

Challenge
SNU 4910.210, Fall 2015
Chung-Kil Hur
due: 12/6(Sun) 23:59

Challenge “Scheme 인터프리터”

Scheme expression을 받아서 해당하는 값을 계산하는 인터프리터 함수 `myeval`을 직접 구현하라.

$\text{myeval} : E \rightarrow V$

- 입력은 아래에 주어지는 규칙에 따라 조합된 E 의 앞에 `quote(')`가 붙어 들어온다. ex) `'(cons 3 5)`
- 출력은 숫자, boolean, pair의 경우 Scheme에서 제공하는 값을 사용하며, lambda 값의 경우는 각자가 임의로 정의한다.
- 입력이 적절하게 처리될 수 없는 경우 예외처리를 통해 실행을 중단하고 알맞은 메시지를 출력해야 한다.
- Scheme에서 제공하는 함수 `eval`은 사용할 수 없다.

$E ::= C$	상수(숫자, #t, #f, '())
x	변수
$(\text{if } E E E)$	conditional
$(\text{cons } E E)$	pair
$(\text{car } E)$	selection
$(\text{cdr } E)$	selection
$(\text{lambda } (x^*) E)$	function
$(E E^*)$	application
$(\text{let } ((x E)^+) E)$	let
$(\text{letrec } ((x E)^+) E)$	letrec
$(+ E E)$	addition
$(- E E)$	subtraction
$(* E E)$	multiplication
$(= E E)$	equality
$(< E E)$	less than
$(> E E)$	greater than

실행 예제

- `(myeval '(let ((p (cons 1 (cons 2 '())))) (cons 0 p)))`
결과: `'(0 1 2)`
- `(myeval '(letrec ((f (lambda (x) (if (= x 0) 0 (+ x (f (- x 1)))))) (f 5)))`
결과: `15`
- `(myeval '((lambda (f) (lambda (x) (f x))) (lambda (x) (+ x 1))))`
결과: `'((lmda (x) (f x)) env ...)` (각자 임의로 정의 가능)