Built-In Abstract Functions

Natural. (Natural -> X) -> (listof X)

produces (list (f 0) ... (f (- n 1)))

(define (build-list n f) ...)

(X -> boolean). (listof X) -> (listof X)

produce a list from all those items on lox for which p holds

(define (filter p lox) ...)

(X -> Y) (listof X). -> (listof Y)

produce a list by applying f to each item on lox

that is, (map f (list x-1 ... x-n)) = (list (f x-1) ... (f x-n))

(define (map f lox) ...)

(X -> boolean) (listof X) -> Boolean

produce true if p produces true for every element of lox

(define (andmap p lox) ...)

(X -> boolean) (listof X) -> Boolean

produce true if p produces true for some element of lox

(define (ormap p lox) ...)

(X Y -> Y) Y (listof X) -> Y

(foldr f base (list x-1 ... x-n)) = (f x-1 ... (f x-n base))

(define (foldr f base lox) ...)

(X Y -> Y) Y (listof X) -> Y

(foldl f base (list x-1 ... x-n)) = (f x-n ... (f x-1 base))

(define (foldl f base lox) ...)