演習問題 r2-(7) に訂正あり．ただし，すでに提出した人は特に変更しなくてもよい．[2024/04/19 17:46]

単一化 (unification)

- **単一化 (unification)**
  直観的には，述語 (ゴール) 同士の変数に矛盾のないような代入をして見た目をそろえること．Prolog を実行したときに起こる基本操作である．
- **head unification**
  ゴール節のゴールと確定節のヘッド部のゴールの間の単一化．この結果確定節のボディ部に単一化の結果が反映される．

例：r1　練習問題 1(3) の論理的意味と動作

```
% database
parent(tom,bob).
parent(tom,liz).
parent(pam,bob).
parent(bob,pat).
parent(pat,jim).

male(tom).
male(bob).
male(jim).
female(pam).
female(liz).
female(pat).

father(X,Y) :- parent(X,Y), male(X).
```

論理的意味
　X が Y の親で，かつ，X が男性ならば，X は Y の父である．
動作
　:- father(X,Y).　を実行すると，まず，parent(X,Y) を実行する．
　このゴールと parent の定義の一番上にある parent(tom,bob) と
　単一化が成功し，X=tom，Y=bob を得る．
　続いて male(tom) を実行すると，male(tom) がデータベースに書かれているので
　この単一化も成功し（変数がないので代入することなく一致）male(tom) というゴールも成功する．
　したがってすべてのボディゴールが成功するので，father(X,Y) が成功し，
　X=tom，Y=bob を最初の解として得る．
　（ ; をタイプすると，バックトラックして別解を探しにいく．）

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.

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，下記 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．

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 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 is Y というゴールに対しては， Y に具体的な数値がはいらなければ成功しない

`X = taro`	X は taro と単一化される
`X is taro`	これは誤り

- 論理的に正しくても実行系依存で正しく動作しない場合がある．
- Prolog が左から右，深さ優先で実行されることに注意し，どの変数が具体化されているのかを考えること
- 原則として，数値計算には is を使用して is の右辺はそれまでに値が具体化されているように書く．
- 項 (オブジェクト) として単一化したい場合は = を使用する．

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' という項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.
```

練習問題

(1) では，実行時のゴール節には必ず第１引数に常に定数がはいっていると考え，`?- a(X,8189).` のようなものを考える必要はない．さらに、特別な指示がなければ解は１個見つかったら終了する．今後の課題はすべてこの仮定でプログラムしてください．

1. $a_0 = 5, a_n = 2a_{n-1} + 3$ という漸化式で定義される数列があるとき、この数列の第 N 項が M であるという関係を表す述語 a(N,M) を再帰的に定義せよ．たとえば，a(10,Y) は Y=8189 となって成功する．ただし，N を 0 以上の自然数とし，これ以外の入力はないものと考える．

2. 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考える．

3. 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,Y) は Y=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考える．

演習問題 (r2)

$*$ のついている問題はオプション課題.

(1) 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考える．注意：0 の 0 乗は 1 である．

(2) 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，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．

(3) fact(N,M) が， 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 fact(5,120) は成功し，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，fact(X,120) は計算できなくてよい．

(4) ssum(N,M) が $M = 0 \cdot 1 + 1 \cdot 2 + 2 \cdot 3 + \ldots (N-1) \cdot N$ を満たす関係を表すとするとき，ssum を定義せよ．(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 ssum(10,330) は成功し，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，ssum(X,330) は計算できなくてよい．(Hint: M と $M-1$ の間の関係を考えよ．)

(5) edge(N,M) が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 dist を edge を用いて**再帰的に**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．)

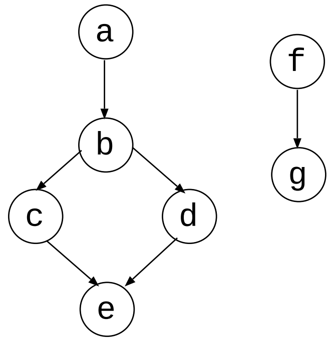

図 2.1

(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 $0, 1, 2 \ldots$ と表現する方法を EXP1 とし，

$$
\begin{array}{ll}
0 & \to 0 \\
1 & \to s(0) \\
2 & \to s(s(0)) \\
3 & \to s(s(s(0))) \\
 & \vdots
\end{array}
$$

のように，$0, s(0), s(s(0)), \ldots$ 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 convert(EXP1,EXP2) を定義せよ．たとえば convert(3,P) は P=s(s(s(0))) を返す．

(7) 練習問題 3 の解答例プログラムの論理的意味を示せ．**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．