First Order Logic Parser Documentation

This program takes a first-order logic input file, parses the given formula and outputs the grammar used to form it and a parse tree corresponding to this grammar.

Installation

The program requires Python 3.5 or above with the *graphviz* interface module and the *graphviz* binaries, these can be installed as follows.

The latest release of Python can be obtained from https://www.python.org/downloads/. It has been tested using Python 3.7.

The graphviz python interface module can be installed using *pip install graphviz*, for Python 3.7 the package version should be around 0.13.2.

The graphviz binaries can either be downloaded for windows as an installer or zip file from graphviz.gitlab.io/_pages/Download/Download_windows.html. The zip file download is recommended if there is no administrator access.

Once installed or extracted, take note of the path, the *bin* folder needs to be added to the system environment variables. To do this, in the start menu search "Environment Variables" (or right click on "This PC" in file explorer, click "Properties" then "Advanced system settings" then "Environment Variables"). If opened with administrator privileges, find the "Path" variable under "System variables" and add the path to the *bin* folder. If opened without administrator privileges, try add the *bin* folder path to the '"Path" variable under "User variables for …".

If the windows environment variables cannot be accessed or changed, run the script with the '*--graphviz*' option (as documented in Usage).

Usage

All options are set by running the program on the command line, for documentation on each of the available options, run:

python foparser.py -h

If an environment variable could not be set up for graphviz, run:

python foparser.py [file] [options] --graphviz "[path to graphviz bin folder]"

In order to run the program on an input file *<file>.txt* where the (unformatted) grammar, parse tree and a log file are all output, call:

python foparser.py <file>.txt -l -g -c

The *-l* option means output to log file (default "*log.txt*" in the script directory), *-g* means output a parse tree graph (to "*<file>.parse-tree.pdf*" by default unless *-d* or *-n* are set), and *-c* means output the corresponding grammar (to "*<file>.grammar.txt*" in the script directory by default unless *-d* or *-n* are set). Adding *-f* will make the grammar more readable by adding line breaks between production rules and union productions and *-e* adds token name labels to the literal values in the grammar (variables, constants and predicate names).

If *-l* is not specified, all logging information is output to the console.

The program exists with a code of zero if everything is parsed correctly, one if there is an error with parsing or two if any invalid command line options are supplied.

Running *python foparser.py <file>.txt* will only parse *<file>.txt*, indicating success or failure (in the case of errors), logging to console.

Specifying *-d* changes the output directory (for all files excluding the log file), specifying *-o* changes the log file name (default “*log.txt*”) and specifying *-n* changes the reference name (and output prefixes) for the first-order logic being parsed.

Examples

example.txt

Running *python example.txt -l -g -c* outputs the following files.

- *example.grammar.txt*
- *example.parse-tree.pdf*
- *log.txt*

example1.txt

Running *python example1.txt -l -g -c -fe -d output -n demo* (with an existing *log.txt*) outputs the following files.

- *output/demo.grammar.txt*
- *output/demo.parse-tree.pdf*
- *log.txt* [appended to]

example2.txt

Running *python example2.txt -l -g -c -fe -d output -o output/info.log* outputs the following files.

- *output/info.log*

Only the log file is output due to the syntax error in parsing.