

[基本プログラムコース]

## 演習問題 (r11) [基本プログラムコース]

- (1) 整数を要素とする長さ  $n$  ( $n > 0$ ) の 2 つのリストをそれぞれ  $L, M$  とする。すべての  $i$  ( $i \geq 0$ ) について、 $L, M$  の  $i$  番目の要素の和がリスト  $N$  の  $i$  番目の要素であるという関係を表す述語 `sum_of_pair(L,M,N)` のプログラムを作成せよ。たとえば、`sum_of_pair([1,2,3],[-1,2,0],N)` は  $N=[0,4,3]$  となって成功する。
- (2) 正の整数からなるリスト  $L$  に対して、その要素で 2 桁の数の個数が  $N$  であるという関係を表す述語 `two_digits(L,N)` のプログラムを作成せよ。たとえば、`two_digits([3,14,1,102,25],N)` は  $N=2$  となって成功する。
- (3) (書籍名, 在庫の有無, 出版年, 出版社) を要素とするリスト  $L1$  に対して、発行年が 2010 年以降で在庫がある書籍のみを集めてその出版社とともに (書籍名, 出版社) という組をつくって要素としたリストが  $L2$  であるという関係を表す述語 `book_list(L1,L2)` のプログラムを作成せよ。たとえば、  
`book_list([(b1,no,2017,pb1),(b2,yes,1998,pb2),(b3,yes,2015,pb3),(b4,no,2000,pb4)],L2)`  
 は  $L2=[(b3,pb3)]$  となって成功する。
- (4) 以下の文法 ( $G$ ) が定義されているとき、与えられた表現  $X$  が式 ( $\text{Expr}$ ) であることを判定する `isExp(X)` のプログラムを作成せよ。たとえば `isExp(plus(0,suc(0)))` は成功し、`isExp(suc(plus(0,0)))` は失敗する。ただし、 $X$  にはこの文法に従う表現のみが入力されると考えてよい。

( $G$ )

```
Expr ::= plus(Expr,Term) | minus(Expr,Term) | Term
Term ::= 0 | suc(Term)
```

- (5) 以下のような木構造が定義されているとする。このように定義された木  $T1$  が与えられたとき、ノードのラベル  $a$  をすべて  $c$  に変更して得られるものが木  $T2$  であるという関係を表す `subst_label(T1,T2)` のプログラムを作成せよ。たとえば、`subst_label(t(a,t(b,a,b),t(a,b)),T2)` は  $T2 = t(c, t(b c, b), t(c, b))$  となって成功する (図 11.1 参照)。ただし、 $T1$  には以下の木構造に従う表現のみが入力されると考えてよい。

```
Tree ::= t(Node,Tree,Tree) | t(Node,Tree) | Node
Node ::= a | b
```

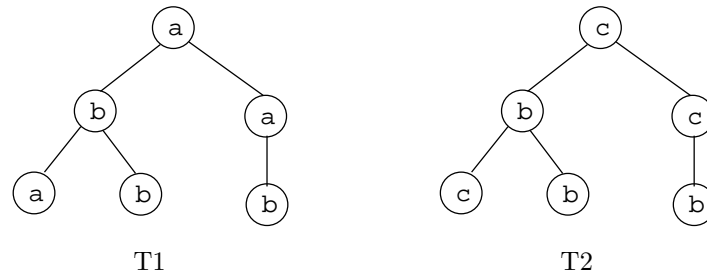


図 11.1

[応用プログラムコース]

### 練習問題 [応用プログラムコースのみ]

1. 整数を要素とする (要素の重複を認める) リスト  $L$  が小さいものから大きなものへと順番にならんでいる場合は成功し, そうでなければ失敗する述語  $\text{increase}(L)$  のプログラムを作成せよ。

### 演習問題 (r11) [応用プログラムコース]

- (6) 練習問題を参考にして  $L$  が等差数列になっているかどうかを判定する述語  $\text{arith\_prog}(L)$  のプログラムを作成せよ。ただし, 長さ 2 以下のリストは入力されないとしてよい。たとえば,  $\text{arith\_prog}([-2,0,2,4,6])$  は成功する。
- (7) 学生を要素とするリストを  $\text{Students}$  とし, 各要素 (学生) は学生番号, 研究室のペアを使った項  $\text{st}(\text{ID}, \text{Lab})$  として表されるとする。  $\text{Students}$  に含まれる学生がすべて同一研究室に所属するかどうかを判定する述語  $\text{same\_lab}(\text{Students})$  のプログラムを作成せよ。たとえば,  $\text{same\_lab}([\text{st}(100, \text{network}), \text{st}(200, \text{network}), \text{st}(222, \text{network})])$  は成功し,  $\text{same\_lab}([\text{st}(100, \text{network}), \text{st}(101, \text{logic}), \text{st}(102, \text{network})])$  は失敗する。
- (8) 以下のような木構造が定義されており, 各ノードには異なる整数がラベルとして付加されているものとする。このように定義された木  $\text{Tree}$  において, 指定されたラベル  $\text{Label}$  が出現するノードが存在するかどうかを判定する述語  $\text{exist\_label}(\text{Label}, \text{Tree})$  のプログラムを作成せよ。ただし,  $\text{Tree}$  には以下の木構造に従う表現のみが入力されると考えてよい。

```
Tree ::= t(Node, Tree, Tree) | t(Node, Tree) | Node
Node ::= Integer
```

たとえば,  $\text{exist\_label}(2, t(1, t(1,2,3), t(2,1)))$  は成功する。

- (9) 演習問題 r10(9) のプログラムを改訂してループのあるグラフに対しても有効であるような縦型探索を行う  $\text{dfs\_loop}(\text{Path})$  の繰り返し型プログラムを作成し図 11.2 のグラフに関して全解 (各ノードを高々 1 回しか通らないもののみを解とする) が求まることを確認せよ。たとえば,  $\text{dfs\_loop}(\text{Path})$  の解の 1 つは  $\text{Path}=[g, f, d, c, a, s]$  であり, これを含めて合計 7 個の解答がある。

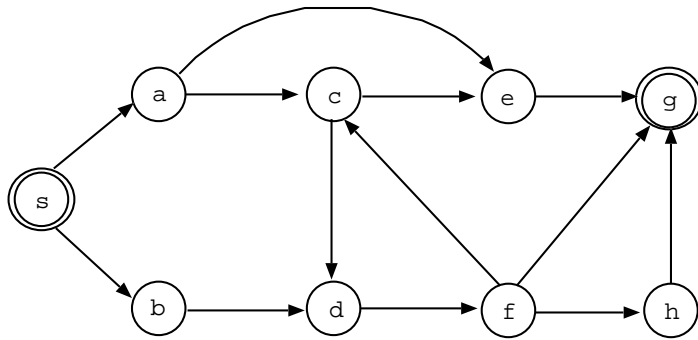


図 11.2

(10) 表現  $Exp$  が以下のような文法で定義されているとする .

$$Exp ::= Nat \mid minus(Exp) \mid sum(Exp,Exp) \mid diff(Exp,Exp)$$

$$Nat ::= 0 \mid 1 \mid 2$$

与えられた表現を下書き換え規則を使って  $diff$  を除去したものに書き換える .

書き換え規則 : ( 任意の表現  $A, B$  に対して左辺から右辺に書き換える . )

[R1]  $diff(A, B) = sum(A, minus(B))$

[R2]  $minus(minus(A)) = A$

与えられた表現  $E$  をまず書き換え規則 [R1] を使って書き換え , そのあと [R2] を使って書き換えた結果が表現  $F$  であるという関係を表す述語  $elim(E, F)$  のプログラムを作成せよ .  
 ただし , この文法にしたがった表現以外のものは入力されないと仮定してよい . たとえば ,  
 $elim(sum(diff(2, minus(1)), sum(diff(1, 2), 1)), A)$  は  
 $A = sum(sum(2, 1), sum(sum(1, minus(2)), 1))$  となって成功する .