Meeting 5: 1/27/2015: Parsing

HW1 Stats

- 11 hours mean, 10 hours median

HW1 Comments

- flatten/recursion hard+++++++++++++
- missing something with no lecture, someone else's question, what to ask+++ 
- Python, no static typing+
- architecture hard+
- getting started, what's needed+ -- example?
- suggestion: all open test cases for the first assignment
- love projects++ 
- like no templates+++ 
- like competition
- like compiler right now+ 
- register allocation hard+
- documentation to install on Mac
- like contributed test cases+ 
- like self paced, class as a hard distance student
- figuring out test cases+
- change nothing++++
- temporary variables+ 
- x86

```c
tmp0 = input()
tmp1 = -tmp0
tmp2 = - tmp1
tmp3 = - tmp2
print tmp3
```
HW2 Questions

* Lex/Yacc Algorithm
  → * Intro to PLY
  → Lexing Tokens

Parser

Po prog a seq of chars
  → [lexer] → seq of tokens
  → [parser] → AST

print

PRINT

lexar described by regular expression

\( r \) where \( r \) is a character
\( r_1 r_2 r \) if \( r_1, r, r_2 \) and \( r_2 r_1 \)
\( r, r_1 r_2 r \)
\( r, r_1 r_2 r \)
\( r, r_1 r_2 r \)
\( r, r_1 r_2 r \)
\( r, r_1 r_2 r \)

meaning is a language

a set of strings

\( \mathcal{L}(r) \) language of the regexp \( r \)

\[
\mathcal{L}(r_1 r_2) = \{ s_1 s_2 | \exists s \in \mathcal{L}(r_1) \text{ and } s_2 \in \mathcal{L}(r_2) \}
\]