**Assignment #02**

**CS424 – Compiler Construction**

**Report**

Introduction:

LR(1) parser is a type of shift-reduce parser that parses a sentence from left to right using a rightmost derivation in reverse. It can parse any context-free grammar, and it can detect shift/reduce and reduce/reduce conflicts. In this report, we will discuss the implementation of an LR(1) parser using the Python programming language.

Implementation:

The given Python code implements an LR(1) parser using the following classes and functions:

1. Grammar class: It represents the grammar and its productions. It also calculates the terminals and non-terminals of the grammar.
2. Item class: It represents an item in the LR(1) parsing table. Each item contains a production, a dot position, and a lookahead.
3. build\_table function: It builds the parsing table for the given grammar.
4. parse function: It uses the parsing table to parse the input string and returns True if the string can be derived from the given grammar, else False.

The LR(1) parser implemented in the given code is based on the canonical LR(1) parsing algorithm. The algorithm works as follows:

1. Create the initial state of the parser by taking the closure of the item {S' -> .S, $}.
2. For each state in the parser, compute the GOTO function for each symbol in the grammar. If the resulting state is new, add it to the parser.
3. Compute the parsing table by populating the ACTION and GOTO entries for each state and symbol.
4. Parse the input string using the parsing table by performing shift or reduce operations according to the entries in the parsing table.
5. If the string can be derived from the grammar, return True; otherwise, return False.

The LR(1) parsing algorithm guarantees that there are no shift/reduce conflicts, and it can handle most of the context-free grammars.

Conclusion:

In this report, we discussed the implementation of an LR(1) parser using the Python programming language. The LR(1) parsing algorithm is a powerful parsing algorithm that can handle most of the context-free grammars. The given Python code implements an LR(1) parser based on the canonical LR(1) parsing algorithm. The parser is implemented using the Grammar, Item, build\_table, and parse functions. The parser can parse an input string and determine if it can be derived from the given grammar or not.