CS3L:
Introduction to Symbolic Programming

Lecture 13:
Bugs and Two Stage Recursion

Summer 2008 Colleen Lewis
colleenL@berkeley.edu

Today

- Lab
  - Tail Recursion
  - Two-stage recursion
  - Bugs in recursion
- Homework
  - **Thurs:** Bowling (Hwk11) due Monday
  - **Fri:** Compress (Hwk12) occurs-in? (Hwk13) due Tuesday
  - **Mon:** Mini-project 2 due Thursday

Non-Tail Recursion

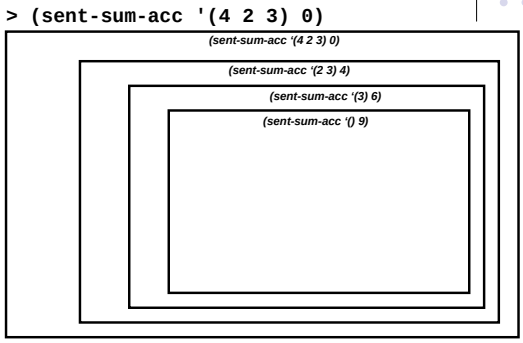
- Scratch Demo
  http://scratch.mit.edu/projects/ColleenBerkeley/213678

Two Stage Recursion

```
(define (sent-sum-acc sent sum-so-far)
  (if (empty? sent)
      sum-so-far
      (sent-sum-acc
                (bf sent)
                (+ (first sent) sum-so-far))))
```

Tail Recursion

```
> (sent-sum-acc '(4 2 3) 0)
```



Tail Recursion

- "Technically" the monkey doesn't have to stick around.
  - Computers are smart and try to make things faster
  - You don't have to worry about this in CS3 but it is an important thing in CS (like at my job at Leapfrog)

Is there a bug?

```
(define (no-pairs? sent)
  (cond
    ((empty? sent) #t)
    ((equal? (first sent)
             (first (bf sent))) #f)
    (else (no-pairs? (bf sent)))))

(no-pairs? '(a))
```

Number Spelling

- How do you spell 40? - FORTY
- No “and”s necessary! (or wanted)