

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 .

変数について

- 実行時に同じ節が再帰的に何度も呼ばれると、システムによって呼び出しごとにその節の変数には異なる変数が準備される . (「sum の実行過程の補足」参照 .)
- ‘_’ (underscore) からはじまる数字はシステムが導入した変数である . 同じ番号のものには同じ値が代入される .

練習問題

1. 整数を要素とするリスト L に含まれる正の数のリストが PL , 負の数のリストが NL であるという関係を表す述語 `pos_neg(L,PL,NL)` のプログラムを作成せよ . たとえば , `pos_neg([3,-10,0,16],PL,NL)` は `PL=[3,16]` , `NL=[-10]` となって成功する .

演習問題 (r4)

* 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) 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 3 で割り切れる要素のリストが L2 であり , 3 で割り切れない要素のリストが L3 であるという関係を表す述語 `divide_by_three(L1,L2,L3)` のプログラムを組み込みオペレータ `mod` を利用して作成せよ . たとえば , `divide_by_three([1,2,3,4,5,6],L2,L3)` は `L2=[3,6]` , `L3=[1,2,4,5]` となって成功する .
 なお , プログラム内で `do(X,Y) :- divide_by_three([1,2,3,4,5,6],X,Y)` . と定義しておく , 実行は `?- do(X,Y)` . と打ち込むだけですむ .

- (2) リスト L_1 の要素がすべて整数であるとする。 L_1 の要素の中でその値が Min 以上 Max 以下のものから成るリストが L_2 であるという関係を表す述語 $\text{mid}(L_1, \text{Min}, \text{Max}, L_2)$ のプログラムを作成せよ。たとえば $\text{mid}([5, 12, 2, 25, 18], 10, 20, L_2)$ は $L_2 = [12, 18]$ となって成功する。
- (3) リスト L_1 の 1 番目から N 番目の要素を順に並べたものがリスト L_2 であるという関係を表す述語 $\text{n_elements}(L_1, N, L_2)$ を定義せよ。ただし、 N としてはリストの長さを越える値は考えなくてよい。たとえば $\text{n_elements}([a, b, c], 2, L_2)$ を実行すると、 $L_2 = [a, b]$ となって成功する。
- (4) アルファベット小文字を要素とするリスト L_1 に含まれる a という文字をすべて削除したリストが L_2 であるという関係を表す述語 $\text{del_element}(L_1, L_2)$ のプログラムを作成せよ。たとえば、 $\text{del_element}([a, b, a, x], L)$ は $L = [b, x]$ となって成功する。
- (5) リスト L_1 に含まれる要素 X をすべて Y に置き換えたリストが L_2 であるという関係を表す述語 $\text{rewrite_list}(X, Y, L_1, L_2)$ のプログラムを作成せよ。たとえば、 $\text{rewrite_list}(a, x, [a, b, a, x], L)$ は $L = [x, b, x, x]$ となって成功する。
- (6)* member の定義を参考にして、長さ 1 以上のリスト L の N 番目の要素が X であるという関係を表す述語 $\text{nth_element}(L, N, X)$ を定義せよ。ただし、 N としてはリストの長さを越える値は考えなくてよい。また、リストの先頭は 0 番目ではなく 1 番目と数える (例: $[a, b, c]$ というリストに対しては N は 1, 2, 3 しか入力されないとしてよい。) たとえば $\text{nth_element}([a, b, c], 2, X)$ を実行すると、 $X = b$ となって成功する。
- [考え方] この問題では、再帰呼出しのたびに N が 1 つずつ減るとともに、リスト L の長さも 1 つずつ減る。 N の制約を考えると、停止条件としては $L = []$ をとるのはなく $N = 1$ をとる。このとき L は長さ 1 以上のリストになるので、そのときの冒頭の値をとってくればよい。
- (7)* (6) の述語の定義 $\text{nth_element}(L, N, X)$ を拡張してリストの長さが N 未満の場合は、エラーメッセージを出力するようにせよ。メッセージの出力には組み込み述語 write を利用し、 $\text{write}(\text{'The length is less than N'})$ のように記述する。また、 N は 1 以上の整数としてよい。
- [考え方] 停止条件が 2 つあることに注意。
- (8) r4 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) $\text{?- pos_neg}([3, -10, 0, 16], \text{PL}, \text{NL})$ 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。)