Two issues:

1. Precedence of operators
2. Parentheses

Rules for binary operators:
- Same precedence: Grouping left to right.
- Higher precedence: Left to right.
- Lower precedence: Right to left.

Sample: $(1 + (2 * 3))$

Evaluating: $1 + (2 * 3) = 1 + 6 = 7$

Questions on Lab 3 [Bi-polar]

Midterm
- Tue: Quiz 3, Review
- Wed: 7pm ECCR 155: Review with Dana Shiao
- Thu: Midterm
- Fri: Party! (No Lab)

Lab 2 Grades:
- Mean: 85
- Median: 14, 34th A: 75
- 3rd Q: 94, 3rd A: 150

Midterm Walk-Next Week

Quiz 2 Grades:
- Mean: 13.5, 3rd A: 25
(2) Semantic issue - how to evaluate

evaluation order

\[ e_1 \# e_2 \]

evaluate \( e_1 \) first then \( e_2 \)
or \( e_2 \) first then \( e_1 \)
or concurrently?

eval order matter? if there are
side effects

\[ \text{print(1) \# print(2)} \]

\[ E_1 - c_1 \uparrow n_1, \quad E_1 + c_2 \uparrow n_2 \]

Evaluation order is unspecified here
\[ 3 + 3 \rightarrow 6 \]

\[ \frac{1 + 2}{3} + 3 \leq 6 \]

Should hold

Should hold

red

Expression e can take one-step reduction

Expression of evaluation to expression parameters

I change my judgment form so I think yes to the steps of evaluation explicit
\[ e' := n \mid e_1 + e_2 \]

\[ n' = n_1 + n_2 \]

\[ n_1 + n_2 \rightarrow n' \]

\[ \text{syntactic plus} \]

Give 2 rules to get L-to-R evaluation

Search Plus 1

\[ e_1 \rightarrow e_1' \]

\[ e_1 + e_2 \rightarrow e_1' + e_2 \]

Search Plus 2

\[ e_2 \rightarrow e_2' \]

\[ n_1 + e_2 \rightarrow n_1 + e_2' \]

Example with Plus

\[ 6 = 3 + 3 \]

\[ 3 + 3 \rightarrow 6 \]

**DoPlus**

\[ 1 + 2 \rightarrow 3 \]

**search plus**

\[ (1+2) + 3 \rightarrow 3+3 \]