

Extra Topics, Week 7

Notes by HF. Thanks HF!

Stuff on Quiz:

- Recursion
- Preconditions

Recursion

- solve problem by solving a simpler version of the problem and extending that solution (recursive case)
 - Lists: smaller version of the problem: one fewer element
- At some point, it's simple enough we can solve it directly (base case)
 - base case test: is it simple?
 - For lists, the base case test is Empty? One element?

Example: counting all #'s < 50 in a list

Input: list of real numbers

Output: integer

If the list is empty (base case test)

there are no number <50 (0) (base case computation)

Otherwise

count how many number < 50 appear in rest if first element <50 add 1 to number < 50

```
(define count-small
  (lambda (lst)
    (if (null? lst)
        0
        (if (< (car lst) 50)
            (+ 1 (count-small (cdr lst)))
            (count-small (cdr lst))))))
```

```
(count-small '(1 60 3 4 75 2))
```

extract all #'s < 50 in a list

Input: list of real numbers

Output: list of real #'s

Example: add all #'s < 50 in a list

Input: list of real numbers

Output: integer

If the list is empty (base case test)

there are no number <50 (0) (base case computation)

Otherwise

add number < 50 appear in rest if first element <50 add first element to number < 50

```
(define add-small
  (lambda (lst)
    (if (null? lst)
        0
        (if (< (car lst) 50)
            (+ (car lst) (add-small (cdr lst)))
            (add-small (cdr lst))))))
```

Example: Multiply #'s < 50

If the list is empty

the product is 1

Otherwise

compute product of remainder of list

if first is small, multiply first by remainder of list

o/w just use the product of the remainder of list

Multiply example with running product

(3 4 110 15 6 83)

update and remove to compute running product

Tail Recursion:

```
(define f
  (lambda (prod remaining)
    (if (null? remaining)
        prod
        (if (< (car remaining) 50)
            (f (* prod (car remaining))
                (cdr remaining))
            (f prod
                (cdr remaining))))))
```

Samuel A. Rebelsky, rebelsky@grinnell.edu

Copyright (c) 2007-2013 Janet Davis, Samuel A. Rebelsky, and Jerod Weinman. (Selected materials are copyright by John David Stone or Henry Walker and are used with permission.)



This work is licensed under a Creative Commons Attribution 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/3.0/> or send a letter to

Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.