

Homework 4

SNU 4190.310, 2018 가을

이 광근

Essay Due: 10/30(화), 14:00

Program Due: 10/26(금) 24:00

이번 숙제의 목적은:

- 상식적인 수준에서 디자인된 명령형 언어의 대표적인 C언어의 역사와, 컴퓨터 실행(기계적인 계산)과 상위 논리의 관계, 프론티어들의 프로그래밍 언어 사용경향, 튜링의 오리지널 논문 이야기등을 읽고 느낀 바를 글로 쓰기.
- 앞으로 프로그래밍 언어 구현에서 넘어야 할 산이, 상식만으로는 넘기 어렵다는 것을 겪어보기.

다음 두 개의 프로그래밍 문제중에 하나만 도전해서 겪어보세요. 채점은 맞고 틀리고가 아니라 어디까지 했나로 결정합니다. 조교의 기준을 참고하세요.

**Exercise 1** (50pts) “논술 에세이”

강의 홈페이지의 읽을거리 [Part I]에 매달린 글들을 읽고 리포트로 작성해서 제출합니다.

- 양: A4용지 총 2쪽 이내.
- 형식: 제목/학번/이름/본문으로 구성합니다. 본문은 두괄식으로 씁니다. 두괄식이란, 각 단락의 요점을 그 단락의 첫 문장에 놓는 것을 말합니다. 단락 내용을 정리한 문장(topic sentence)이 단락의 첫 문장입니다.
- 내용: 3개 섹션으로 구성해서 작성합니다. [알게 된 것](5점), [느낀 것](30점), 그리고 [질문하고 싶은 것](15점).

- 반드시 컴퓨터로 양면 출력해서 제출.
- 10/30(화) 수업시간에 제출. No delay acceptable.

□

## Exercise 2 (30pts) “탐사 준비”

탐사해야 할 지역의 지도를 보고 탐사를 성공리에 마치기 위해 필요한 최소의 준비물을 알아내는 프로그램을 작성해 보자.

탐사는 지도에 나타난 길을 따라 이동하면서 길에 놓인 보물상자를 열고 보물을 모아가는 것이고, 모든 보물이 모아지면 그 탐사는 성공한 것이다. 준비물은 모든 보물상자를 열 수 있는 열쇠들이다.

### 보물상자와 열쇠:

- 보물상자에는 고유의 알파벳 이름이 표시되어 있다.
- 이름없이 “\*”라고 찍혀있는 보물상자도 있다.
- 같은 이름의 보물 상자는 같은 열쇠로 열린다.
- 하나의 열쇠는 외갈래 혹은 두갈래로 갈라진 가지구조(tree)이다.
- 열쇠는 반복해서 사용할 수 있다.

보물상자와 열쇠를 OCaml 타입으로 정의하면,

```
type treasure = StarBox | NameBox of string
type key = Bar | Node of key * key
```

### 탐사지도:

- 시작 지점은 하나이다.
- 길들은 모두 외갈래이거나 두 갈래로 나뉘어 진다.
- 보물상자들은 모두 막다른 골목의 끝에 있다.
- 갔던 길을 되돌아 오지 않고 왔던 곳으로 다시 오는 방법은 없다(tree).
- 길목에 세워진 안내판에는 앞으로 만날 보물상자의 알파벳 이름이 쓰여져 있다.
- 모든 안내판의 이름은 모두 다르다.

탐사지도를 OCaml 타입으로 정의하면,

```
type map = End of treasure
         | Branch of map * map
         | Guide of string * map
```

**보물상자마다 필요한 열쇠의 모양**은 보물상자의 위치가 전체 탐사지도에서 어디냐에 따라 결정되는데, 지도에서 각 지역이 암시하는 열쇠의 모양은 다음의 조건으로 결정된다:

현재위치(지도) $e$	위치의 뜻	열쇠모양의 조건
$\star$	$\star$ 보물상자	- (Bar)
$x$	$x$ 라는 이름의 보물상자	현재 위치에서 $x$ 를 열어줄 열쇠 모양
$\boxed{x}e_1$	안내판 $\boxed{x}$ 이 앞에있는 지도 $e_1$	$e_1$ 안에서 만날 보물상자 $x$ 의 열쇠가 $\alpha$ 이고 $e_1$ 의 시작점이 암시하는 열쇠모양을 $\beta$ 라고 하면, 현재 위치가 암시하는 열쇠모양은 $(\alpha, \beta)$ (왼쪽가지 $\alpha$ , 오른쪽가지 $\beta$ ).
$e_1 e_2$	$e_1$ 과 $e_2$ 로 갈라지는 갈림길	$e_1$ 의 시작점이 암시하는 열쇠모양은 $(\alpha, \beta)$ 이어야 하고 $e_2$ 의 시작점이 암시하는 열쇠모양은 $\alpha$ 이어야 한다. 이때, 현재 위치가 암시하는 열쇠모양은 $\beta$ .

예를들어, 각 지도를 성공적으로 탐험할 최소의(열쇠들 크기의 합을 기준으로) 열쇠꾸러미는 다음과 같다:

1. 지도  $x$  에는  $\{-\}$ .
2. 지도  $\boxed{x}x$  에는  $\{-\}$ .
3. 지도  $(\boxed{x}x)|\star$  에는  $\{-\}$ .
4. 지도  $(\boxed{x}(x|x))|\star$  를 성공적으로 탐험하는 것은 불가능.
5. 지도  $(\boxed{x}x)|((\boxed{y}y)|\star)$  에는  $\{-\}$ .
6. 지도  $(\boxed{x}x)|(\boxed{y}y)$  에는  $\{-, (-, -)\}$ .
7. 지도  $x|\star$  에는  $\{-, (-, -)\}$ .

다음의 타입에 맞도록, 위와같은 일을 하는 `getReady` 함수

```
getReady: map → key list
```

를 정의하기 바랍니다. □

**Exercise 3** (30pts) “즐거운 고민”

저의 고민은 조카들이 모두 행복해 할 수 있도록 가장 저렴하게 선물을 준비하는 것입니다. 저의 조카들은 시샘이 많습니다. 매년 이맘때쯤이면 저는 조카들에게 선물을 한 꾸러미씩 나눠주는데, 받고나면 조카들끼리 다른 형제들이 받은 선물을 시샘하면서 서로 조르고 울고. 그래서 다시 정리해서 주면 또 만족스럽지 않아서 조르고 울고.

저는 그 고민을 다음과 같이 풀기로 했습니다. 선물 쇼핑을 나가기전에 조카들에게 올해 받을 선물의 후보들을 알려주고 각자는 그중의 부분집합을 선물로 받을 것이라고 선언합니다. 그러곤 조카들에게 각자가 만족할(싸우지않을) 조건을 얘기하라고 합니다. 저는 가장 적은 비용으로 이러한 조건들을 모두 만족시키도록 선물꾸러미들을 준비합니다.

조카들의 조건들은 이런식입니다: “나는 최소한 만년필과 동생 영희가 받은 선물만큼은 받아야 해요.” “나는 최소한 철수오빠와 숙희언니의 선물들에 공통된 것들 하고, 영숙이 선물중에서 CD뽐 것은 가져야 해요” 등등. 예를 들어 A, B, C 세명의 조카가 있다면, 조건에 따라 받는 선물은 다음과 같지요:

- 샘만 많은 조카들은 아무것도 못받습니다. A: “최소한 B 만큼”, B: “최소한 A 만큼”, C: “최소한 B 만큼.”
- 까다로운 조카들도 아무것도 못받습니다. A: “최소한 B 만큼에서 만년필 말고”, B: “최소한 A 만큼에서 CD 말고”, C: “최소한 B 만큼에서 USB 말고.”
- 탐욕스런 조카들도 아무것도 못받습니다. A: “최소한 B와 C만큼”, B: “최소한 A와 C만큼”, C: “최소한 A와 B만큼.”
- 샘이 없는 조카들은 원하는 것만 받습니다. A: “최소한 만년필”, B: “최소한 CD”, C: “최소한 USB.”

구현을 위해서, 조카의 조건(require)은 다음과 같은 꼴로 표현된다고 정합시다:

나는 최소한 ( $cond_1$  그리고 ... 그리고  $cond_k$ )을 받아야 해요.”

OCaml 타입으로 정리하면 다음과 같습니다:

```
type require = id * (cond list)
and cond
    = Items of gift list          (* 선물들 *)
    | Same of id                  (* 어느 조카의 선물들 *)
```

$\mid \text{Common of cond} * \text{cond}$  (\* 두조건에 공통된 선물들 \*)  
 $\mid \text{Except of cond} * \text{gift list}$  (\* 조건에서 어느 선물들은 빼  
 고 \*)

$\text{and gift} = \text{int}$  (\* 선물 번호 \*)  
 $\text{and id} = A \mid B \mid C \mid D \mid E$  (\* 조카 이름 \*)

위의 다섯 조카들의 조건(require)을 받아서 **최소의** 선물쇼핑 리스트를 작성하는 shoppingList를 작성하기 바랍니다:

$\text{shoppingList: require list} \rightarrow (\text{id} * \text{gift list}) \text{ list}$

결과는 조카마다 사주어야 할 선물들의 리스트입니다. 예를들어, 조카들의 조건이 다음과 같을때

$A$ : 최소한  $\{1, 2\}$  하고  $\text{common}(B, C)$ 를 받아야.  
 $B$ : 최소한  $\text{common}(C, \{2, 3\})$ 를 받아야.  
 $C$ : 최소한  $\{1\}$  하고  $(A \text{ except } \{3\})$ 를 받아야.

그러면 최소의 선물꾸러미들은  $A$ 에게  $\{1, 2\}$ ,  $B$ 에게  $\{2\}$ ,  $C$ 에게  $\{1, 2\}$ 이므로, shoppingList의 결과는

$[(A, [1, 2]), (B, [2]), (C, [1, 2]), (D, \text{nil}), (E, \text{nil})]$

입니다. (조카마다 받는 선물꾸러미는 “집합”입니다, 즉, 한 선물 꾸러미에는 같은 선물이 두개이상 포함되지는 않습니다).

□