
Lecture 34

CPSC 110

Peyton Seigo

2018-11-28

Lecture 34

Final Exam

- Final will be heavy on modules 10-12 and you need to be good at module 6 to do these.
- Hint: Final might have a HtDF problem at a level of complexity such that you NEED to write examples for accumulators. (multiple accumulators?)

Mutation

Template for for-each:

```
(@template for-each)
(define (fn lox)
  ;; acc: <Type>, <invariant>

  (local [(define acc...)]
    (begin (for-each ... lox) ; lambda expression or local fn
           (... acc))))
```

Summing elements in a list:

```
;; Pure Tail Recursion
(@template (listof Number) accumulator)
(define (sum lon0)
  ;; acc is Number: sum of numbers in lon0 seen so far

  (local [(define (sum lon acc)
              (cond [(empty? lon0) acc]
                    [else
                     (sum (rest lon0) (+ (first lon0)
                                           lon0))]))]
    (sum lon0 0)))

;; With for-each
(@template for-each)
(define (sum lon)
  ;; acc: Natural; sum of numbers in lon0 seen so far

  (local [(define acc 0)]
```

```
(begin (for-each (lambda (n)
                  (set! acc (+ n acc)))
              lon)
      acc)))
```