Ejercicios de Programación Declarativa

Curso 2020/21

Soluciones Hoja 6

1. Supongamos que no utilizamos la aritmética de Prolog, sino que los números naturales se representan mediante la constance c o mediante la aplicación de una función s de aridad uno aplicada a un natural. Es decir, un predicado $nat(X)$ para comprobar si un término representa un número natural sería:

```
nat(c).
nat(s(X)) :- nat(X).
```

   (a) Representa el árbol de resolución del objetivo `nat(X)` hasta obtener tres éxitos.

   (b) Escribe un programa Prolog para implementar los siguientes predicados:
   $sum(X, Y, Z) \longleftrightarrow X, Y, Z$ son números naturales, es decir satisfacen la relación `nat`, $Z = X + Y$.

```
sum(X,c,X) :- nat(X).
sum(X,s(Y),s(Z)) :- sum(X,Y,Z).
```

   $prod(X, Y, Z) \longleftrightarrow X, Y, Z$ son números naturales, $Z = X \times Y$.

```
prod(X,c,c) :- nat(X).
prod(X,s(Y),Z) :- prod(X,Y,Z1), sum(X,Z1,Z).
```

   $pot(X, N, Y) \longleftrightarrow X, N, Y$ son números naturales, $X \neq 0, X^N = Y$.

```
pot(s(X),c,s(c)) :- nat(X).
pot(s(X),s(N),Y) :- pot(s(X),N,Z), prod(s(X),Z,Y).
```

   $fact(X, Y) \longleftrightarrow X, Y$ son números naturales, $X! = Y$.

```
fact(c,s(c)).
fact(s(X),Y) :- nat(X), fact(X,Z), prod(s(X),Z,Y).
```

   $fib(N, Y) \longleftrightarrow N, Y$ son números naturales, Y es el N-ésimo número de Fibonacci.

```
fib(c,s(c)).
fib(s(c),s(c)).
fib(s(s(X)),Y) :- fib(s(X),Z1), fib(X,Z2), sum(Z1,Z2,Y).
```

   (c) Define los predicados anteriores, pero utilizando recursión final si no lo has hecho antes.

PRODUCTO

```
prod(X,Y,Z) :- prod(X,Y,c,Z).
prod(c,Y,Ac,Ac).
prod(s(X),Y,Ac,Z) :- sum(Ac,Y,NAc), prod(X,Y,NAc,Z).
```

POTENCIA

```
pot(s(X),Y,Z) :- pot(s(X),Y,s(c),Z).
pot(s(X),c,Ac,Ac).
pot(s(X),s(N),Ac,Z) :- prod(s(X),Ac,NAc), pot(s(X),N,NAc,Z).
```

FACTORIAL

```
fact(X,Y) :- fact(X,s(c),Y)
fact(c,Ac,Ac).
fact(s(X),Ac,Y) :- prod(s(X),Ac,NAc), fact(X,NAc,Y).
```

FIBONACCI

```
fib(X,Y) :- fib(X,c,s(c),Y).
fib(c,Ac1,Ac2,Ac2).
fib(s(X),Ac1,Ac2,Y) :- sum(Ac1,Ac2,Ac3), fib(X,Ac2,Ac3,Y).
```

2. Considera la definición recursiva final del producto de naturales del ejercicio anterior. Determina el árbol de búsqueda para el objetivo:

   ?− `prod(X,s(s(c)),s(s(s(s(c))))).`

3. Escribe un programa Prolog con recursión final para hallar los polinomios de Fibonacci, con la siguiente especificación:

   $polfib(N, X, PF) \longleftrightarrow PF$ es el valor del polinomio de Fibonacci de grado N para el número natural X.

   Esto es: $PF = a_o + a_1X + a_2X^2 + \ldots + a_NX^N$. Donde a_i es el i-ésimo número de Fibonacci.

```
pol_fib(c,X,s(c)) :- nat(X).
pol_fib(s(N),X,M) :- pol_fib(N,X,M1), fib(s(N),M2), pot(X,s(N),M3),
      prod(M2,M3,M4), sum(M1,M4,M).
```

   Versión recursiva final:

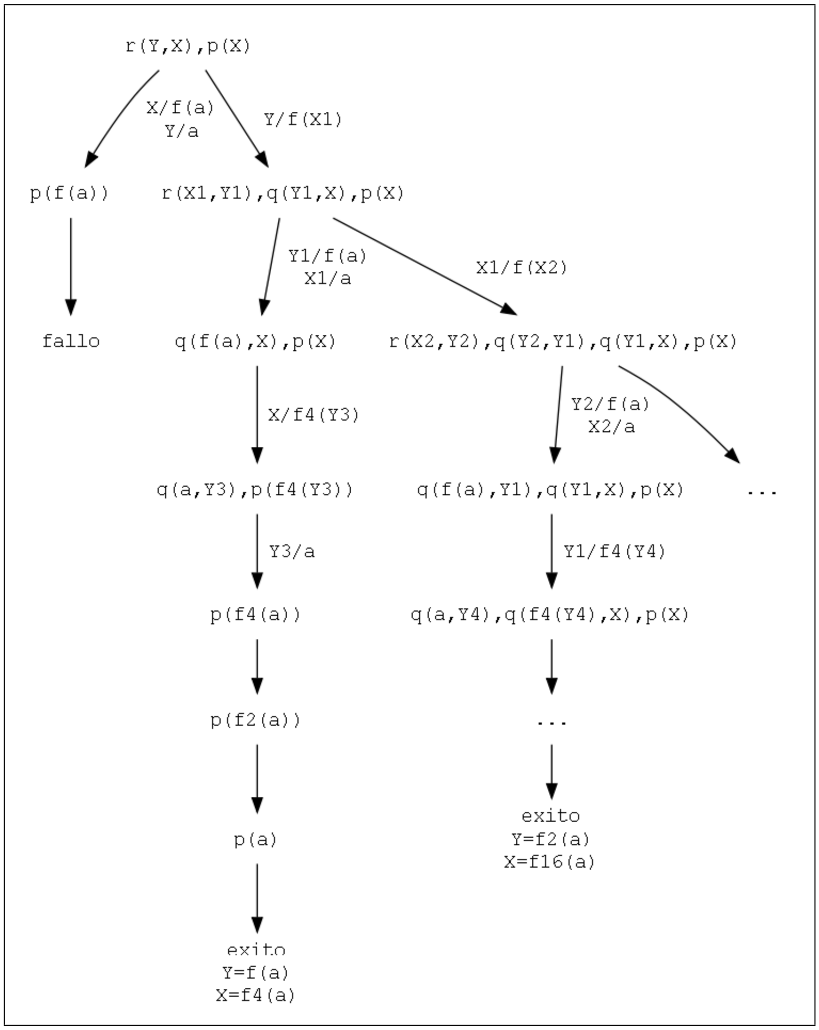
```
pol_fib(c,X,s(c)) :- nat(X).
pol_fib(s(N),X,M) :- pol_fib(s(N),X,s(c),s(c),c,s(c),M).

pol_fib(c,X,Act,Acupot,Acu1,Acu2,Act).
pol_fib(s(N),X,Act,Acupot,Acu1,Acu2,P) :- sum(Acu1,Acu2,Acu3),
      prod(Acupot,X,Acupot1), prod(Acu3,Acupot1,Acu4),
      sum(Act,Acu4,NAct),
      pol_fib(N,X,NAct,Acupot1,Acu2,Acu3,P).
```

4. Sea P el programa definido mediante las siguientes cláusulas:

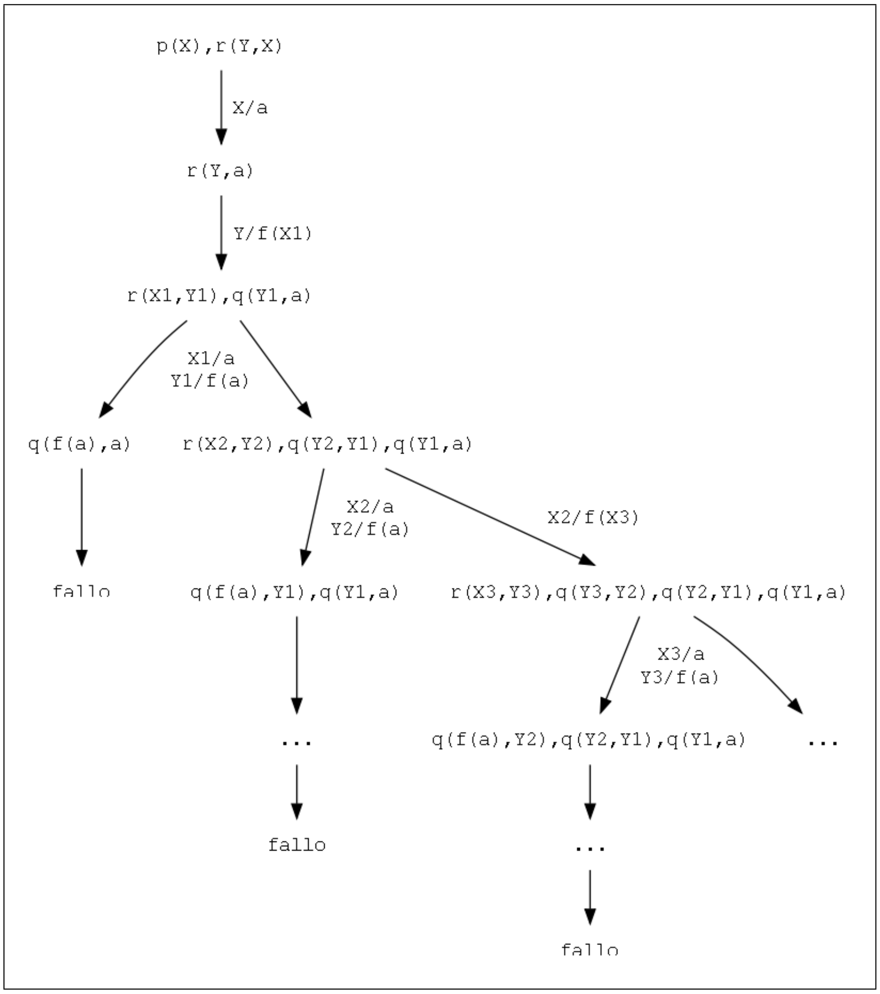
```
p(a).
p(f(f(X))) :- p(X).
q(a,a).
q(f(X),f(f(f(f(Y))))) :- q(X,Y).
r(a,f(a)).
r(f(X),Z) :- r(X,Y), q(Y,Z).
```

   (a) Computa los siguientes objetivos siguiendo la estrategia de Prolog hasta conseguir dos éxitos si es posible.

   ?− `r(Y, X), p(X).`


r(Y,X),p(X)
X/f(a)
Y/a
Y/f(X1)
p(f(a))
r(X1,Y1),q(Y1,X),p(X)
Y1/f(a)
X1/a
X1/f(X2)
fallo
q(f(a),X),p(X)
r(X2,Y2),q(Y2,Y1),q(Y1,X),p(X)
X/f4(Y3)
Y2/f(a)
X2/a
q(a,Y3),p(f4(Y3))
q(f(a),Y1),q(Y1,X),p(X)
...
Y3/a
Y1/f4(Y4)
p(f4(a))
q(a,Y4),q(f4(Y4),X),p(X)
p(f2(a))
...
exito
Y=f2(a)
X=f16(a)
p(a)
exito
Y=f(a)
X=f4(a)

?– p(X), r(Y, X).



(b) ¿Qué significado tendrían los predicados de este programa y estos objetivos si a fuera la constante 0 y f la función sucesor de los naturales?

$p(X) \leftrightarrow X$ es un número par.

$q(X, Y) \leftrightarrow Y = X \times 4$.

$r(X, Y) \leftrightarrow Y = 4^X$.