

## デバッグについて

- 実行のステップごとのトレース
  - ?- trace,goal.<CR>
- 割り込み – 実行中に停止しなくなった場合に数回実行するとよい
  - C-c
- その他よく使う命令
  - a (abort) デバッグの中断
  - w (write) 省略された部分を完全に表示
  - s (skip) それ以下の実行の表示をスキップ
- トレースの終了
  - ?- notrace.<CR>
- デバッグモードの終了
  - ?- nodebug.<CR>

詳しくは教科書または SWI-Prolog の URL を参照のこと .

## 練習問題

1. `are_edges(N,M)` は、有向グラフにおいてノード  $N$  を始点とするエッジの終点のリストが  $M$  であるという関係を表すとする . 図 5.1 において成り立つ `are_edges` をすべて記述してデータベースを作成せよ . (たとえばノード  $a$  を始点とする場合は `are_edges(a,[b,c])` となる .) 次に、演習問題 r3(1) で作成した述語 `list_length` を使って、ノード  $N$  を始点とするエッジの数が  $K$  本であるという関係を表す述語 `n_of_edges(N,K)` のプログラムを作成せよ . 図 5.1 だけでなく、一般のグラフも扱えるようにすること . たとえば、`n_of_edges(a,K)` は  $K=2$  となって成功する .

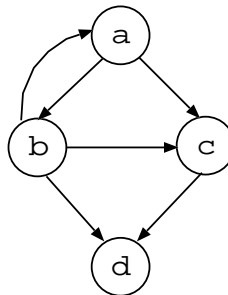


図 5.1

2. 組み込み述語 `member` を使ってリスト  $L$  が重複する要素をもつかどうかを判定する述語 `has_duplication(L)` の再帰的なプログラムを作成せよ . たとえば、`has_duplication([a,b,a])` は `true` となり、`has_duplication([a,b])` は `false` となる .

## 演習問題 (r5)

\* のついている問題はオプションなのでできる者のみ解答せよ。

以下の問題において、リストの先頭は0番目ではなく1番目と数える。

- (1) 有向グラフにおいて、全ノードのリストを  $G$ 、各ノード  $N$  に対して、それを始点とするエッジのリストが練習問題 1 のように述語 `are_edges` を使ってデータベースに記述されているとき、このグラフのエッジの総数は  $E$  であるという関係を表す述語 `n_of_all_edges(G,E)` のプログラムを練習問題 1 で作成した述語 `n_of_edges` を使って作成せよ。たとえば、図 5.1 の場合、`n_of_all_edges([a,b,c,d],E)` は  $E=6$  となって成功する。
- (2) 整数から成る長さ 1 以上のリスト  $L$  のすべての要素の平均値が  $V$  であるという関係を表す述語 `average(L,V)` のプログラムを、r3 練習問題 2 の `sum_list` および演習問題 r3(1) の `list_length` を使って作成せよ。
- (3) 組み込み述語 `member` を使って、リスト  $L1, L2$  が共通の要素を持っていることを表す述語 `has_common_element(L1,L2)` のプログラムを作成せよ。たとえば `has_common_element([a,b,c],[d,b])` は `true` となり、`has_common_element([a,b,c],[d])` は `false` となる。
- (4) 組み込み述語 `member` を使って、リスト  $L1$  から重複している要素をすべて削除した結果がリスト  $L2$  であるような関係を表す述語 `del_duplication(L1,L2)` のプログラムを作成せよ。ただし、重複している要素がある場合、リストの後の方のものを残し、前の方のものを削除するものとする。たとえば、`del_duplication([a,b,a,a],L2)` は  $L2=[b,a]$  となって成功する。
- (5) リスト  $List$  の要素がすべて整数であるとする。 $List$  の要素の中で、ある整数  $X$  未満のものリストが  $S$  であり、 $X$  以上の要素のリストが  $L$  であるという関係を表す述語 `partition(List,X,S,L)` のプログラムを作成せよ。 $List, X$  の具体値が与えられた時、 $S,L$  の具体値が求まればよいものとする。たとえば、`partition([4,6,2],5,S,L)` は  $S=[4,2], L=[6]$  となって成功する。
- (6)\* リスト  $L1, L2$  と、ある整数  $X$  に対して、 $L1$  と  $L2$  の間に  $X$  を挿入したリストが  $L$  であるという関係を表す述語 `connect_lists(L1,X,L2,L)` のプログラムを、組み込み述語 `append` を使って作成せよ。たとえば、`connect_lists([1,2],3,[4],L)` は  $L=[1,2,3,4]$  となって成功する。
- (7)\* (5) で作成した述語 `partition` および (6) で作成した述語 `connect_lists` を使って、要素が整数であるリスト  $L1$  をクイックソートによって昇べきの順 (小さいものから大きなものへの順) に並びかえたリストが  $L2$  であるという関係を表す述語 `qsort(L1,L2)` のプログラムを作成せよ。ただし、与えられたリストの最初の要素をピボット (大きい数と小さい数をわけするための基準値) としてとるものとし、 $L1$  の具体値が与えられた時、 $L2$  の具体値が求まればよいものとする。たとえば、`qsort([5,4,6,2],L2)` は  $L2=[2,4,5,6]$  となって成功する。
- (8) 練習問題 2 の解答例プログラムの (i) 各節の論理的意味および (ii) `?- has_duplication([a,b,a])` を実行したときの実行過程を示せ。(i) については**命題の形になっていること**、すなわち、引数への入出力を書くのではなく、「 $C$  である」「 $A$  かつ  $B$  ならば  $C$  である」のように記述すること。 $[X|L1]$  は、「頭部が  $X$ 、尾部が  $L1$  のリスト」と書いてもよいし、このままでもよい。(ii) については第 3 回資料にある参考例、第 1 回の資料「Prolog の実行過程」、第 2 回資料「sum の実行過程の補足」などを参考に、「ゴール」「実行」「単一化 (ユニフィケーション)」という用語をすべて用いて段階的に説明せよ。(トレースを貼り付けてはいけない。)