Homework 1
CS420/KAIST Spring 1999
이 광근
Due: March 18, 24:00

1999년 3월 11일

Exercise 1 우리가 중고등 수학시간에 숱하게 썼던 다음의 "씨그마"를 ML로 정의하라:

$$\Sigma_{n=a}^{b} f(n)$$

씨그마의 타입은

```
sigma :  int * int * (int -> int) -> int.
```

□

Exercise 2 위의 "씨그마" 함수에서 여러분이 사용한 모든 상수들을 (+, 0, 1, >) 인자로 하는 "왕씨그마"(일반화된 씨그마) 함수를 정의하라. □

Exercise 3 위의 "왕씨그마"를 이용해서 적분값을 임의의 정확도를 가지고 (주어진 뜸뜸이로) 계산해내는 함수 "인테그랄"을 정의하라:

$$\int_a^b f(x)dx$$

인테그랄의 타입은

```
integral :  real * real * (real -> real) * real -> real.
```

□

Exercise 4 114 전화 시스템에서는 찾은 전화번호를 음성합성으로 알려주는데, 이 시스템의 핵심함수, `vocalize`를 작성하라:

```
vocalize:  string -> string list list
```

일곱자리 양의 정수 string을 받아서 세자리와 네자리로 나누어서 읽어주도록 하는 함수이다. 예를 들어,

```
vocalize "8693536"
= [["팔","백","육","십","구"],["삼","천","오","백","삼","십","육"]]
```

단, 0을 읽지않아도 될 경우 읽지않는다. 예를들어, 0022900은

```
[["이"], ["이", "천", "구", "백"]]
```

□

Exercise 5 임의의 나무를 여러분 바지의 "지퍼"로 구현할 수 있답니다.

- 나무구조 타입은 아래와 같이 정의되겠지요:
  ```
  datatype tree = LEAF of item
                | NODE of tree list
  ```
- 아래의 zipper가 나무의 줄기를 타고 자유자재로 찢어놓기도 하고 붙여놓기도 합니다.
  ```
  datatype zipper = TOP
                  | HAND of tree list * zipper * tree list
  ```
  현재 나무줄기의 어느지점에 멈춰 있는 지퍼손잡이 HAND(l,z.r)에서, l은 왼편에 찢어져서 매달린 나무들(elder siblings)이고 r은 오른편에 찢어져서 매달린 나무들(younger siblings)이다.
- 나뭇가지에서의 현재 위치 location는 현재위치를 뿌리로하는 나무자체와 지퍼(zipper)로 표현되는 주변 나무들로 구성된다.
  ```
  datatype location = LOC of tree * zipper
  ```
- 예를들어, "$a \times b + c \times d$" 가 다음과 같은 나무구조로 표현되는데:
  ```
  NODE [ NODE [LEAF a, LEAF *, LEAF b],
         LEAF +,
         NODE [LEAF c, LEAF *, LEAF d]
       ]
  ```
  두번째 곱셈표에의 위치는 다음과 같다:
  ```
  LOC (LEAF *,
       HAND([LEAF c],
            HAND([LEAF +, NODE[LEAF a, LEAF *, LEAF b]],
                 TOP,
                 []),
            [LEAF d]))
  ```
- 자, 주어진 위치에서 이제 자유자재로 나무를 탈 수 있습니다. 왼편으로 옮겨가는 것은 다음과 같지요:
  ```
  fun goLeft (LOC(t, TOP)) = raise NOMOVE ‘‘left of top’’
    | goLeft (LOC(t, HAND(l::left,up,right) = LOC(l,HAND(left,up,t::right)
    | goLeft (LOC(t, HAND([],up,right) = raise NOMOVE ‘‘left of first’’
  ```
- 다음의 나머지 함수들을 정의하세요:
  ```
  goRight:  location -> location
  goUp:  location -> location
  goDown:  location -> location
  ```

□