

再び単一化 (=) と評価 (is)

```

%-----
% r3_2 偶数要素の数

number_of_evens([],0).
number_of_evens([E|L1],C) :- 0 is mod(E,2), number_of_evens(L1,C1), C is C1+1.
number_of_evens([_|L1],C) :- number_of_evens(L1,C).

%-----
% r3_3 リスト内の特定の名前の数

number_of_name([],0).
number_of_name([ann|X1],N) :- number_of_name(X1,N1), N is N1+1.
% number_of_name([X|X1],N) :- X=ann, number_of_name(X1,N1), N is N1+1.
number_of_name([_|X1],N) :- number_of_name(X1,N).

% number_of_name([Z|X1],N) :- X=1, number_of_name(X1,N1), N is N1+1.
% number_of_name([1|X1],N) :- number_of_name(X1,N1), N is N1+1.
% これらも単一化なので記述可能。is を使用するのはあくまでも右辺の計算が必要な場合のみ

```

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。

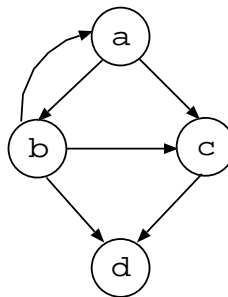


図 5.1

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。`member(X,Y)` は X がリスト Y の要素である関係を表す述語である。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 `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)` のプログラムを作成せよ。
[考え方] 要素の合計の計算，要素の個数の計算に対応する述語を生成し，それらを使って `average` を定義する。
- (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 以上の要素のリストが L であり， X 未満のものリストが S であるという関係を表す述語たとえば `partition(List,X,L,S)` のプログラムを作成せよ。List, X の具体値が与えられた時， L, S の具体値が求めればよいものとする。たとえば，`partition([4,6,2],5,L,S)` は $L=[6], S=[4,2]$ 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， $L2$ の具体値が求め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) r5 の練習問題 1 についてレポートせよ。以下を記述すること。(i) プログラムの各節の論理的意味 (命題の形になっていること)，(ii) `?- n_of_edges(a,K)` を実行したときの動作 (トレースを貼り付けてはいけない。「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どのゴールとどの節のヘッドが単一化されて変数がどう書き換わり，どのゴールが呼ばれるなどを段階的に記述すること)。