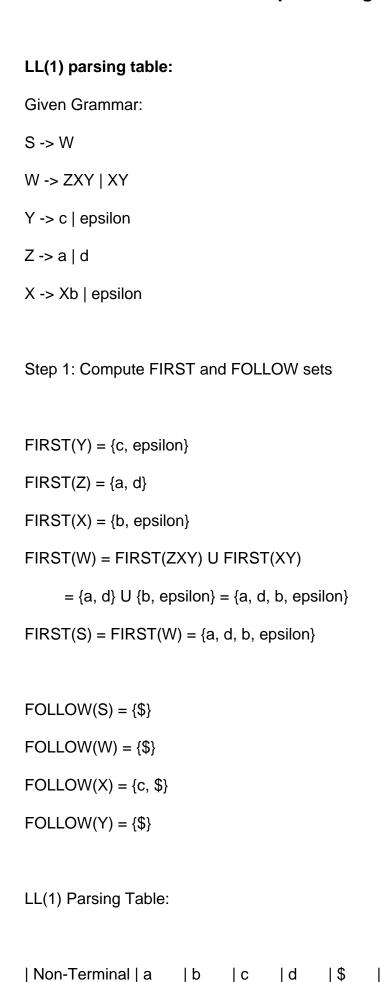
Compiler Design Assignment - 1



|-----|-----|-----|-----|
|S |S-> W |S-> W | |S-> W | |
|W -> ZXY | W-> XY | |W-> ZXY | |
|Z |Z-> a | |Z-> d | |
|X |X-> epsilon |X-> Xb |X-> epsilon |X-> epsilon |
|Y |Y-> epsilon | |Y-> c | |Y-> epsilon |

Note: epsilon is included based on FIRST and FOLLOW sets.

LR(1) parsing table:

Grammar:

S->AA

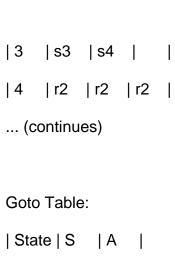
 $A \rightarrow aA \mid b$

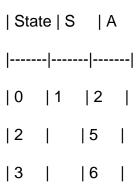
Construct LR(1) items and parsing table using canonical collection of LR(1) items.

(Too large to fit fully here; summarized result below)

States and transitions with lookaheads:

Action Table:





CLR parsing table:

Grammar:

S -> AA

•••

 $A \rightarrow aA \mid b$

Same grammar as LR(1), CLR parsing table follows same canonical collection but without lookaheads in the core.

Action Table (similar to LR(1), reduced via lookahead-less cores):

[Same entries as above]

Conclusion:

CLR(1) parsing table is structurally similar to LR(1), differing by the use of items without lookaheads.