

상명대학교 컴퓨터과학부 "EA0011: 프로그래밍 언어론" 숙제 5

제출일: 2017년 11월 7일 수업 전

아래 표 1에 정의된 syntax를 가지는 언어 EXP에 대하여 표 2의 프로그램 설명을 만족하는 Scheme 프로그램을 파일 "~/pl/hw05/parse.scm"에 작성하라. 프로그램 작성 시 template 파일인 "/home/pl/hw05/parse.scm"을 복사하여 사용한다. 작성된 프로그램은 표 3과 같이 테스트 프로그램 "/home/pl/hw05/test.scm"를 수행하여 ok 통과하여야 한다. 숙제 제출물은 ① 연필로 쓴 프로그램 코드 및 설명 (4점), ② 테스트 프로그램 "/home/pl/hw05/test.scm"을 수행시킨 후 출력된 ok 갯수 쓰고 (1점), 테스트 결과 화면 출력 첨부하기 (6점: ok 1개당 1점), 및 ③ 프로그래밍 서버 acacia.smu.ac.kr에 작성한 프로그램 파일 "~/pl/hw05/parse.scm"이다.

표 1 언어 EXP의 syntax

Concrete syntax	Abstract syntax
Exp ::= Var	var-exp (var)
Exp ::= (lambda (Var) Exp)	abs-exp (var exp1)
Exp ::= [Exp Exp]	app-exp (exp1 exp2)

표 2 작성할 프로그램 설명

프로그램 설명	프로그램 모습
언어 EXP의 abstract syntax를 표현하는 datatype exp의 정의.	(define-datatype exp ...)
언어 EXP의 concrete syntax로 표현된 프로그램을 abstract syntax로 표현된 프로그램으로 변환하는 SLLGEN parser의 scanner rule 및 parser rule의 정의.	(define scanner-rules ...) (define parser-rules ...)
언어 EXP의 abstract syntax로 표현된 프로그램을 concrete syntax로 표현된 string으로 변환하는 함수 exp->string의 정의.	(define exp->str ...)

표 3 테스트 프로그램 수행

<pre>\$ mzscheme -M eopl -f parse.scm -f /home/pl/hw05/test.scm Welcome to MzScheme v372 [3m], Copyright (c) 2004-2007 PLT Scheme Inc. > (test) ===== exp1 ===== concrete: "x" abstract: #(struct:var-exp x) string: "x" ok ===== exp2 ===== concrete: "(lambda (x) x)" abstract: #(struct:abs-exp x #(struct:var-exp x)) string: "(lambda (x) x)" ok ===== exp3 ===== concrete: "[f x]"</pre>

```

abstract: #(struct:app-exp
  #(struct:var-exp
    f)
  #(struct:var-exp
    x))
string: "[f x]"
ok
===== exp4 =====
concrete: "[(lambda (x) x) y]"
abstract: #(struct:app-exp
  #(struct:abs-exp
    x
    #(struct:var-exp
      x))
  #(struct:var-exp
    y))
string: "[(lambda (x) x) y]"
ok
===== exp5 =====
concrete: "(lambda (f) (lambda (x) [f x]))"
abstract: #(struct:abs-exp
  f
  #(struct:abs-exp
    x
    #(struct:app-exp
      #(struct:var-exp
        f)
      #(struct:var-exp
        x))))
string: "(lambda (f) (lambda (x) [f x]))"
ok
===== exp6 =====
concrete: "[[(lambda (f) (lambda (x) [f x])) g] y]"
abstract: #(struct:app-exp
  #(struct:app-exp
    #(struct:abs-exp
      f
      #(struct:abs-exp
        x
        #(struct:app-exp
          #(struct:var-exp
            f)
          #(struct:var-exp
            x))))
    #(struct:var-exp
      g))
  #(struct:var-exp
    y))
string: "[[(lambda (f) (lambda (x) [f x])) g] y]"
ok
> (exit)
$

```

끝.