

ECE570 Lecture 3: Higher-Order Procedures

Jeffrey Mark Siskind

School of Electrical and Computer Engineering

Fall 2013



Tail Recursion

```
(define (sum list)
  (if (null? list)
      0
      (+ (first list) (sum (rest list)))))
```

```
(define (sum list)
  (define (loop list result)
    (if (null? list)
        result
        (loop (rest list) (+ result (first list)))))
  (loop list 0))
```

Recursion—I

$$n! = \prod_{i=1}^n i$$

```
(define (loop i n)
  (if (> i n)
      1
      (* i (loop (+ i 1) n))))

(define (product-from-one-to n) (loop 1 n))
```

Recursion—II

$$\sum_{i=1}^n i$$

```
(define (loop i n)
  (if (> i n)
      0
      (+ i (loop (+ i 1) n))))

(define (sum-from-one-to n) (loop 1 n))
```

Internal Definitions

```
(define (product-from-one-to n)
  (define (loop i n)
    (if (> i n)
        0
        (* i (loop (+ i 1) n))))
  (loop 1 n))
```

```
(define (sum-from-one-to n)
  (define (loop i)
    (if (> i n)
        0
        (+ i (loop (+ i 1)))))
  (loop 1))
```

Recursion—III

$$\sum_{i=m}^n i$$

```
(define (sum-from-to m n)
  (define (loop i)
    (if (> i n)
        0
        (+ i (loop (+ i 1)))))
  (loop m))
```

Recursion—IV

$$\sum_{i=m}^n i^2$$

```
(define (sum-sqr-from-to m n)
  (define (loop i)
    (if (> i n)
        0
        (+ (sqr i) (loop (+ i 1)))))
  (loop m))
```

$$\sum_{i=m}^n \sqrt{i}$$

```
(define (sum-sqrt-from-to m n)
  (define (loop i)
    (if (> i n)
        0
        (+ (sqrt i) (loop (+ i 1)))))
  (loop m))
```


Higher-Order Procedures

$$\sum_{i=m}^n f(i)$$

```
(define (sum-from-to f m n)
  (define (loop i)
    (if (> i n)
        0
        (+ (f i) (loop (+ i 1)))))
  (loop m))
```

```
(define (sum-sqr-from-to m n)
  (sum-from-to sqr m n))
```

```
(define (sum-sqrt-from-to m n)
  (sum-from-to sqrt m n))
```

Lambda Expressions (Anonymous Procedures)

$$\sum_{i=m}^n \sin^2 i + \cos^2 i$$

```
(define (sin-sqr+cos-sqr x)
  (+ (sqr (sin x)) (sqr (cos x))))
```

```
(define (sum-sin-sqr+cos-sqr-from-to m n)
  (sum-from-to sin-sqr+cos-sqr m n))
```

```
(define (sum-sin-sqr+cos-sqr-from-to m n)
  (sum-from-to
    (lambda (x) (+ (sqr (sin x)) (sqr (cos x)))))
    m
    n))
```

Reduction—I

$$\sum_{i=m}^n f(i) \qquad \prod_{i=m}^n f(i)$$

```
(define (sum-from-to f m n)
  (define (loop i)
    (if (> i n)
        0
        (+ (f i) (loop (+ i 1)))))
  (loop m))

(define (product-from-to f m n)
  (define (loop i)
    (if (> i n)
        1
        (* (f i) (loop (+ i 1)))))
  (loop m))
```

Reduction—II

```
(define (reduce-from-to g identity f m n)
  (define (loop i)
    (if (> i n)
        identity
        (g (f i) (loop (+ i 1)))))
  (loop m))
```

```
(define (sum-from-to f m n)
  (reduce-from-to + 0 f m n))
```

```
(define (product-from-to f m n)
  (reduce-from-to * 1 f m n))
```

```
(define (max x y) (if (> x y) x y))
(define (max-from-to f m n)
  (reduce-from-to max *minus-infinity* f m n))
```