COEN 175

Phase 2 - Week 1

TAs

- Chris Desiniotis: cdesiniotis@scu.edu
  - Office Hours: Friday 12 - 2 pm
- Antonio Gigliotti: agigliotti@scu.edu
  - Office Hours: Thursday 11 - 1 pm

Extra Help/Tutoring

- Tau Beta Pi Tutoring
- Link to Tutoring schedule
  - https://sites.google.com/scu.edu/scutaubetapi/tutoring?authuser=1&pli=1

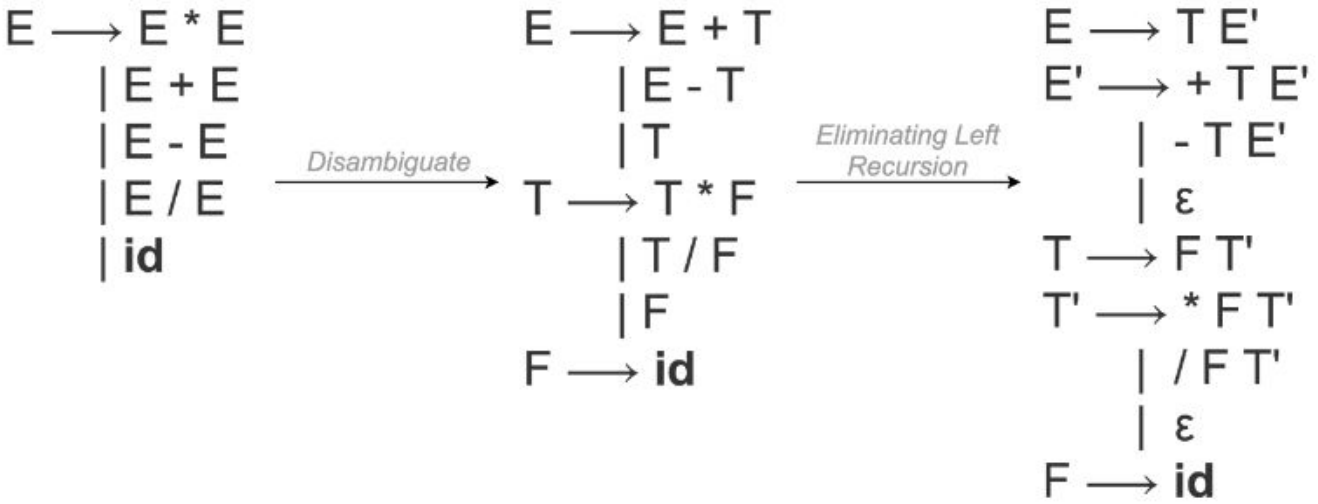
Phase 2 - Syntax Analysis

1. Disambiguate expression grammar
2. Modify lexer.cpp to return tokens from tokens.h
3. Write parser.cpp for expressions

- Due at 11:59PM on Sunday January, 24th
- Completing half this week
    - Should have expressions complete by start of next lab

1. Disambiguate Expression Grammar

- Use the operator associativity/precedence table to disambiguate all of the expression grammar on the assignment 2 description



2. Modify lexer.cpp

- Start from phase 1 solutions
  - Download solution.tar from camino (Project → 1)
- Edit **tokens.h** to include all tokens
  - All unique operators (e.g. +,-,/,%)
  - ID, num, string, done
  - All keywords (given)
- Modify **lexer.cpp**
  - Remove **main()**
  - Change **lexan()** to return an int (token)
  - Return appropriate token instead of printing them out
    i. Ex: return ASSIGN instead of printing “operator: =”

3. Writing the Parser

- Remember to import **lexer.h** and **tokens.h**
- Write your **main()** and **match()** functions (remember the global **lookahead**)
  a. Read lecture 4 slides for examples
- Write the code for expression:
  a. Start with Algebraic Binary (+, -, *, /, %)
  b. Then Prefix (!, &, -, *, sizeof)
  c. The rest of expression
- Remember to print out the output for each operator once the whole operation has been matched and completed (Phase 2 Assignment has the required output for each operator)
- Goal is to have expression written and thoroughly tested by the beginning of your next lab section.

Examples

```
[agigliot@linux10606 phase2]$ ./scc
a + b * d - c / d
mul
add
div
sub
[agigliot@linux10606 phase2]$
```

```
[agigliot@linux10606 phase2]$ ./scc
(a > b + c) * (1234 - sizeof(a[b]) || c && d)
add
gtn
index
sizeof
sub
and
or
mul
[agigliot@linux10606 phase2]$
```