PPL 1/29

Reagan Shirk

January 29, 2020

Regular Expressions

- He spent the first 20 minutes giving an overview of regular expressions, nothing that wasn't talked about last class and in Theory of Comp

Flex

- Flex has something to do with the .yy file we're working with in our first assignment
- We're extending the rules he made "in some way"
- I was really intimidated by this assignment but it's not looking as bad as he shows it
- You have to install flex, I think. He said you have to install something and I assumed
- Sorry my notes aren't better, most of the class has been showing code that he already has written
- I missed everything he said about the Makefile because I was setting up my VM for this class, I'll go to office hours for help and update my notes later

```
int num_lines = 0, num+chars = 0;

%% \n ++num_lines;
++num_chars;
. ++num_chars;
%%

int main()
{
    yylex();
    printf("# of lines = %d, # of chars = %d\n", num_lines, num_chars);
}
```

Scanner Stuff

- So far he's talking about DFAs, did I even need to come to class today?
- Implementation Options
  - Scanners tend to be built in three ways
    - Semi-mechanical/DFA
      - He described this but I didn't catch what he said, I think it's basically just drawing the DFA though
    - Table driven DFA
      - basically a parser
    - Ad-hoc: very case specific

Semi-Mechanical, Pure DFA

- Essentially translate regular expressions to code

- Rule: identifier → char +
- Pseudocode:

```
lexeme = "";

do {
    c := read_character(my_file);
    if (is_letter(c)) lexeme := lexeme + c;
} while (is_letter(c));

put_character_back(my_file, c);

if (lexeme != "") {
    return true;
}

return false;
```