

PL Practice Report



[04] map function

2021/03/29

201702083 최 현 석

주어진 문제

- List 와 tuple 을 활용한 Map 을 구현하시오. (tuple = string * string)
 - Add key value map : (key, value) 쌍을 map 에 추가한다.
 - 이미 존재하는 key 라면 value 를 업데이트
 - Find key map: map 에서 key 를 찾아 value 값을 반환, 없다면 에러메세지 출력
 - Erase key map : map 에서 key 를 찾아 해당 tuple 을 삭제, 없다면 에러메세지 출력

해결 과정 – add key value map

1. 내부에 rec add_rec 를 선언하여 map 을 mh 와 ml 로 나눈다. mh 는 []로 시작

```
let add key value map =  
  let rec add_rec key value mh ml =  
    match ml with
```

2. 그리고 ml 에 대해 패턴 매칭을 한다.

```
| a :: tl ->  
  if ( (fst a) = key ) then ( (key,value) :: tl )  
  else (add_rec key value (a::mh) tl)  
| [] -> (key,value) :: map  
in
```

- 원소가 존재 -> 해당 원소의 key 가 입력받은 key 와 같다면 taillist 인 tl 에 추가한다.
 - 같지 않다면 (a::mh) tl 로 mh 와 ml 를 넣어 재귀 호출한다.
- Empty -> map 에 바로 추가해준다.

3. Lrec add_rec 의 body 가 in 으로 끝나면 add_rec key value [] map 을 호출한다.

```
in  
  add_rec key value [] map
```

해결 과정 – find key map

```
let rec find key map =  
  match map with  
  | a :: tl -> if (fst a) = key then (snd a) else find key tl  
  | [] -> failwith " No such key exists "
```

- map 에 대해 패턴 매칭을 한다.
 - 원소 존재 -> 원소의 key 가 일치한다면 value 를 반환 / 아니라면 key tl 에 대해 재귀 호출
 - Empty -> “No such key exists” 메시지 출력

해결 과정 – erase key map

```
let erase key map =  
  let rec erase_rec key mh ml =  
    match ml with  
    | a :: tl -> if (fst a) = key then mh@tl else (erase_rec key (a::mh) tl)  
    | [] -> failwith " No such key exists "  
  in  
    erase_rec key [] map
```

1. Add 함수와 마찬가지로 map 을 mh 과 ml 로 나누어 내부에 rec 함수를 선언한다.
2. ml 에 대해 패턴 매칭을 한다.
 - 원소 존재 -> 원소의 key 가 같다면 원소를 제외한 mh 와 tl 을 @ 연산을 통해 합친다.
아니라면 a::mh tl 에 대해 재귀 호출한다.
 - Empty -> “No such key exists” 메시지를 출력
3. Let - in 이 끝나면 mh 를 비어있는 리스트 [] 로 넣고 호출한다.
 - ↳ erase_rec key [] map

결과(test case)

```
let print_map map =  
  let _ = Format.printf " [ " in  
  let rec print map =  
    match map with  
    | [] -> Format.printf " ]\n"  
    | a :: tl -> match tl with  
      | [] -> Format.printf "( %s %s )" (fst a) (snd a); print tl  
      | _ :: _ -> Format.printf "( %s %s ); " (fst a) (snd a); print tl  
  in  
  print map
```

- (String*string) list 출력하는 함수를 만들어 test case 출력

```
let _ = print_map map2 in  
let map2' = ListMap.erase "n"  
let _ = print_map map2' in  
let map2'' = ListMap.erase "  
let _ = print_map map2'' in  
Format.printf "%s\n"
```

```
macbook@MacDevCHS hw (main) $ ./_build/default/map.exe  
name : kihoon  
city : Daejeon  
age : 23  
[ ( city Daejeon ); ( name kihoon ) ]  
[ ( age 23 ); ( city Daejeon ) ]  
[ ( age 23 ); ( name kihoon ) ]
```

느낀 점

- add 와 erase 함수를 map 을 그대로 사용해서 재귀 호출하여 `a :: (add key value tl)` 과 같이 non-tail-rec 하게 만들었는데 tail-rec 하게 구현하는 것이 좋은 방법이라 하여 방법을 찾던 중 `a :: <-` 이 부분을 함수의 매개변수로 넣으면 되겠다는 생각을 했다. 그래서 내부에 새로운 함수를 선언하여 tail-rec 하게 구현했다. 앞으로 non-tail-rec 한 코드를 tail-rec 한 코드로 변환하고 싶을 때 새로운 매개변수를 갖는 함수를 만드는 게 좋은 방법이 될 것 같다.
- 과제 해결 시간 : 실습 시간 내
- 수강생들이 출력 부분을 어려워해서 Test case 와 함께 결과를 출력해주는 함수가 포함되면 좋을 것 같다.