Functional Programming

(Lists-More)

Professor: Suman Saha

Continue: List Example

PennState

- Define subs, which takes a b l, and replace a with b in l

```
(define (subs a b l)
    (if (null? l) '()
        (let ((rest (subs a b (cdr l))))
            (if (equal? (car l) a)
                (cons  b       rest)
                (cons (car l) rest)))))
```

Examples w/o Higher-Order Functions

PennState

- Compute the sum of a list

```
init = 0
foreach (elem in lst)
   init += elem
```

- Computer the product of a list

```
init = 1
foreach (elem in lst)
  init *= elem
```

Foldl

- foldl f a (e_1 e_2 e_n) : returns
- f e_n(… (f e_2 (f e_1 a)) ..)

```
(foldl + 0 '(1 2 3 4 5))
```

```
(foldl * 1 '(1 2 3 4 5))
```

Foldr

- foldr f a (e_1 e_2 e_n) : returns
- f e_1(… (f e_{n-1} (f e_n a)) ..)

```
(foldr cons `() `(1 2 3))
(foldl cons `() `(1 2 3))
```

Foldl and Foldr

```
(define (add x y) (+ x y))
```

```
(foldl add 0 (list 1 2 3))
(foldl add (+ 1 0) (list 2 3))
(foldl add (+ 2 1) (list 3))
(foldl add (+ 3 3) (list ))
```

```
(foldr add 0 (list 1 2 3))
(foldr add (+ 3 0) (list 2 3))
(foldr add (+ 2 3) (list 3))
(foldr add (+ 1 5) (list ))
```