Ejercicios de Programación Declarativa

Curso 2019/20

Hoja 5

2. 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).
```

Escribe un programa Prolog para implementar los siguientes predicados:

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

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

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

Puedes escribir predicados para otras operaciones previas que te sean útiles.

Los siguientes predicados son necesarios para definir $pot(X, N, Y)$, $fact(X, Y)$ y $fib(N, Y)$.

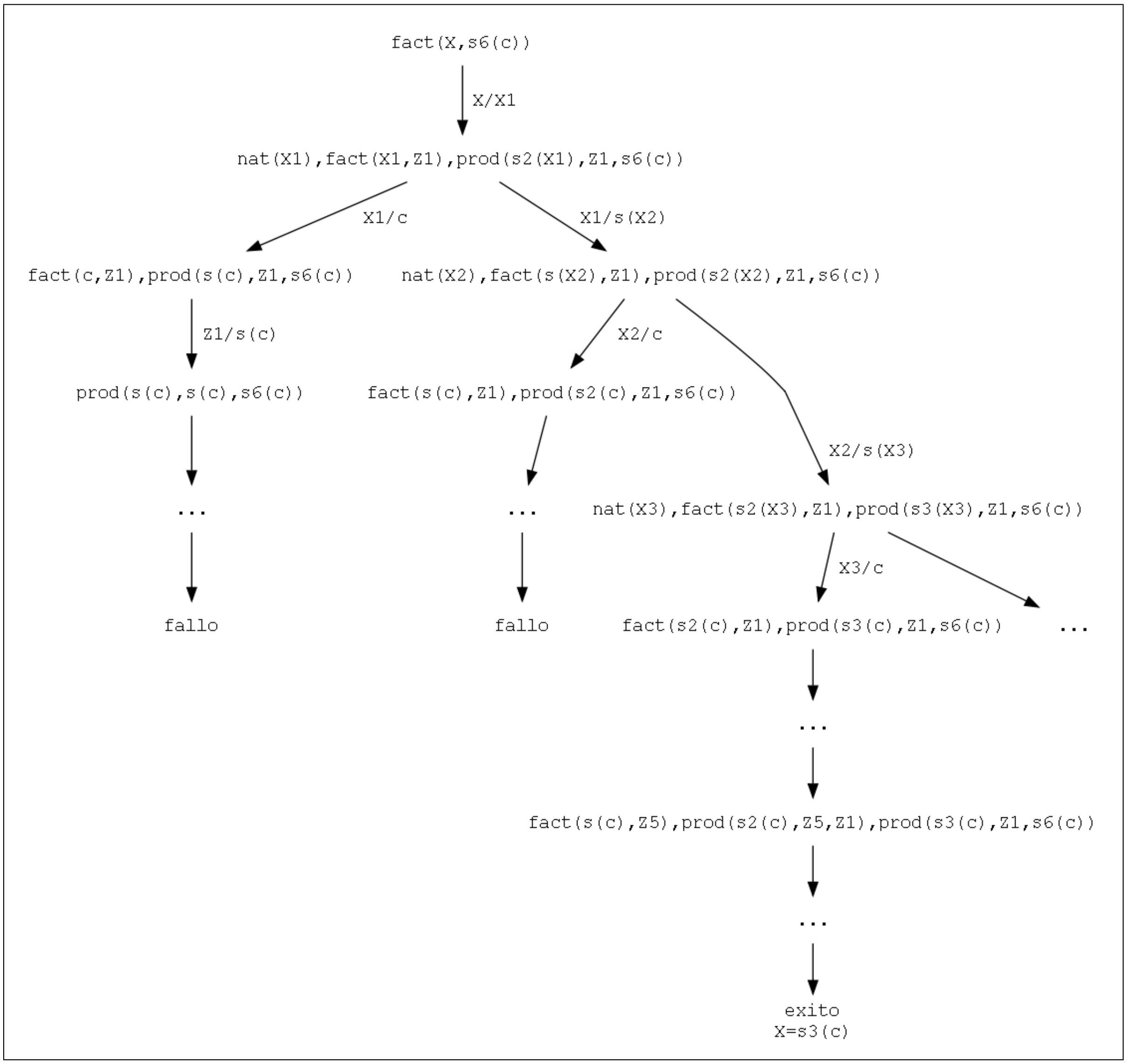
```
sum(X,c,X).
sum(X,s(Y),s(Z)) :- sum(X,Y,Z).

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).

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

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).
```

Con recursión final:

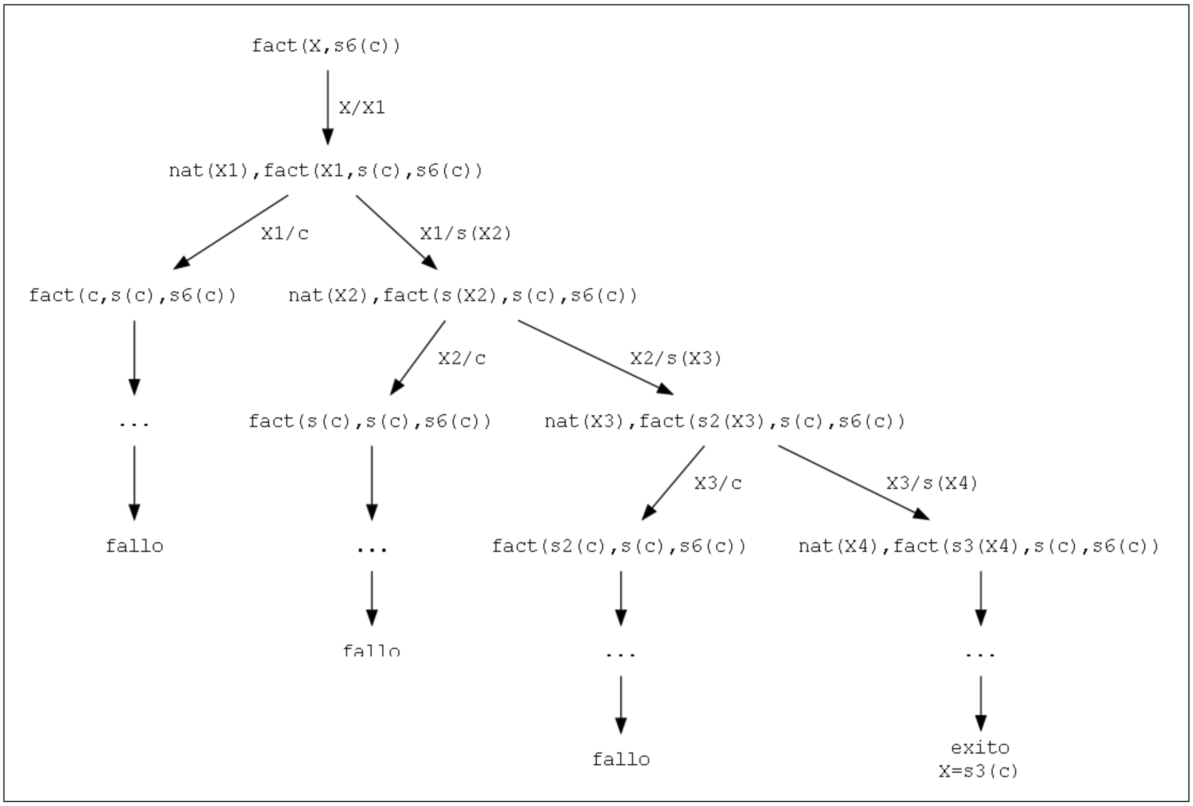
```
fib(X,Y) :-
    nat(X),
    fib(X,c,s(c),Y).

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

3. Considera el programa recursivo del factorial de naturales definido en el ejercicio anterior. Determina el árbol de búsqueda para el objetivo:

   ?− `fact(X, s(s(s(s(s(s(c))))))).`



Repite el ejercicio, pero utilizando ahora definiciones con recursión final para el producto y el factorial.



4. 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:

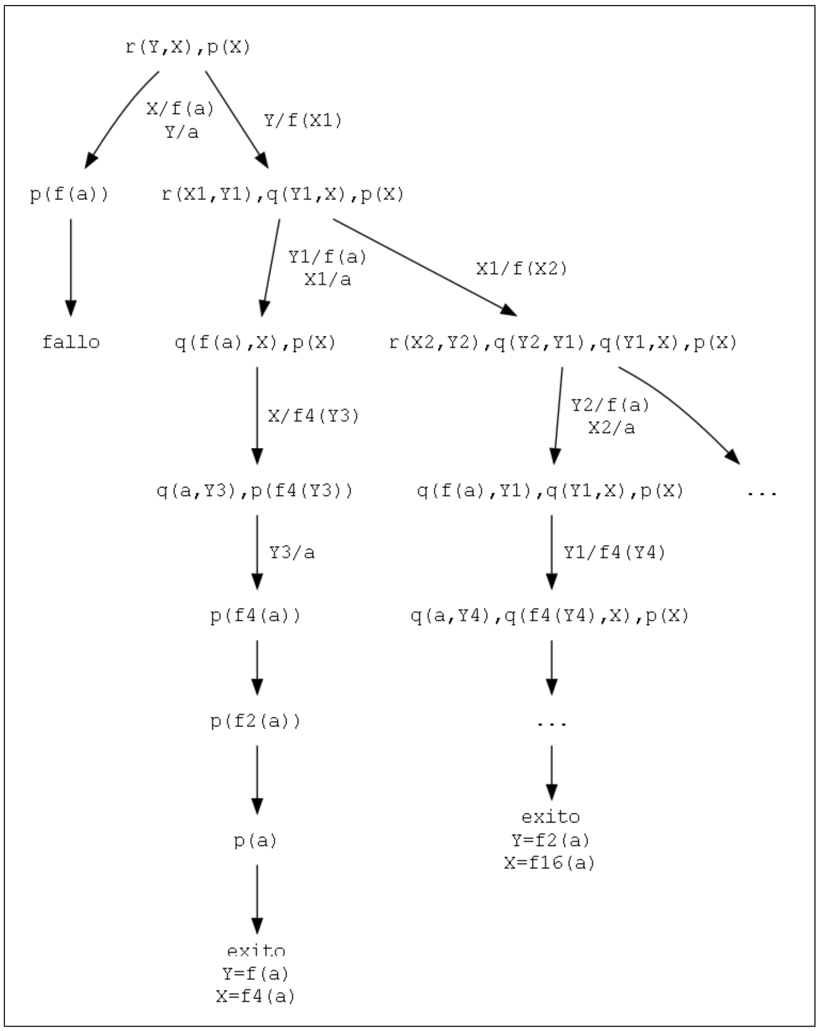
```
pol_fib(c,X,s(c)) :- nat(X).
pol_fib(s(N),X,M) :- pol_fib(s(N),X,s(c),s(c),c,s(c),M).

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

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

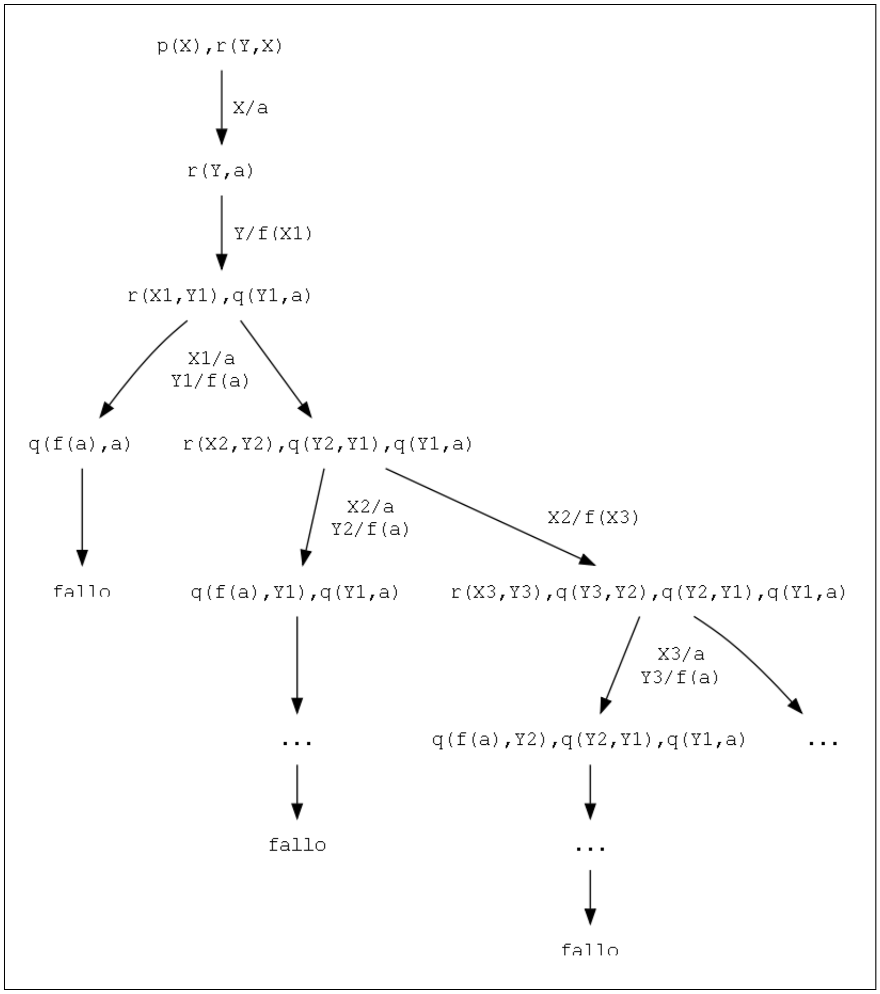
```
p(a).
p(f(f(X))) :- p(X).
q(a,a).
q(f(X),f(f(f(f(Y))))) :- q(X,Y).
r(a,f(a)).
r(f(X),Z) :- r(X,Y), q(Y,Z).
```

(a) Computar los siguientes objetivos siguiendo la estrategia de Prolog hasta conseguir dos éxitos y comparar los espacios de búsqueda producidos.

?– `r(Y, X), p(X).`



?– 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.
$q(X, Y) \leftrightarrow Y = X \times 4$.
$r(X, Y) \leftrightarrow Y = 4^X$.