COMPILERS CS31003

## Arpit Tarang Saxena, 11CS30005

#### Date: 9/10/2013

**Differences between LR(1) and SLR :**

**SLR : [ A 🡪 α · a β ]**

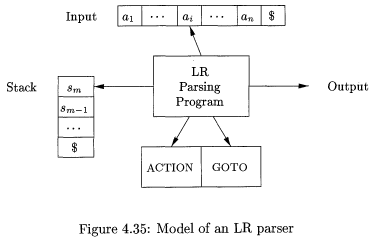
**LR(1): [ A 🡪 α · a β, b ] where ‘b’ is the *lookahead*.**

**G1: SLR Grammar: SLR Table (No Conflicting entries)**

**G2: Non SLR Grammar.**

**LR(1) Automaton:**

1. **Parse Table: Action, Goto**
2. **State i, Ii ϵ [**

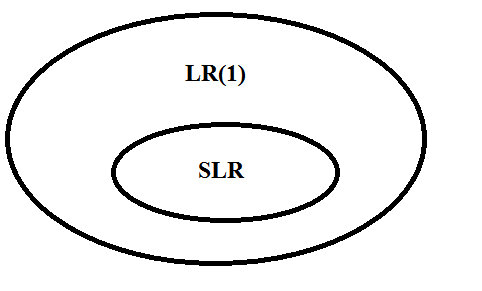
**An LR Parser consists of an input, an output, a stack, a driver program, and a parsing table that has two parts: ACTION and GOTO. The driver program is the same for all LR parsers (including SLR); only the parsing table changes from one parser to another. The parsing program reads characters from an input buffer one at a time. Where the shift-reduce parser would shift a symbol, an LR parser shifts a state. Each state summarizes the information contained in the stack below it.**

**Structure of the LR Parsing Table:**

The parsing table consists of two parts: a parsing function ACTION and a goto function GOTO.

1. The ACTION function takes as arguments a state *i* and a terminal *a* (or $, the input end marker). The value of ACTION[*i*, *a*] can have one of four forms:
   1. Shift *j*, where *j* is a state. The action taken by the parser effectively shifts input *a* to the stack, but uses state *j* to represent *a*.
   2. Reduce A 🡪 β. The action of the parser effectively reduces β on the top of the stack to head A.
   3. Accept. The parser accepts the input and finishes parsing.
   4. Error. The parser discovers an error in its input and takes some corrective action.
2. Extend the GOTO function, defined on sets of items, to states: if GOTO[ Ii, A] = Ij, then GOTO also maps a state *i* and a nonterminal A to state *j*.

**Comparison of Parsers:**



**LALR Parsers:**

Goal:

1. Reduce the number of states in LR(1) parser.
2. Keep the same grammar.

**Construct Automata for LALR Parser:**

1. Construct LR(1) automata (parsing table)
2. *Core Items*: Those that share the same derivations as the first component. Identify those states which share the same core. Aggregate these states.