

## 実行過程

### 基本的な実行順序 - 深さ優先 (depth-first)

例 1

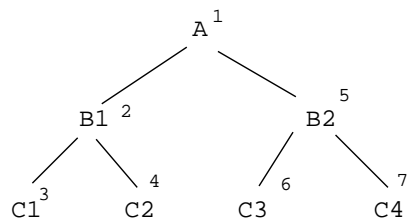
A :- B1, B2.    % 確定節 (rule) B1 /\ B2 -> A    の意味

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

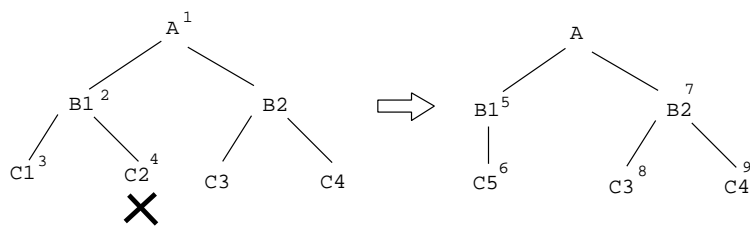
B1 :- C1, C2.

B1 :- C5.

B2 :- C3, C4.

% 単位節 (fact)

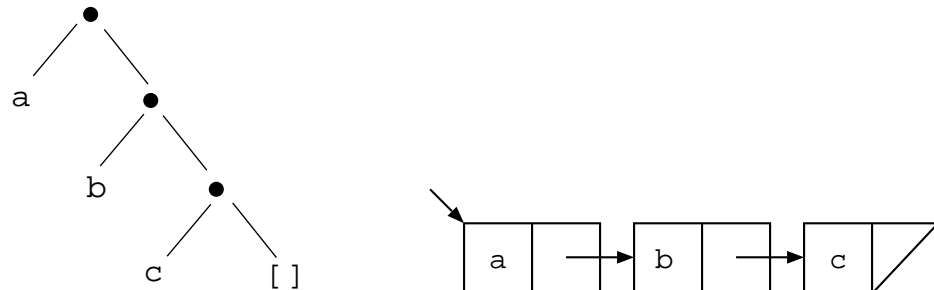
C1.   C3.   C4.   C5.



## リスト構造

注意：ここでいう「リスト構造」は「単方向連結リスト」をさすもので、たとえば Python の「リスト」や C の「配列」とは異なるものである。

$[a, b, c]$



- 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：リストの接続 (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.  $L_1, L_2$  を要素がすべて整数であるようなリストとする． $L_1$  の各要素を 2 倍した値が  $L_2$  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．例 2 の `copy_List` を参考にせよ．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  $L_1$  の要素がすべて整数であるとする． $L_1$  の要素の中で正の数のみを取り出したリストが  $L_2$  であるという関係を表す述語 `positive_numbers(L1,L2)` のプログラムを作成せよ．たとえば `positive_numbers([3,-5,4,0,-8],L2)` は  $L_2=[3,4]$  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．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(すなわち要素の個数)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) 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．
- (3) 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．
- (4) 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．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  $L_2=[4,10,8]$  となって成功する．たとえば `number_of_evens([3,5,4,10,8],C)` は  $C=3$  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．) この問題はリストを使用しない．

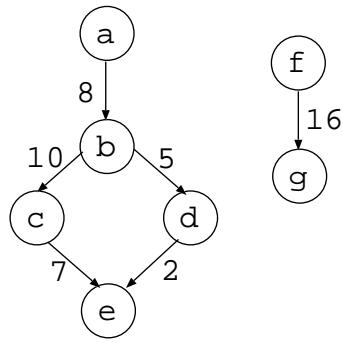


図 3.1

- (6)\* 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  $\text{insert\_end}(L1, A, L2)$  のプログラムを作成せよ。たとえば,  $\text{insert\_end}([a, b], c, L2)$  は  $L2 = [a, b, c]$  となって成功する。(Hint: `copy_list` を参考にせよ。)
- (7) 次ページにある `copy_list` の論理的意味と実行過程の説明を参考に, 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)  $\text{double\_num}([1, 2, 3], L)$  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) については今回の資料「list の実行過程」や以下の例などを参考に「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。
- (8) 今回の演習問題 r3 (1)-(7)(論理的意味と動作説明を含む) の解答にあたって生成 AI を少しでも使用したものの番号をすべて記述せよ。

演習問題 r3-(8) の参考例: `copy_list` の場合 (単にコピーするのではなく、意味を理解してから使用してください.)

(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  $X_1$  をコピーしたものがリスト  $Y_1$  であるならば ,

頭部が  $X$  , 尾部が  $X_1$  のリストをコピーしたものは

頭部が  $X$  , 尾部が  $Y_1$  のリストである .

(後半は「リスト  $[X|X_1]$  をコピーしたものがリスト  $[X|Y_1]$  である」と書いてもよい.)

(ii) :- `copy_list([1,2,3],L).` を実行したときの実行過程

`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  $X=1, X_1=[2,3], L=[1|Y_1]$  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  $X'=2, X_1'=[3], Y_1=[2|Y_1']$  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  $X''=3, X_1''=[], Y_1'=[3|Y_1'']$  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、 $Y_1''=[]$  となって成功する .

この結果  $Y_1'=[3]$  となる .

この結果  $Y_1=[2,3]$  となる .

この結果  $L=[1,2,3]$  となる .