

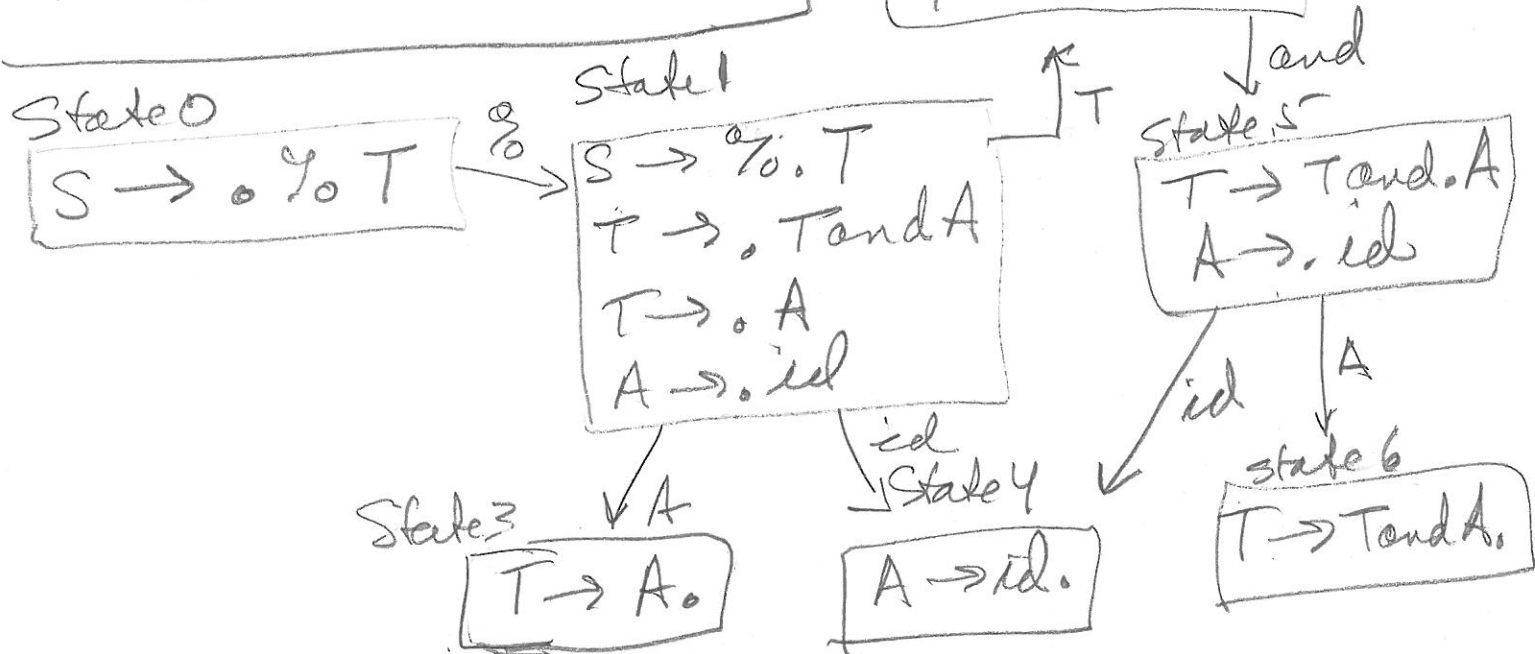
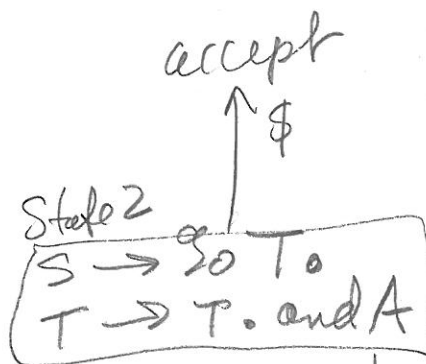
SLR Parsing Example!

PDA

(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 - $T \rightarrow T \text{ and } A$
- Rule 3 - $T \rightarrow A$
- Rule 4 - $A \rightarrow id$



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

SLR Parsing Example

Compute Follow sets for S, T, A.

Follow(S) = { \$ } ; Follow(T) = { \$, and }

Follow(A) = { \$, and }

because $S \rightarrow T$

Follow(S) \subseteq Follow(T)

because $T \rightarrow A$

Follow(T) \subseteq Follow(A)

State	Input	Action				Goto:	
		\$	ϵ	and	u	T	A
0			s1				
1					s4	2	3
2	accept			s5			
3	r3			r3			
4	r4			r4			
5					s4		6
6	r2			r2			

s_k means shift and go to state k
 r_j means reduce \equiv pop the handle,
 push the lhs nonterminal Φ
 continue.

Parse sequence for $go\ id\ and\ id$ - 3-

Stack \rightarrow top

input

\$ 0

$go\ id\ and\ id\ \$$ // state 0 is initial state

\$ 0 $go\ 1$

$id\ and\ id\ \$$

\$ 0 $go\ 1\ id\ 4$

$and\ id\ \$$

\$ 0 $go\ 1\ A\ 3$

$and\ id\ \$$ // reduce to A and in state 1 so goto state 3.

\$ 0 $go\ 1\ T\ 2$

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

\$ 0 $go\ 1\ T\ 2\ and\ 5$

$id\ \$$

\$ 0 $go\ 1\ T\ 2\ and\ 5\ id\ 4$

$\$$

\$ 0 $go\ 1\ T\ 2\ and\ 5\ A\ 6$

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

\$ 0 $go\ 1\ T\ 2$

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

ACCEPT

Have rhs of $S \rightarrow go\ T$
on top of stack and
 $\$$ in input