Comp 442 A1

Alexander De Laurentiis

March 6, 2022

1 Attribute Grammar

Last minute compile errors have led me to exclude it from here but it would be available to be traced in an almost syntactically identicle fashion starting from line 879 of my program.

2 Design

Taking the input of a semantically correct file, it will start with converting it to a wordlist by parsings its structure of the characterlist of the file. Then it will parse out the comments then attempt to break it down into its abstract syntax tree format. The usage of prolog to do so was due to its ability to translate the sytax of the language directly into a ast and fit it assuming a syntactically correct program. With how it is constructed, I can test each branch of my program by calling $phrase(< desired - branch > (X), < token - list >)$. from within my program. This has accelerated testing and will help show validity of certain paths assuming a correct LL1 grammar which allows each path.

3 Tools Used

3.1 Scryer-Prolog

The main driver, interpreter, and lifeline of this project. Around line 1113 of my code will be the main ast generator predicate which if given the name of an input and output file, it will execute the process to generate the ugly output of the ast. In order to prettyfy it the python function must be used from the cli with the command in the next section.

3.2 Python

Used to prettify the output of the prolog program because its outout was of the datastructure created into a file. The script to do so can be called as in the way $pythonprettify.py < inputfile >< outputfile >$.