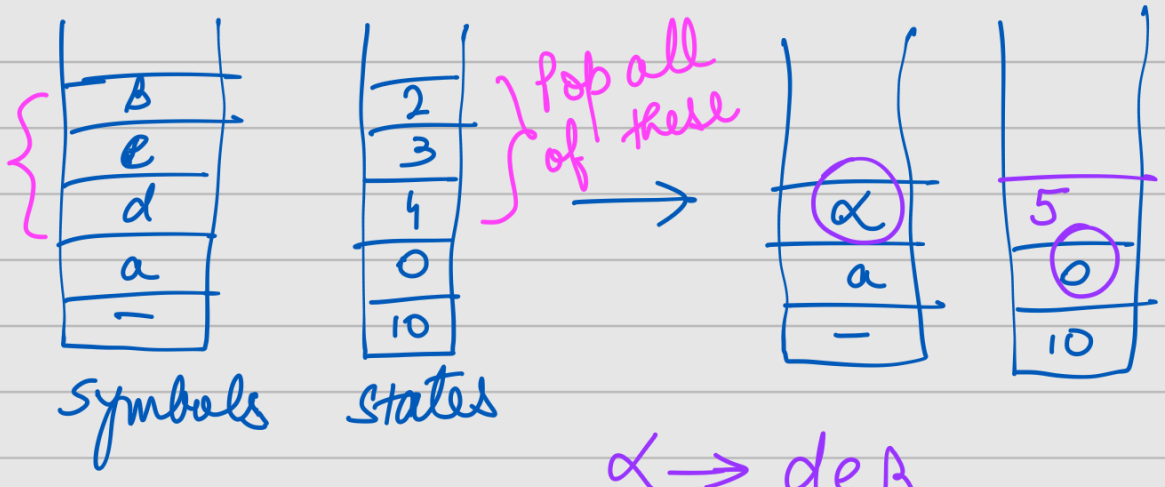
Lec 24/03

Parse Table for Shift Reduce

s_i → shift and goto state i

r_i → reduce using rule # i



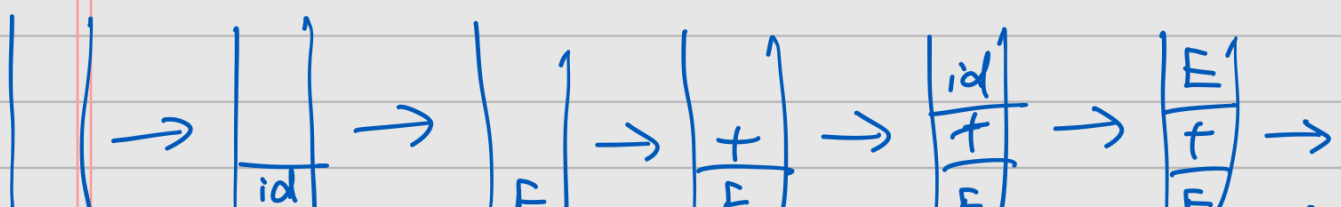


$\alpha \rightarrow de\beta$

State 5 is found using state 0 & symbol α.

- additional prodⁿ $S' \rightarrow S\$$ so that we always know when to stop as $S \rightarrow \alpha/\beta/i\$$.

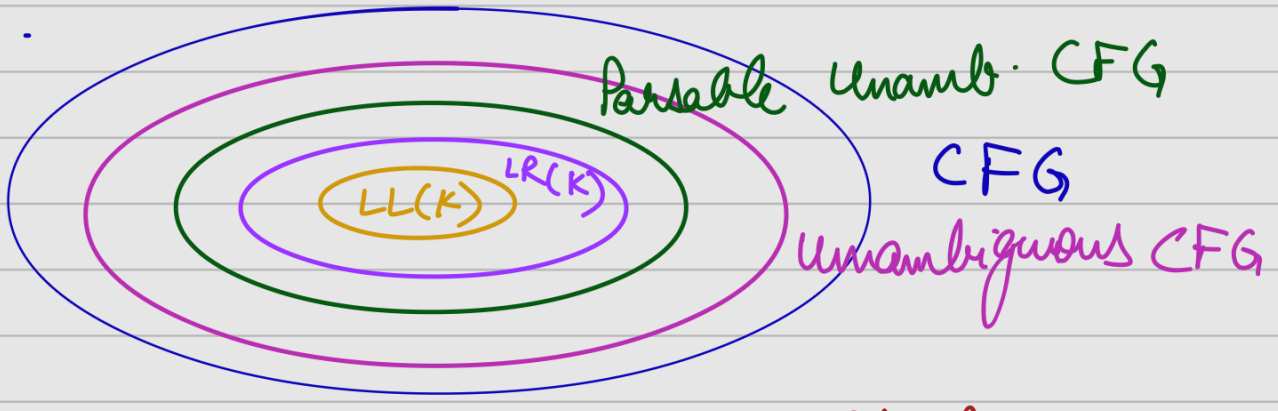
$E \rightarrow id+E/id$ $\quad$ $id+id+id$



(bad strategy: reduce whenever you can) → E + E + E

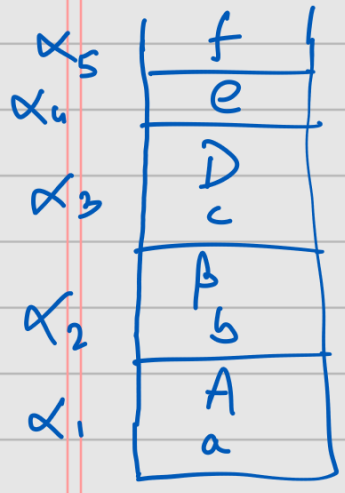
Handle — β is a handle if reducing β to A keeps me on the correct path (i.e. rightmost derivation)

"Whenever u see a handle, prune it, i.e. reduce it & u are done".

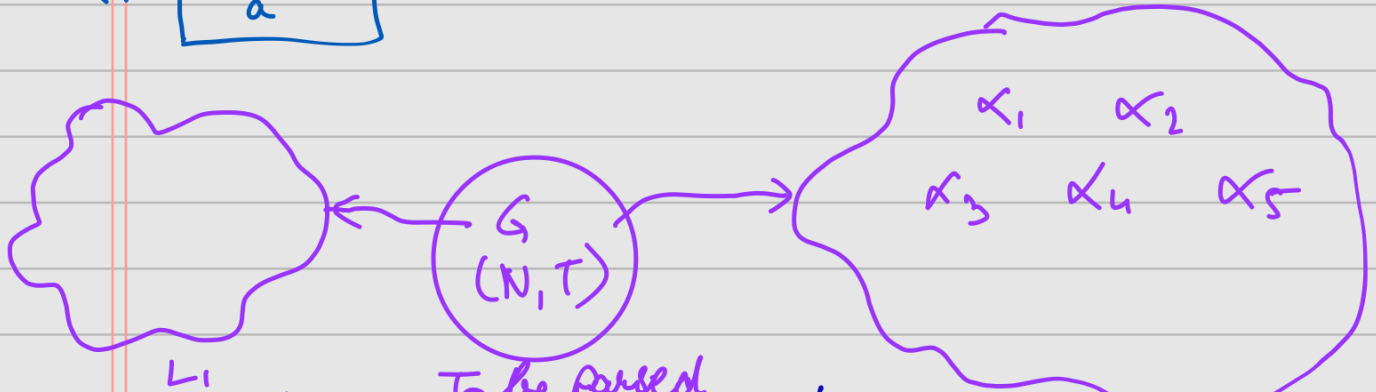


outside LR ⇒ we can get a Shift Reduce conflict

"Valid Parser Stack": sequences of prefixes of handles.



α_i — Viable prefixes. ↳ ≡ viable stack config^n → seq of prefixes of handles



(lang of G_1)

L_2 (lang. cont. all viable prefixes of gramm G_1)

↳ Terminals of $L_2 \in T \cup N$

entire

read the i stack config^n as a string and do membership check on the α_i's reg lang