14. Functional Programming
: 함수 호출에 기반을 둔 programming 기법

14.1 명령형 언어의 특징

p419
cf) imperative programming language (명령형 PL)
o 변수
o assignment statement
o 반복

p421 명령형 언어의 문제점

14.2 Functional PL의 본질
reading
(표 14.1)
Lamda Calculus
square(x) = x * x
⇒ λx.x*x
(λx.x*x)2 = 2*2 = 4

14.2.3 p428
Functional Language = Applicative Language
구성요소
① a set of primitive functions
② 함수합성기법
③ applicative operation
④ set of data objects

* LISP p435
① object : atom, list
문자열 (a people)
a
people (atom list)
( ) ≡ NIL
② function
o QUOTE : 인수 자체 반환
인수를 상수 취급
(QUOTE(A)) ≡ A
(QUOTE(A B C)) ≡ (A B C)
o CAR : list 첫 원소 반환

o CDR : 첫 원소를 제외한 나머지 list 반환

o CONS : 두 번째 parameter에 첫 번째 계산결과 부가

(CAR(QUOTE(A B C))) ≡ A

(CDR(QUOTE(A B C))) ≡ (B C)

(CONS(QUOTE A)(QUOTE B C)) ≡ (A B C)

o λx,y.x+y

(LAMBDA(x y)(plus x y))

((LAMBDA(x y)(plus x y)) 2 3)

(define(ADD(LAMBDA(x y)(plus x y))))

(ADD 2 3)

p439 14.5 applicative PL 과 imperative PL의 비교