

Homework 7

SNU 4190.210 Fall 2014

Chung-Kil Hur

due: 12/03 (Wed) 24:00

이번 숙제의 목적은 예전의 숙제를 다시 구현하면서

- 타입으로 프로그램을 정리하는(typeful programming) 연습하기.
- 자동으로 타입 검증을 해주는 환경에서 프로그래밍 연습하기.
- 모듈(module) 프로그래밍 기본 연습하기.
- 수업시간에 배운 “inductive refinement”를 연습해 본다. 수업시간에 다른 문제에서 한 가지가 추가된 경우이다.

Exercise 1 “튜링기계_{m1}”

지난 HW5에서 Scheme으로 제작한 튜링기계를 OCaml로 구현해 봅시다. 적어도 아래의 모듈타입 TM만큼은 정의된 모듈 TuringMachine를 제작해서 제출합니다. 함수들이 하는 일은 HW5와 같습니다.

```
module type TM =
  sig
    type symbol = string
    type move = Right | Left | Stay
    type todo = Erase | Write of symbol
    type state = string
    type rule = state * symbol * todo * move * state
    type ruletable = rule list
    type tape
    type tm
```

```

(* tape part *)
val init_tape: symbol list -> tape
val read_tape: tape -> symbol
val write_tape: tape -> symbol -> tape
val move_tape_left: tape -> tape
val move_tape_right: tape -> tape
val print_tape: tape -> unit

(* rule table part *)
val match_rule: state -> symbol -> ruletable -> todo * move * state

(* main *)
val make_tm: symbol list -> state list -> state -> ruletable -> tm
val run_tm: tm -> tm
val print_tm: tm -> int -> unit

end
□

```

Exercise 2 “SKI 용액 반응기_{ml}”

HW4의 “SKI” 용액 반응기를 Ocaml로 구현해 봅시다. 적어도 다음의 모듈 타입 SKI만큼은 정의된 모듈 SkiLiquid을 제작해서 제출합니다.

```

module type SKI =
  sig
    type liquid = S
                | K
                | I
                | V of string          (* variable *)
                | M of liquid * liquid (* mix of two liquids *)

    val react: liquid -> liquid
    val pprint: liquid -> unit
  end
□

```

Challenge 3 “어울리지 않아”

스트링은 빈 스트링을 포함해서 0, 1, 2로 구성된다: 예) 0000, 1102201, 2010110012 등. “스트링 s 가 코드 c 와 어울린다”는 것은 코드 c 가 표현하는 스

트링 집합에 s 가 포함된다는 뜻이다. 코드 c 는 다음과 같이 정의되고:

$$c \rightarrow 0 \mid 1 \mid 2 \mid c \cdot c \mid c+c \mid c? \mid c^*$$

코드 c 가 뜻하는 스트링의 집합 $\llbracket c \rrbracket$ 는 다음과 같이 정의된다. 빈 스트링은 ϵ 로 표현한다.

$$\begin{aligned} \llbracket 0 \rrbracket &= \{0\} \\ \llbracket 1 \rrbracket &= \{1\} \\ \llbracket 2 \rrbracket &= \{2\} \\ \llbracket c_1 \cdot c_2 \rrbracket &= \{s_1 s_2 \mid s_1 \in \llbracket c_1 \rrbracket, s_2 \in \llbracket c_2 \rrbracket\} \\ \llbracket c_1+c_2 \rrbracket &= \llbracket c_1 \rrbracket \cup \llbracket c_2 \rrbracket \\ \llbracket c? \rrbracket &= \{\epsilon\} \cup \llbracket c \rrbracket \\ \llbracket c^* \rrbracket &= \{\epsilon\} \cup \llbracket c \rrbracket \cup \llbracket c \cdot c \rrbracket \cup \llbracket c \cdot c \cdot c \rrbracket \cup \dots \end{aligned}$$

주어진 스트링 s 가 코드 c 와 어울리는 지를 결정하는 함수 `smatch`를 정의합시다. 예를 들어,

```
smatch "11" 1·0*·1?
```

은 참을,

```
smatch "11" (10)*·1
```

은 거짓을 내야 한다.

적어도 다음의 모듈타입 `SMATCH` 만큼 정의된 모듈 `Smatch`를 제작해서 제출합니다.

```
module type SMATCH =
  sig
    type c = Zero | One | Two
      | Mult of c * c
      | Sum of c * c
      | Opt of c           (* c? *)
      | Star of c          (* c* *)
    val smatch: string -> c -> bool
  end
```

□