

Homework 4

SNU 4190.210 Fall 2008

Kwangkeun Yi

due: 10/24(Fri) 24:00

Exercise 1 “즐거운 고민”

올바른 프로그램인지를 확인해 보기 까다로운 또 다른 예입니다.
여러분의 프로그램이 왜 올바른 지 쉽게 주장할 수 있는 방안이 있는 지 궁리해 보세요.

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 CD뺀 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.”

조카의 조건은 다음과 같은 꼴로 표현된다고 정합시다:

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

이제 조카들의 조건들을 받아서 최소의 선물쇼핑 리스트를 작성하는 `shoppingList`를 작성하기 바랍니다:

```
shoppingList: (id * cond) list -> (id * gift list) list
```

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

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

B: 최소한 $common(C, \{2, 3\})$ 를 받아야.

C: 최소한 ($\{1\}$ 하고 (A except $\{3\}$))를 받아야.

최소의 선물꾸러미들은 A에게 $\{1, 2\}$, B에게 $\{2\}$, C에게 $\{1, 2\}$ 이므로, `shoppingList`의 결과는

((A . (1 2)) (B . (2)) (C . (1 2)))

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

구현을 위해서, 선물의 조건을 만드는 함수와 사용하는 함수는 다음과 같이 주어집니다.

만드는 함수들:

```
mustItems: giftlist -> cond          (* 가져야할 선물들 *)
mustBeTheSame: id -> cond            (* 어느 조카의 선물들 *)
mustHaveExceptFor: cond * gift list -> cond (* 조건에서 어느 선물들은 빼고 *)
mustHaveCommon: cond * cond -> cond   (* 두 조건에 공통된 선물들 *)
mustAnd: cond * cond -> cond         (* 두 조건 모두 만족해야 *)
```

사용하는 함수들:

```
isItems: cond -> bool              isSame: cond -> bool
```

```

isExcept: cond -> bool
isAnd: cond -> bool
whichItems: cond -> gift list
condExcept: cond -> cond
condCommon: cond -> cond * cond
isCommon: cond -> bool
whoTheSame: cond -> id
itemsExcept: cond -> gift list
condAnd: cond -> cond * cond

```

위에서 gift는 정수로, 조카 이름 id는 Scheme의 심볼로 구현됩니다. 예를 들어, 위에서 조카 A의 조건(cond)은 다음과 같이 만들어 지겠지요:

```

(mustAnd (mustItems '(1 2))
         (mustHaveCommon (mustBeTheSame 'B)
                          (mustBeTheSame 'C)))

```

□