The Rules

I. Atoms

- ■ **Self-Evaluating:** Numbers, Strings, #t, #f
- ■ **Symbols:** (aka variables) look for first binding

II. Lists (aka Compound Expressions)

Take the **car** of it, if it's a special form go to III.

Otherwise you're calling a procedure!

a. So evaluate **ALL** subexpressions by **The Rules** then...

b. If **car** is primitive → apply by magic

c. If **car** is a λ then…

   i. Create frame, f

   ii. Point f to where λ points

   iii. Bind formal parameters of λ in f & make f the current frame

   iv. Use **The Rules** to evaluate the body of λ in f

III. Special Forms

If it's a define or let…DESUGAR!

- □ Define
  1. Write variable name in current frame
  2. Evaluate body by **The Rules** in CF (current frame)
  3. Point 1 (variable name) → 2 (evaled body)
- □ Let
  - ■ ((lambda (args) body) vals) ← just evaluate by **The Rules**
- □ λ → procedure (λ (params) body)
  - ■ Draw Bubbles!
  - ■ Left Bubble points to parameters and body
  - ■ Right Bubble points to the CF where it's being evaluated
- □ Set!
  - ■ Evaluate the second argument by **The Rules**
  - ■ Find first binding of first argument
  - ■ Change the variable (first argument) to the evaled second argument
- □ Other Special forms follow their own rules of evaluation (cond, if, etc.)