

Homework 2
SNU 4190.310, Fall 2010
Kwangkeun Yi
due: 9/26(Sun), 24:00

Exercise 1 “이쁘개”

Homework 1의 “대진표 스트링” 문제와 연관된다. 다음의 함수, `pptree`를 작성하라:

```
pptree: tourna -> unit
```

`pptree`는 `tourna`를 받아서 그 트리를 “아름답게” 화면에 그려주는 프로그램이다. 다음의 “아름다운 이진트리”의 조건을 만족하도록 한다.

- 잎새노드들은 모두 맨 밑에 있다.
- 모든 가지는 직각의 꼴이고, “|” 와 “-”만을 사용한다
- 모든 subtree의 root는 “|”이고, 그 가지의 중앙에 있다.

단, 잎새노드에서 노드값(team)은 print하지 않는다. 예를 들어 3 팀의 대진표 트리의 `pptree`결과는 아래와 같이 출력된다:

```
  |
  |---|
 |-|  |
```

□

Exercise 2 “계산기 `mathemadiga`”

다음의 계산기

```
mathemadiga: exp -> float
```

를 만듭시다.

```

type exp = X
  | INT of int
  | REAL of float
  | ADD of exp * exp
  | SUB of exp * exp
  | MUL of exp * exp
  | DIV of exp * exp
  | SIGMA of exp * exp * exp
  | INTEGRAL of exp * exp * exp

```

예를들어 우리가 쓰는 수식이 exp타입으로는 다음과 같이 표현된다:

$$\sum_{x=1}^{10} (x * x - 1) \quad \text{SIGMA}(\text{INT } 1, \text{INT } 10, \text{SUB}(\text{MUL}(X, X), \text{INT } 1))$$

$$\int_{x=1.0}^{10.0} (x * x - 1) dx \quad \text{INTEGRAL}(\text{REAL } 1.0, \text{REAL } 10.0, \text{SUB}(\text{MUL}(X, X), \text{INT } 1))$$

적분식을 계산할때의 알갱이 크기(dx)는 0.1로 정한다.

□

Exercise 3 “계산해줘!”

정수식들의 구조를 다음의 타입으로 정의했습니다:

```

type expr = NUM of int
  | PLUS of expr * expr
  | MINUS of expr * expr
  | MULT of expr * expr
  | DIVIDE of expr * expr
  | MAX of expr list

```

주어진 expr를 받아서 정수값을 만들어내는 함수 eval

```
eval: expr -> int
```

를 정의하세요. 이때, MAX [NUM 1, NUM 3, NUM 2] = 3, 즉 MAX는 정수식 리스트에서 가장 큰 정수를 찾아내는 정수식입니다. 빈 리스트의 경우는 0을 의미하는 정수식 입니다. □

Exercise 4 “Queue = 2 Stacks”

큐는 반드시 하나의 리스트일 필요는 없습니다. 두개의 스택으로 큐를 효율적으로 구현할 수 있습니다. 큐에 넣고 빼는 작업이 거의 한 스택에 이루어

질 수 있습니다. (하나의 리스트위를 더듬는 두 개의 포인터를 다루었던 C의 구현과 장단점을 비교해 보세요.)

각각의 큐 연산들의 타입들은:

```
emptyQ: queue
enQ: queue * element -> queue
deQ: queue -> element * queue
```

큐를 $[a_1; \dots; a_m; b_1; \dots; b_n]$ 라고 합시다 (b_n 이 머리). 이 큐를 두개의 리스트 L 과 R 로 표현할 수 있습니다:

$$L = [a_1; \dots; a_m], \quad R = [b_n; \dots; b_1].$$

한 원소 x 를 삼키면 새로운 큐는 다음이 됩니다:

$$[x; a_1; \dots; a_m], [b_n; \dots; b_1].$$

원소를 하나 빼고나면 새로운 큐는 다음이 됩니다:

$$[a_1; \dots; a_m], [b_{n-1}; \dots; b_1].$$

빨 때, 때때로 L 리스트를 뒤집어서 R 로 გადა 놔야하겠습니다. 빈 큐는 $([], [])$ 이겠지요.

다음과 같은 Queue 타입의 모듈을 작성합니다:

```
module type Queue =
  sig
    type element
    type queue
    exception EMPTY_Q
    val emptyQ: queue
    val enQ: queue * element -> queue
    val deQ: queue -> element * queue
  end
```

다양한 큐 모듈이 위의 Queue 타입을 만족시킬 수 있습니다. 예를들어:

```
module IntListQ =
  struct
    type element = int list
    type queue = ...
```

```
exception EMPTY_Q
let emptyQ = ...
let enQ = ...
let deQ = ...
end
```

는 정수 리스트를 큐의 원소로 가지는 경우겠지요. 위의 모듈에서 함수 `enQ`와 `deQ`를 정의하기 바랍니다.

이 모듈에 있는 함수들을 이용해서 큐를 만드는 과정의 예는:

```
let myQ = IntListQ.emptyQ
let yourQ = IntListQ.enQ(myQ, [1])
let (x,restQ) = IntListQ.deQ yourQ
let hisQ = IntListQ.enQ(myQ, [2])
```

□