CS3L:
Introduction to Symbolic Programming

Lecture 11:
Accumulating Recursion

Summer 2008

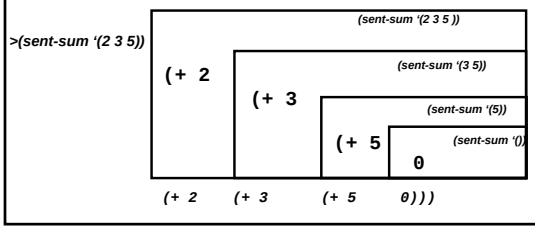
Colleen Lewis
colleenL@berkeley.edu

Today

- Upcoming Homework
  - **Thurs:** Bowling (Hwk11) due Monday
  - **Fri:** Compress/occurs-in? (Hwk12) due Tuesday
  - **Mon:** Mini-project 2 due Wednesday
- Lunch yesterday was fun ~10 people came
- Accumulating Recursion
- Recursion Patterns

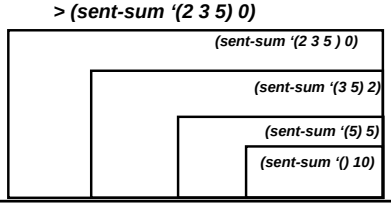
Not Accumulating Recursion

```
(define (sent-sum sent)
    (if (empty? sent)
        0
        (+ (first sent) (sent-sum (bf sent)))))
```



Accumulating Recursion

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



Recursion Patterns

- Split into groups of 4
  - I'll pass out a set of worksheets for each group
  - Work on the work sheet as a team
- Join another group and explain your worksheet to them

Application-To-All

```
(define (proc-applied-to-all sent)
  (if (empty? sent)
      '()
      (se (proc (first sent))
      (proc-applied-to-all (bf sent)))))
```

Examples of Application to All

- Square-all
- Add-1-to-all
- Add-2-to-all
- Grocery-to-cost (food->cost)
- Calorie-counter (food -> calories)
- French-to-english
- English-to-piglatin

Filtering

```
(define (filtered sent)
  (cond
      ((empty? sent) '())
      ((interesting? (first sent))
            (sent (first sent) (filtered (bf sent))))
      (else (filtered (bf sent)))))
```

Examples of Filtering

- Keep-Multiples-of-5
- Keep-Evens
- Keep-vowels
- Keep-words-with-even-num-letters
- Keep-Michael
- Gather-with-hair-color