

Homework 3

SNU 4190.310, 2026 봄

이 광근

Due: 4/6, 24:00

이번 숙제의 목적은:

- 수업시간에 살펴본, 꼬마 명령형 언어(K--)의 정확한 정의를 이해하기.
- 그 실행기(interpreter)를 구현해 보기.
- 조교가 제공하는 실행기 틀과 친숙해지기.

Exercise 1 (20pts) “K-- 실행기”

꼬마 명령형 언어 K--의 문법과 의미의 정확한 정의는 강의페이지 [숙제]란에 제공된다. 프로시저와 레코드는 아직 없는 언어다. 이번 숙제는 K-- 프로그램을 의미정의대로 실행시키는 함수(interpreter)를 작성하는 것이다.

아래의 KMINUS 꼴을 가지는 모듈 K를 정의하라.

```
module type KMINUS =
sig
  exception Error of string
  type id = string
  type exp = NUM of int | TRUE | FALSE
           | VAR of id
           | ADD of exp * exp
```

```

| SUB of exp * exp
| MUL of exp * exp
| DIV of exp * exp
| EQUAL of exp * exp
| LESS of exp * exp
| NOT of exp
| SEQ of exp * exp          (* sequence *)
| IF of exp * exp * exp    (* if-then-else *)
| WHILE of exp * exp      (* while loop *)
| LETV of id * exp * exp   (* variable binding *)
| ASSIGN of id * exp      (* assign to variable *)
| READ of id
| WRITE of exp

type program = exp
type memory
type env
type value
val emptyMemory: memory
val emptyEnv: env
val run: memory * env * program -> value
end

```

K-- 프로그램이 어떻게 *exp*들로 표현될지는 쉽게 추측할 수 있을 것입니다. *exp*으로 표현된 K-- 프로그램이 *S*라고 하면,

$$K.run (K.emptyMemory, K.emptyEnv, S)$$

는 프로그램 *S*를 실행시키게 되는데, 성공적으로 끝나면 최후의 값을 내어주게 됩니다. 이때 프로그램은 실행중에 I/O를 하면서 프로그램이 하는 일을 바깥세상에 드러내게 됩니다. 실행중에 타입이 맞지 않는 프로그램이면 **Error**라는 예외상황을 발생시키고 프로그램 실행이 중단되어야 합니다. “Error”란 (\Leftarrow) 정의된 의미 규칙으로는 그 프로그램의 의미가 정의될 수 없는 경우입니다. 입출력은 정수만

가능합니다. 출력은 정수를 화면에 뿌리고 다음 줄로 넘어갑니다. □