Compilers

Grammars

LL (2)

A grammar is LL(2) if it can be parsed by an LL parser that uses 2 tokens of lookahead.
- **Is LL(2)**: - The parser can decide which production to use by looking at the current input symbol and the next input symbol (2 tokens total). - No left recursion is allowed. - No common prefixes of length 2 or less in the right-hand sides of productions for the same non-terminal.
- **Is not LL(2)**: - If it requires more than 2 tokens of lookahead to make parsing decisions. - If it contains left recursion. - If it has common prefixes of length 2 or less in the right-hand sides of productions for the same non-terminal.

LR (1)

A grammar is LR(1) if it can be parsed by an LR parser that uses 1 token of lookahead.
- **Is LR(1)**: - The parser can decide which action to take (shift, reduce, or accept) by looking at the current state and the next input symbol. - No shift/reduce or reduce/reduce conflicts when constructing the LR(1) parsing table.
- **Is not LR(1)**: - If there are any shift/reduce or reduce/reduce conflicts in the LR(1) parsing table. - If it requires more than 1 token of lookahead to resolve parsing decisions.

LR (2)

A grammar is LR(2) if it can be parsed by an LR parser that uses 2 tokens of lookahead.
- **Is LR(2)**: - The parser can decide which action to take by looking at the current state and the next two input symbols. - No shift/reduce or reduce/reduce conflicts when constructing the LR(2) parsing table.
- **Is not LR(2)**: - If there are any shift/reduce or reduce/reduce conflicts in the LR(2) parsing table. - If it requires more than 2 tokens of lookahead to resolve parsing decisions.

LALR (1)

LALR(1) (Look-Ahead LR) is a subset of LR(1) grammars that can be parsed with a more compact parser.
- **Is LALR(1)**: - It is LR(1). - When LR(1) states with the same core are merged, no new conflicts are introduced.
- **Is not LALR(1)**: - If it's not LR(1). - If merging LR(1) states with the same core introduces new conflicts.

SLR (1)

SLR(1) (Simple LR) is the simplest and most restrictive subset of LR(1) grammars.
- **Is SLR(1)**: - It is LR(0) (can be parsed with no lookahead). - Any remaining conflicts in the LR(0) parsing table can be resolved using the FOLLOW sets of the non-terminals.
- **Is not SLR(1)**: - If it's not LR(0). - If conflicts remain after considering the FOLLOW sets of non-terminals.