

関数と述語

ゴールにかけるのは述語であって関数ではない.

- 項：オブジェクトを表す
 $t ::= c \mid x \mid f(t, \dots, t)$
ただし, c は定数, x は変数, f は関数記号.
ann, X, parent_of(a) など.
- 述語：命題 (関係) を表す.
 $A ::= P \mid P(t, \dots, t)$
ただし, P は述語記号, t は項.
fr(apple), parent(X,Y), age(tom,80), correct, X is Y など. ('is' は中置として定義されている)

式の評価 (計算) = と is

$X = 1+2$ 左辺, 右辺とも評価をしない. X は $1+2$ と単一化される.

$X \text{ is } 1+2$ 左辺は評価をしない. 右辺は評価をする. X は 3 と単一化される.

数値については大体は下の書き方になる.

$X \text{ is } Y$ というゴールに対しては, Y に具体的な数値がいなければ成功しない

$X = \text{taro}$ X は taro と単一化される

$X \text{ is taro}$ これは誤り

再帰プログラミングの方法

0 から N までの自然数の和が M であるという関係を表す述語 `sum(N,M)` を定義する.

(1) 0 から 0 までの和は 0 である.

`sum(0,0).`

(2) 0 から N までの和 M と 0 から N-1 までの和 M1 の関係は? M1 に N を加えると M になる.

`M is M1+N`

(3) M,M1 はそれぞれ 0 から N までの和, 0 から N-1 までの和であるから `sum(N,M)`, `sum(N-1,M1)` が成り立つ.

`sum(N,M) :- sum(N-1,M1), M is M1+N.`

(4) `sum(N-1,M1)` と書くと, たとえば `2-1` は `'2-1'` という文字列と見なされ評価されない. これを評価させるため, 以下のように書き換える.

`sum(N,M) :- N1 is N-1, sum(N1,M1), M is M1+N.`

単一化と計算の方向性

Prolog は本来入出力の方向性をもたない. たとえばプログラムに書かれた

`parent(jim,pat).`

という節に対し,

?- `parent(jim,pat).` には yes を,

?- `parent(X,pat).` には `X=jim` を,

?- `parent(jim,X).` には `X=pat` を,

?- `parent(X,Y).` には `X=jim,Y=pat` をそれぞれ返す.

これは `parent(jim,pat)` と `parent(X,Y)` などを単一化 (一見した形を同じにする) 機構のためである.

しかし, 方向性を意識したプログラミングも可能である. たとえば, 上記 `sum(N,M)` は第 1 引数が入力, 第 2 引数が入力または出力という使われ方しか想定していない. したがって, ?- `sum(N,10).` のようなゴールは考える必要はない.

また, **全解探索の指示がない場合は解を 1 つ見つけて成功すればよく, 全解探索をする必要はない.**

練習問題

- 2 の X 乗が Y であることを表す述語 `pow2(X,Y)` を引き算を使って再帰的に定義せよ. たとえば, `pow2(5,Y)` は `Y=32` となって成功する. ただし, X を 0 以上の自然数とし, これ以外の入力はないものとする.
- N の X 乗が Y であることを表す述語 `pow(N,X,Y)` を引き算を使って再帰的に定義せよ. たとえば, `pow(2,5,Y)` は `Y=32` となって成功する. ただし, N,X を 0 以上の自然数とし, これ以外の入力はないものとする. 注意: 0 の 0 乗は 1 である.
- X を 3 で割った時のあまりが Z である関係を表す述語 `rem3(X,Z)` を引き算を使って再帰的に定義せよ. たとえば, `rem3(5,Z)` は `Z=2` となって成功する. ただし, X を 0 以上の自然数とし, これ以外の入力はないものとする.

演習問題 (r2)

* のついている問題はオプション課題.

- (1) X を Y で割った時のあまりが Z である関係を表す述語 rem を引き算を使って再帰的に定義せよ. たとえば, $\text{rem}(5,3,Z)$ は $Z=2$ となって成功する. ただし, X, Y は 0 以上の自然数で, $Y \neq 0$ とする.
- (2) $\text{fact}(N,M)$ を M が N の階乗であるような関係を表すとするとき, fact を定義せよ. たとえば $\text{fact}(5,120)$ は成功し, $\text{fact}(5,X)$ は $X = 120$ を返す. ただし, N, M は 1 以上の自然数とする. また, $\text{fact}(X,120)$ は計算できなくてよい.
- (3) $\text{ssum}(N,M)$ を $M = 1 \cdot 2 + 2 \cdot 3 + \dots + (N-1) \cdot N$ を満たす関係を表すとするとき, ssum を定義せよ. ただし, N は 1 以上, M は 0 以上の自然数とする. (それ以外の入力については考慮する必要はない.) たとえば $\text{ssum}(10,330)$ は成功し, $\text{ssum}(10,X)$ は $X = 330$ を返す. ただし, N は自然数で, $N \geq 1$ とする. また, $\text{ssum}(X,330)$ は計算できなくてよい.
- (4) 第1回練習問題の親子関係において, X が Y の姪であるという関係を表す述語 $\text{niece}(X,Y)$ を定義せよ. データベースも一緒に提出すること.
- (5) $\text{edge}(N,M)$ を有向グラフにおいてノード N から ノード M への長さ 1 のエッジがあるという関係を表すとする. 図 2.1 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 $\text{dist}(N,M,L)$ を edge を用いて定義せよ. N, M, L を変数としたとき, 全解が得られることを確認せよ. (ただし経路が複数あるものはその数だけ同一解が得られる.)

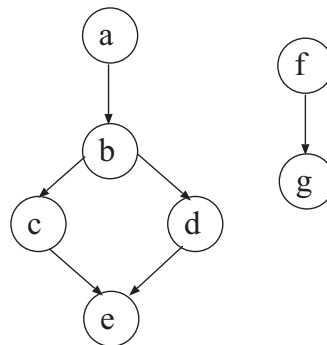


図 2.1

- (6)* s を自然数から自然数への関数とし, $s(x)$ は $x+1$ を表すものとする. このとき, 自然数を $0, 1, 2, \dots$ と表現する方法を EXP1 とし,

$$\begin{aligned}
 0 &\rightarrow 0 \\
 1 &\rightarrow s(0) \\
 2 &\rightarrow s(s(0)) \\
 3 &\rightarrow s(s(s(0))) \\
 &\vdots
 \end{aligned}$$

のように, $0, s(0), s(s(0)), \dots$ で表現する方法を EXP2 とする.

このとき, EXP1 から EXP2 への変換に相当する述語 $\text{convert}(\text{EXP1}, \text{EXP2})$ を定義せよ. たとえば $\text{convert}(3, P)$ は $P = s(s(s(0)))$ を返す.

- (7) 解答例 r1_7 についてレポートせよ. 以下を記述すること. (i) 論理的意味, (ii) 実際に動かしたときの動作, (iii) 自分が正しいプログラムができなかった場合は間違った点とその理由 (正しいプログラムができていた場合は「正しくできた」の一言でよい), (iv) 新たな知見や疑問点 (もしあれば).