

『論理と形式化』 Prolog 演習 の略解

亀山幸義 (kam[at]cs.tsukuba.ac.jp)

下記の演習を行い、レポートを作成して、1つのpdfファイルの形でmanabaシステムを通じて提出しなさい。(締切は別途連絡する。)

演習 1:

サンプルプログラムのうち、basic.plに含まれる様々な例題を試して、理解を深めなさい。

その上で、以下のことを意味するゴールを書き、実際に実行して、そのような解をすべて求めなさい。(ゴールと、それを実行して得た結果をレポートにはりつけてください。)

- $X + X = 4$ となる自然数 X が存在するか。

答.

```
?- add(X,X,s(s(s(s(0))))).  
X = s(s(0)) n  
false.
```

($X=2$) という解が得られた。

- $X * X = 4$ となる自然数 X が存在するか。

答.

```
?- times(X,X,s(s(s(s(0))))).  
X = s(s(0)) n  
(このあとは無限ループ)
```

($X=2$) という解が得られた。その後、2つ目の解を探そうとして(そんなものは本当は存在しないのだが、そのことを知らないPrologシステムは)無限ループになってしまった。

- $X + Y = 5$ かつ $X * Y = 6$ となる自然数 X, Y の組は存在するか。

答.

```
?- add(X,Y,s(s(s(s(s(0))))),times(X,Y,s(s(s(s(s(s(0))))))).  
X = s(s(0)),  
Y = s(s(s(0))) n  
X = s(s(s(0))),  
Y = s(s(0)) n  
false.
```

($X=2, Y=3$) と ($X=3, Y=2$) という解が得られた。

演習 1': 以下のプログラムは停止しない。なぜか考えよ。(考えるだけでよく、レポートとして提出することは求めない。)

```
?- times(X, s(s(0)), Z), times(Y, s(s(0)), s(Z)).
```

答. このゴールは、「 $2X=Z$ かつ $2X=Z+1$ となる自然数 X, Z はあるか？」という意味なので、もちろん解は存在しないのだが、Prologはこのゴールに対して有限的に失敗してくれず無限ループになる。

なぜかといえば、timesの定義を見ると、

```
times(0,_,0).
times(s(X),Y,Z) :- times(X,Y,W), add(W,Y,Z).
```

となっており、第 1 引数に関する帰納的定義になっている (times の再帰呼び出しにおいて、第 1 引数の値をどんどん減らしていく)。このため、このゴールのように第 1 引数に変数であると、いつまでも減らすことができしまい、停止しない。

演習 2:

basic.pl で定義されている add という述語に対して、以下のゴールを実行してすべての解を得なさい。(これらの解をレポートに書く必要はありません。)

```
?- add(X, Y, s(s(s(0)))).
```

これらの解が表示される順番を調べることにより、複数の解が存在するゴールに対して、Prolog 処理系がどのような順番で解を探索するかを推測して、書きなさい。(書きかたは特に問わない。また、あなた自身の「推測」で構わない。) 答. これは授業で説明した通り。(Prolog ではゴールの中に含まれる原子論理式を左から右に解こうとする。また、探索は「深さ優先」である。)

演習 3:

basic.pl のプログラムを参考にして、以下の Prolog のプログラムを書きなさい。また、それに対するテスト例 (そのプログラムを、具体的なゴールに対して起動して、結果がでたもの)3 つ程度作成し、それをレポートに書きなさい。

- power(X,Y,Z): 自然数 X, Y に対して、 $Z = X^Y$ (X の Y 乗) であることを意味する。

たとえば、power(s(s(0)),s(s(0)),s(s(s(s(0))))) が成立する。

答. 一例は次の通り。(どの引数についての帰納法かを考えるとよい。この関数 power のときは、Y に関する帰納法にするとよい。)

```
power(X,0,s(0)).
power(X,s(Y),Z) :- power(X,Y,W),times(W,X,Z).
```

- (発展課題) nthprime(X,Y): 自然数 X に対して、 Y が X 番目の素数であることを意味する。

たとえば、nthprime(s(s(0)),s(s(s(0))))、つまり、「2 番目の素数は 3 である」が成立する。

答. 一例は次の通り。(いろいろな書き方があるが、普通にと書くと、Prolog システムがあまりにも非効率な探索をするために、「5 番目の素数は何か?」という程度の goal も解いてくれない。以下のは、比較的効率がよく 10 番目ぐらいまでは楽々もとめてくれるプログラムである。)

```
lt(0,s(_)).
lt(s(X),s(Y)) :- lt(X,Y).
ge(_,0).
ge(s(X),s(Y)) :- ge(X,Y).
mod(X,Y,X) :- lt(X,Y).
mod(X,Y,Z) :- ge(X,Y),sub(X,Y,X1),mod(X1,Y,Z).
isNonDivisible(_,Y,Y).
isNonDivisible(X,Y,s(Z)) :- mod(X,Z,s(_)),isNonDivisible(X,Y,Z).
isPrime(s(s(X))) :- isNonDivisible(s(s(X)),s(s(0)),s(s(X))).
isDivisible(X,Y,s(Z)) :- lt(Y,s(Z)),mod(X,Z,0).
isDivisible(X,Y,s(Z)) :- isDivisible(X,Y,Z).
nextPrime(X,X) :- isPrime(X).
nextPrime(X,Y) :- isDivisible(X,s(s(0)),X),nextPrime(s(X),Y).
```

```
nthPrime(s(0),s(s(0))).
nthPrime(s(s(X)),W) :- nthPrime(s(X),s(s(Y))),nextPrime(s(s(s(Y))),W).
```

ここで $lt(X,Y)$ は $X < Y$ を、 $ge(X,Y)$ は $X \geq Y$ を、 $mod(X,Y,Z)$ は X を Y で割った余りが Z であることを意味する。 $isNonDivisible(X,Y,Z)$ は、 X が Y 以上 $Z-1$ 以下のすべての自然数で割り切れないことを、 $isDivisible(X,Y,Z)$ は、 X が Y 以上 $Z-1$ 以下のある自然数で割り切れることを意味する。 $isPrime(X)$ は X が素数であることを、 $nextPrime(X,Y)$ は X 以上の最小の素数が Y であることを、 $nthPrime(X,Y)$ は X 番目の素数が Y であることを意味する。

実行例は以下の通り。

```
% ?- nthPrime(s(0),X).
% X = s(s(0))
%
% ?- nthPrime(s(s(0)),X).
% X = s(s(s(0)))
%
% ?- nthPrime(s(s(s(0))),X).
% X = s(s(s(s(s(0)))))
%
% ?- nthPrime(s(s(s(s(0)))),X).
% X = s(s(s(s(s(s(0))))))
%
% ?- nthPrime(s(s(s(s(s(0))))),X).
% X = s(s(s(s(s(s(s(s(0)))))))
%
% ?- nthPrime(s(s(s(s(s(s(s(s(0)))))))),X).
% |      X = s(s(s(s(s(s(s(s(s(s(0))))))))) w
% X = s(s(s(s(s(s(s(s(s(s(s(s(s(s(s(s(s(s(s(s(s(s(s(0))))))))))))))))))
% (10th prime is 29)
% typing "w" lets the system to print the whole term
```

最後の例は、答えの項が大き過ぎたためにシステムが「...」と省略してしまったものである。SWI-Prolog では「w」とタイプすると、答えの項全体を表示してくれる。上記の応答は、「10 番目の素数が 29 である」ことを表している。

(補足) 上記では、 $isDivisible$ と $isNonDivisible$ という 2 つの述語を独立に定義したが、Prolog には not (否定) があるので、 $isNonDivisible$ を not と $isDivisible$ で定義することも可能ではある。