**Muhammad Hasan Khan**

**2019328**

**CS-424 Assignment 2**

**CODE REPORT:**

The code implements a parser for a context-free grammar (CFG) using the LR1 parsing technique. The LR1 parser works by maintaining a stack of states, where each state is a set of items. Each item is a production rule from the grammar with a "dot" indicating the current position of the parser in that rule. The parser reads input symbols from left to right, shifting them onto the stack, and reducing rules when the parser reaches the end of a rule (i.e., when the dot is at the end of the rule).

The Grammar class represents a CFG and provides methods for computing the closure and go to sets of LR1 items. The Item class represents an LR1 item and provides methods for advancing the dot position and computing the production associated with the item.

The build\_table function builds the parse table for the grammar using the canonical LR1 collection method. The parse table is a two-dimensional dictionary where each entry maps a state and a symbol to an action or a new state. The action can be a shift (denoted by 'S') or a reduce (denoted by 'R'), and the new state is an index into the canonical collection of LR1 item sets.

The parse function uses the parse table to parse an input string. It maintains a stack of states and a cursor to the current position in the input string. It repeatedly looks up the current state and input symbol in the parse table and performs the corresponding action. If the input string is accepted, the function returns True; otherwise, it returns False.

In the example provided, the grammar represents simple arithmetic expressions with addition and multiplication operators. The input string "id+id\*id" is parsed successfully, resulting in the output True.