

理解度テスト合格者は応用プログラムコース (2-3 ページ) または基本プログラムコース (1 ページ) にすすみ, 不合格者 (欠席者を含む) は基本プログラムコース (1 ページ) にすすむ。理解度テストに合格しなければ単位は取得できないが, テストに合格しても単位取得を確約するものではない (通常のレポートや出席状況も反映するため)。最終回に再度理解度テストを行う。

応用プログラムコースにすすんだ者は 100 点を最高点とし, 基本プログラムコースにとどまった者は 80 点を最高点とし, いずれも最終的に習得した能力で本科目の最終評価点とする。途中で変更はできないのでよく考えて選択すること。

[基本プログラムコース]

演習問題 (r10)[基本プログラムコース]

- (1) X, Y を n 個の実数から成る長さの等しいリスト $[X_1, \dots, X_n], [Y_1, \dots, Y_n]$ とする。各 i ($i = 1, \dots, n$) について X_i, Y_i がともに正の数のおきのみ X_i, Y_i の差をとり, それらの 2 乗の和が D であるという関係を表す述語 $\text{dist}(X, Y, D)$ のプログラムを作成せよ。
- (2) リスト L に含まれるアトム a, b の個数を合わせたものが C であるという関係を表す述語 $\text{count_letters}(L, C)$ のプログラムを作成せよ。
- (3) 整数から成るリスト L に含まれる偶数のリストが Even , 奇数のリストが Odd であるという関係を表す述語 $\text{even_odd}(L, \text{Even}, \text{Odd})$ のプログラムを作成せよ。
- (4) (名前, 点数) を要素とするリスト L に対して, 点数が 60 以上の要素の個数が N であるという関係を表す述語 $\text{n_of_passed}(L, N)$ のプログラムを作成せよ。
- (5) 以下の文法が定義されているとき, 与えられた項 Term に a, b 両方ともが出現するかを判定する述語 $\text{both_occur}(\text{Term})$ のプログラムを作成せよ。ただし, Term には文法にかなったものしか入力されないと仮定してよい。また, 必要に応じて自分で新しい述語を定義せよ。(対象言語とプログラミング言語を明確に区別して考えること。)

```
Term ::= Alphabet | Digit | f(Term) | g(Term, Term)
Alphabet ::= a | b
Digit ::= 0 | 1
```

データの段階的な集積方法

末端再帰型と繰り返し型

例：1 から 5 までの整数のリストを作成．

- 末端再帰型 – リストは前から値がはいっていき，最後に空リストがはいる

```
accumulate1(L) :- accum1(1,L).  
  
accum1(6, []).  
accum1(N, [N|X]) :- N1 is N+1, accum1(N1,X).  
  
?- accumulate1(L).  
L = [1,2,3,4,5]
```

- 繰り返し型 – リストは第 3 引数にスタックされていき，最後に第 2 引数に渡される

```
accumulate2(L) :- accum2(1,L, []).  
  
accum2(6,S,S).  
accum2(N,L,S) :- N1 is N+1, accum2(N1,L, [N|S]).  
  
?- accumulate2(L).  
L = [5,4,3,2,1]
```

練習問題 [応用プログラムコースのみ]

1. ベクトル $X = (X_1, \dots, X_n)$, $Y = (Y_1, \dots, Y_n)$ がそれぞれリスト $[X_1, \dots, X_n]$, $[Y_1, \dots, Y_n]$ で表されるとき， X, Y の内積，すなわち $\sum_{i=1}^n X_i \cdot Y_i$ が IP であるという関係を表す述語 `inner_product(X,Y,IP)` の再帰型プログラムは以下の通りである．

```
inner_product([], [], 0).  
inner_product([X|X1], [Y|Y1], Z) :- inner_product(X1,Y1,Z1), Z is X*Y+Z1.
```

このプログラムの繰り返し型版 `inner_product2(X,Y,IP)` を作成せよ．たとえば，`inner_product2([1,2],[3,4],IP)` を実行すると，`IP=11` となって成功する．

演習問題 (r10) [応用プログラムコース]

- (6) `inner_product` を参考に、リスト L の長さ N を求める `list_length2(L,N)` の繰り返し型プログラムを作成せよ。たとえば、`list_length2([1,2,3],N)` を実行すると、 $N=3$ となって成功する。(Hint: `list_length2` は、スタックに相当する引数が追加された別の述語を呼び出す。)
- (7) 2 ページ目の繰り返し型の説明で用いた `accum2` を参考にリスト $L1$ の要素を逆向きに並べかえたものがリスト $L2$ になっているような `reverse2(L1,L2)` の繰り返し型プログラムを作成せよ。たとえば、`reverse2([1,2,3],L2)` を実行すると、 $L2=[3,2,1]$ となって成功する。(Hint: `reverse2(L1,L2) :- rev(L1,L2,[])`。として、`rev` のプログラムを作成せよ。ただし、`rev` の第 3 引数はスタックに相当する。)
- (8) リスト $L1$ の要素がすべて整数であるとする。 $L1$ の要素の中でその値が Min 以上 Max 以下のものすべてから構成したリストが $L2$ であるような関係を表す述語 `mid2(L1,Min,Max,L2)` のプログラムを繰り返し型で作成せよ。たとえば、`mid2([5,12,2,25,18],10,20,L2)` は $L2=[18,12]$ となって成功する。`mid2` は演習問題 r4(2) で作成した `mid` の繰り返し型プログラムであり、要素がリストの後から順に求められるため $L2$ は `mid` の場合と逆順になる。
- (9) ループのない有向グラフとそのノード 2 つが初期ノード、目標ノードとして与えられたとき、初期ノードから目標ノードに到る経路 $Path$ を縦型探索を使って求める `dfs2(Path)` の繰り返し型プログラムを作成し、図 10.1 のグラフに関して全解が求まることを確認せよ。`dfs2` は演習問題 r9(5) で作成した `dfs` の繰り返し型プログラムであり、要素がボトムアップに求められるため $Path$ は `dfs` の場合と逆順になる。(r9 練習問題 2, r9(5) と同様に、`start(S)`, `goal(G)`, `edge(N,M)` を使用してグラフの形状を記述して考えよ。)
- (10) (9) のプログラムを改訂して初期ノードから目標ノードに到る経路 $Path$ の探索の深さを $MaxDepth$ までに限定した縦型探索を行う `dfs_bound(MaxDepth,Path)` の繰り返し型プログラムを作成し図 10.1 のグラフに関して深さ 1,2,3,4 それぞれについて(同じ部分道を通るものもそれぞれ別解とする) $Path$ が求まることを確認せよ。たとえば、`dfs_bound(2,Path)` の解としては $Path=[g,b,s]$ のみが求められる。

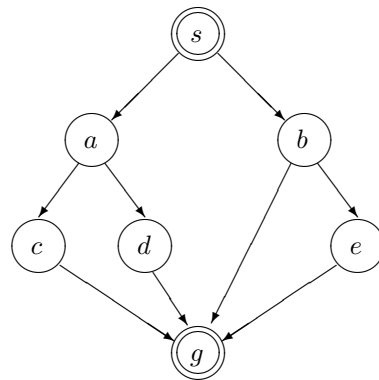


図 10.1