SLR Parsing Example

1. Parser is like an FSA + stack, so better to draw it that way.
2. Need to have states & grammar symbols on stack to keep track of PDA Computation.

Rule 1 - $S \rightarrow \%T$
Rule 2 - $T \rightarrow T \& A$
Rule 3 - $T \rightarrow A$
Rule 4 - $A \rightarrow id$

State 0
$S \rightarrow \%T$

State 1
$S \rightarrow \%T$
$T \rightarrow T \& A$
$A \rightarrow id$

State 2
$S \rightarrow \%T$
$T \rightarrow T \& A$

State 3
$T \rightarrow A$

State 4
$A \rightarrow id$

State 5
$T \rightarrow T \& A$

State 6
$T \rightarrow T \& A$

This is our recognizer - now to build table from it and then trace the parse. Note there are SLR not LR items (no lookahead).
**SLR Parsing Example**

Compute follow sets for $S, T, A$.

- $\text{Follow}(S) = \{ \$ \}$
- $\text{Follow}(T) = \{ \$, \$, \$ \}$, and $\$$(\because S \rightarrow \$$T$$)$
- $\text{Follow}(A) = \{ \$, \$$ \}$, and $\$$(\because T \rightarrow A$$)$
- $\text{Follow}(T) \subset \text{Follow}(A)$

<table>
<thead>
<tr>
<th>State</th>
<th>Input</th>
<th>Action</th>
<th>Go to</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$$</td>
<td>$S_1$</td>
<td>$T$</td>
</tr>
<tr>
<td>1</td>
<td>accp</td>
<td>$S_4$</td>
<td>$2, 3$</td>
</tr>
<tr>
<td>2</td>
<td>r$3$</td>
<td>$r_3$</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>r$4$</td>
<td>$r_4$</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>$S_4$</td>
<td>$6$</td>
</tr>
<tr>
<td>5</td>
<td>r$2$</td>
<td>$r_2$</td>
<td>$1$</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$S_K$ means shift and go to state $K$, $r_J$ means reduce = pop the handle, push the rhs nonterminal & continue.
Parse sequence for S -> 50 T and id - 3-

Stack -> top

$0

$0 %1

$0 %1 id 4

$0 %1 A3

$0 %1 T2

$0 %1 T2 and 5

$0 %1 T2 and 5 id 4

$0 %1 T2 and 5 A6

$0 %1 T2

input

S -> id and id $  // initial state

id and id $

and id $

// reduce to A and in state 1 so goto state 3

id $

// reduce to T and in state 1 so goto state 2

id $

// reduce to A and in state 5 so goto state 6

$  // reduce to T and in state 1 so goto state 2

ACCEPT

Have rhs of S -> 50 T on top of stack and $ in output