

# **EECS6083: Compiler Theory**

**Samir Silbak**

April 30, 2014

## 1 Project Description

For this project we were required to implement a recursive decent parser (LL(1) compiler). We start off by creating a scanner which communicates with the parser. The scanner feeds each token to the parser. The parser will then make sure syntax and semantics are correct. Once this is ensured, we will then generate C code. This generated C code is then compiled and linked with runtime functions. The language for which the compiler is written in is Python. The source code can be found in the code directory.

## 2 Usage

compiler.py was written to combine all of the source code together, hence this is the top level module. Running this Python program will generate, compile and run the executable. There are several tests that have been added in the tests directory. An example of using the program would be like so:

```
$ ./compiler.py tests/factorial.src
3628800
```

If the program is run without any arguments, the generated output will be shown:

```
usage: compiler.py [-h] filename
compiler.py: error: too few arguments
```

Passing the h flag will let the user know that the program takes only src file extensions:

```
./compiler.py -h
usage: compiler.py [-h] filename
```

EECS 6083 Compiler

```
positional arguments:
  filename      input .src file
```

```
optional arguments:
  -h, --help  show this help message and exit
```