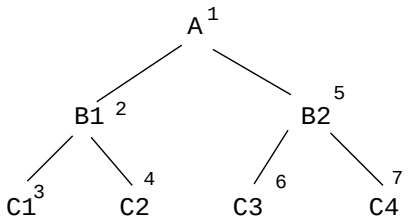
実行過程

基本的な実行順序 - 深さ優先 (depth-first)

例 1

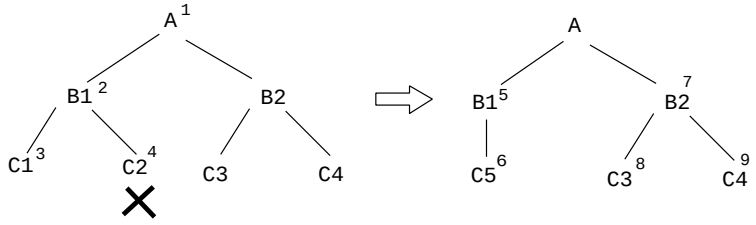
```
A :- B1, B2.   % 確定節 (rule) B1 /\ B2 -> A  の意味

B1 :- C1, C2.
B2 :- C3, C4.

% 単位節 (fact)
C1.  C2.  C3.  C4.
```



後戻り (backtrack)

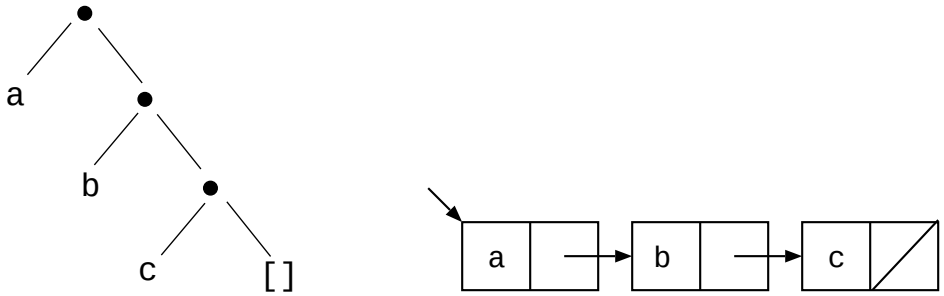
例 2

```
A :- B1, B2.   % 確定節 (rule) B1 /\ B2 -> A  の意味

B1 :- C1, C2.
B1 :- C5.
B2 :- C3, C4.

% 単位節 (fact)
C1.  C3.  C4.  C5.
```



リスト構造

[a,b,c]



- リストは頭部 (head) と尾部 (tail) から成る.
- **尾部はまたリストである.**
- リストの要素の数をリストの長さという.
- H を頭部，T を尾部とするリストを [H|T] と表現する.
  表記がまぎらわしいが，list(H,T) と書いてあるのと同じと思えばよい．要はリストが 2 引数から構成されるデータ構造であり，H はリストの要素，T はリストそのものになることを理解すること.

実行の結果行われる単一化 (unification)

```
?- [H|T]=[a,b,c].
       H=a, T=[b,c]
?- [H|T]=[c].
       H=c, T=[]
?- [H1,H2|T]=[a,b,c].
       H1=a, H2=b, T=[c]
```

例 1：メンバー (text p.69 参照)

```
member(X,[X|T]).
member(X,[Y|T]) :- member(X,T).
```

例 2：リストのコピー

```
copy_list([],[]).
copy_list([X|X1],[X|Y1]) :- copy_list(X1,Y1)
```

例 3：リストの連接 (text pp.69-70 conc と同じ)

```
append([],Y,Y).
append([X|Xs],Y,[X|Zs]) :- append(Xs,Y,Zs).
```

練習問題

1. L1, L2 を要素がすべて整数であるようなリストとする．L1 の各要素を 2 倍した値が L2 の各要素になっているという関係を表す述語 double_num(L1,L2) のプログラムを作成せよ．たとえば double_num([1,2,3],L) は L=[2,4,6] となって成功する．
2. 要素がすべて整数であるようなリスト L の要素の和が S であるという関係を表す述語 sum_list(L,S) のプログラムを作成せよ．たとえば，sum_list([1,2,3],S) は S=6 となって成功する．
3. リスト L1 の要素がすべて整数であるとする．L1 の要素の中で偶数のみを取り出したリストが L2 であるという関係を表す述語 even_list(L1,L2) のプログラムを作成せよ．組み込みオペレータ mod を使用してよい．(mod(N,M) は整数 N を M で割った余りを返す関数，text p.87 参照．) たとえば even_list([3,5,4,10,8],L2) は L2=[4,10,8] となって成功する．(注意：確定節を１つ増やすことで if-else に対応させられる．)

演習問題 (r3)

∗ 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．また，(5)(6) はリストを使用しない．最初に正しい解が得られればよく，別解を求める必要はない．

(1) リスト L の長さが N であるという関係を表す述語 list_length(L,N) のプログラムを作成せよ．たとえば，list_length([a,b,c],N) は N=3 となって成功する．

(2) リスト L の要素がすべて整数であるとする．L の要素の中で偶数の個数が C 個であるという関係を表す述語 number_of_evens(L,C) のプログラムを作成せよ．たとえば number_of_evens([3,5,4,10,8],C) は C=3 となって成功する．

(3) 名前のリスト L の要素の中で ann の個数が C 個であるという関係を表す述語 number_of_name(L,C) のプログラムを作成せよ．たとえば number_of_name([ann,bob,cris,ann],C) は C=2 となって成功する．

(4) 要素が正の整数であるリスト L に対して，L の要素で値が 10 以下であるものすべての和が S であるという関係を表す述語 sum_list_small(L,S) のプログラムを作成せよ．たとえば sum_list_small([15,2,9],X) を実行すると，X=11 となって成功する．

(5) edge2(N,M,D) が有向グラフにおいてノード N から ノード M への長さ D のエッジがあるという関係を表すとする．図 3.1 において，与えられたノード N から M までの距離を L とするとき，N,M,L の関係を表す述語 dist2(N,M,L) を edge2 を用いて**再帰的に**定義せよ．ただし，2 点を入力，距離を出力として正常動作すればよいものとする．(複数解が存在する場合，全解が求められることを確認せよ．)

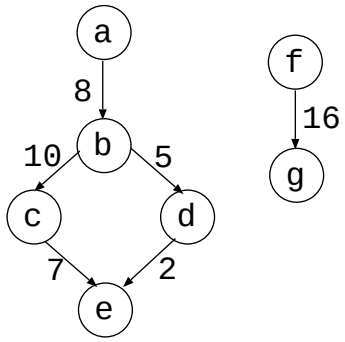


図 3.1

(6)* exp(N,M,X) を N の M 乗が X があるという関係を表すとするとき，exp のプログラムを作成せよ．(ただし，N,M,X には 0 以上の自然数のみが入力されるのと考えてよい．) たとえば exp(3,4,81) は成功し，exp(3,4,X) は X = 81 を返す．また，実行時には第 1, 第 2 引数には入力されるものとし，exp(3,X,81), exp(X,4,81) などは計算できなくてよい．

(7)* 定数 A をリスト L1 の最後尾に挿入した結果がリスト L2 であるような insert_end(L1,A,L2) のプログラムを作成せよ．たとえば，insert_end([a,b],c,L2) は L2=[a,b,c] となって成功する．(Hint: copy_list を参考にせよ．)

(8)* リスト L1 がリスト L2 の先頭側部分リストかどうかを判定する述語 prefix(L1,L2) のプログラムを作成せよ．(部分リストは空リストおよびそれ自身を含む．) たとえば，prefix([a,b],[a,b,c,d]) は成功する．L1，L2 にはリストしか入力されないと仮定し，L1 の長さが L2 の長さ以下であると仮定してよい．

(9) 練習問題 1 の解答例プログラムの (i) 各節の論理的意味および (ii) `:- double_num([1,2,3],L).` 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」と書いてよい．(ii) については第 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．

演習問題 (9) の参考例: copy_list の場合

(i) 論理的意味
第 1 節：空リストをコピーしたものは空リストである.
第 2 節：リスト X1 をコピーしたものがリスト Y1 であるならば、
頭部が X、尾部が X1 のリストをコピーしたものは
頭部が X、尾部が Y1 のリストである.
（後半は、「リスト [X|X1] をコピーしたものがリスト [X|Y1] である」と書いてもよい.）

(ii) :- copy_list([1,2,3],L). を実行したときの動作

copy_list([1,2,3],L) を実行すると,
まず, copy_list の第 1 節と単一化しようとするが失敗する.
次に第 2 節と単一化すると成功して X=1,X1=[2,3],L=[1|Y1] となる.
この節のボディゴール copy_list([2,3],Y1) を呼び出す.

  copy_list([2,3],Y1) を実行すると,
  まず, copy_list の第 1 節と単一化しようとするが失敗する.
  次に第 2 節と単一化すると成功して X'=2,X1'=[3],Y1=[2|Y1'] となる.
  この節のボディゴール copy_list([3],Y1') を呼び出す.
  [ここで単一化される節は最初のものとは別ものなので変数名を変えていることに注意.
  つまり単一化される節は
    copy_list([X'|X1'],[X'|Y1']) :- copy_list(X1',Y1').
  である.]

    copy_list([3],Y1') を実行すると,
    まず, copy_list の第 1 節と単一化しようとするが失敗する.
    次に第 2 節と単一化すると成功して X''=3,X1''=[],Y1'=[3|Y1''] となる.
    この節のボディゴール copy_list([],Y1'') を呼び出す.

      copy_list([],Y1'') を実行すると,
      copy_list の第 1 節と単一化し、Y1''=[] となって成功する.

    この結果 Y1'=[3] となる.

  この結果 Y1=[2,3] となる.

この結果 L=[1,2,3] となる.