

PROGRAMACIÓN DECLARATIVA

3^{er} Curso, Grado en Ingeniería en Informática
Universidad Rey Juan Carlos

Programación Lógica

Construcción detallada de un Árbol de Resolución

Considere el programa Prolog

```
p(X,X).                % cláusula C1 (hecho)
p(X,Z) :-              % cláusula C2 (regla)
    p(Y,Z),
    q(X,Y).
q(a,b).                % cláusula C3 (hecho)
```

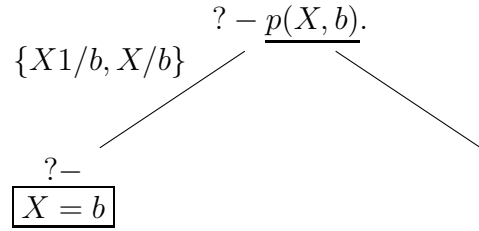
y la consulta $?- p(X,b)$, cuyo objetivo es averiguar, a partir del conocimiento expresado en el programa anterior, si existe algún objeto X relacionado con el objeto b mediante la relación p . En lo que sigue se detalla la construcción, paso a paso, del árbol de Resolución correspondiente a la consulta anterior y se explica cómo, a partir de él, se deduce qué respuesta(s) ofrecería Prolog ante esa consulta, y en qué orden facilitaría dichas respuestas.

Construcción del Árbol de Resolución

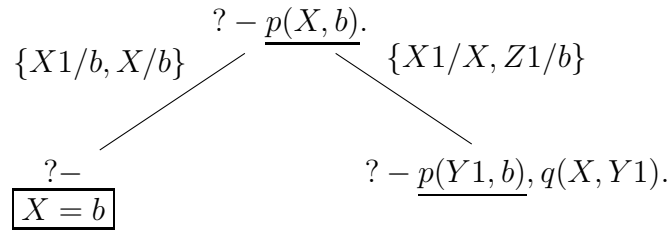
Los árboles de Resolución de Prolog, descritos en el apartado 3.2. del tema PL-2 de los apuntes, se construyen en profundidad por la izquierda y con retroceso. La raíz del árbol no es otra cosa que la consulta, que tendrá tantos hijos como cláusulas haya en el programa con las que se pueda aplicar la Regla de Resolución, es decir, cláusulas cuya cabeza unifique con $p(X,b)$ (en general, con el sub-objetivo de más a la izquierda de la consulta). Este objetivo unifica tanto con la cabeza de la cláusula C1 como con la cabeza de la cláusula C2, aunque será necesario renombrar ambas ya que hay conflicto con los nombres de las variables (la variable X aparece tanto en el objetivo como en C1 y en C2). Dado que Prolog desarrolla el árbol en profundidad por la izquierda, en este momento sólo se calcula el nodo correspondiente a la rama de más a la izquierda, la producida por C1 (aunque se deja trazada la otra rama para cuando haya que desarrollarla al retroceder en la construcción del árbol). Aplicando el algoritmo de unificación entre el objetivo a resolver, $p(X,b)$, y la cabeza de C1 renombrada, $p(X1,X1)$, el unificador de máxima generalidad (u.m.g.) resultante es $\{X1/b, X/b\}$, con el que se obtiene la cláusula vacía $?-$ como cláusula resolvente. Se ha llegado así a un nodo éxito, por lo que Prolog calcula la solución asociada aplicando a cada una de las variables de la consulta todos los u.m.g.'s de la rama, en orden, empezando desde la raíz. En este caso solo hay una variable, X , a la que se aplica el único u.m.g. de su rama:

$$X\{X1/b, X/b\} = b$$

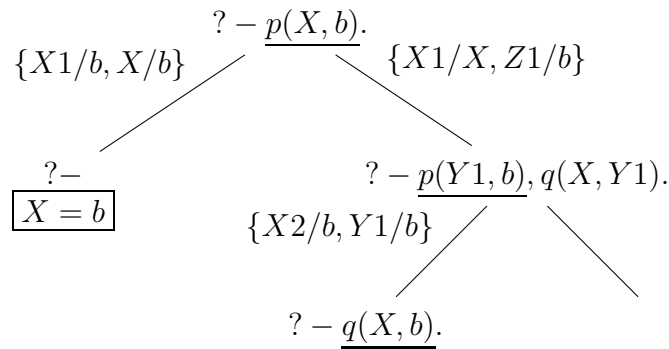
Por lo tanto, el árbol de Resolución es, por el momento, el siguiente:



Dado que se ha terminado de desarrollar la rama de más a la izquierda, Prolog retrocede hasta el siguiente nodo pendiente de cálculo, el segundo hijo de la raíz, que se obtiene aplicando la Regla de Resolución entre el objetivo $?- p(X, b)$ y la regla C2 renombrada, $p(X1, Z1) :- p(Y1, Z1), q(X1, Y1)$, mediante el u.m.g. $\{X1/X, Z1/b\}$ (el otro u.m.g. posible sería $\{X/X1, Z1/b\}$, pero da lo mismo elegir uno que otro), que da lugar a la cláusula resolvente $?- p(Y1, b), q(X, Y1)$, completándose así el primer nivel del árbol de Resolución:

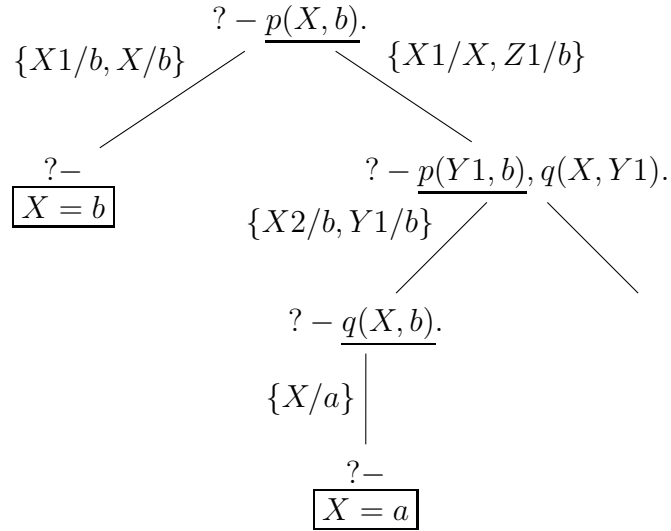


En el siguiente nivel del árbol habrá dos nodos, puesto que el sub-objetivo de más a la izquierda, $p(Y1, b)$, unifica tanto con la cabeza de C1 como con la de C2. Por el momento solo se desarrolla el hijo izquierdo (con C1), que, con u.m.g. $\{X2/b, Y1/b\}$, da lugar a la cláusula resolvente $?- q(X, b)$. Obsérvese que se ha renombrado la cláusula C1 (con subíndice 2 puesto que ahora estamos en el segundo nivel del árbol) debido a que su variable X ya está presente en algunos nodos del árbol.

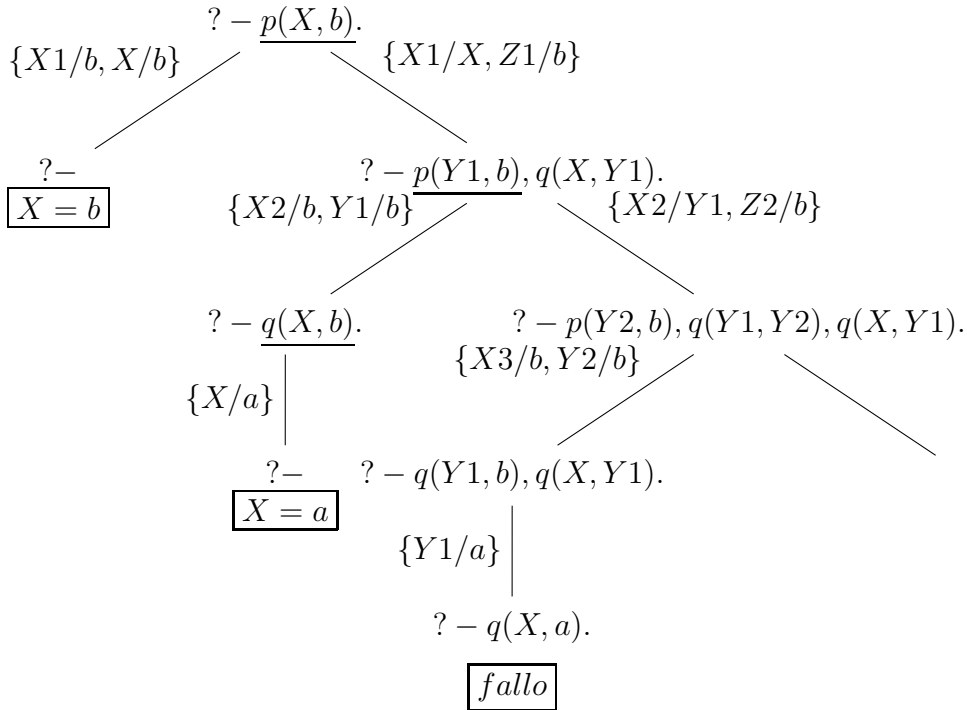


Siguiendo el desarrollo del árbol en profundidad por la izquierda, el nuevo nodo a expandir es $?- q(X, b)$, que tiene un único hijo con C3 mediante el u.m.g. $\{X/a\}$. La cláusula resolvente es la cláusula vacía, por lo que se trata de un nuevo nodo éxito, con solución asociada

$$X \{X1/X, Z1/b\} \{X2/b, Y1/b\} \{X/a\} = a$$



El siguiente nodo que hay que calcular es el segundo hijo del nodo $?- p(Y1,b), q(X,Y1)$, que, unificando $p(Y1,b)$ con la cabeza de la regla C2 (renombrada) mediante el u.m.g. $\{X2/Y1, Z2/b\}$, da lugar a la cláusula resolvente $?- p(Y2,b), q(Y1,Y2), q(X,Y1)$. Esta última, a su vez, tiene dos hijos, obtenidos unificando $p(Y2,b)$ con C1 (hijo izquierdo) y con C2 (hijo derecho). Como siempre, Prolog desarrolla primero el hijo izquierdo, obteniendo el resolvente $?- q(Y1,b), q(X,Y1)$ mediante el u.m.g. $\{X3/b, Y2/b\}$, y a partir de él el nuevo resolvente $?- q(X,a)$ (haciendo la sustitución $\{Y1/a\}$), que falla puesto que no hay ninguna cláusula en el programa que lo resuelva. El árbol es ahora el siguiente:

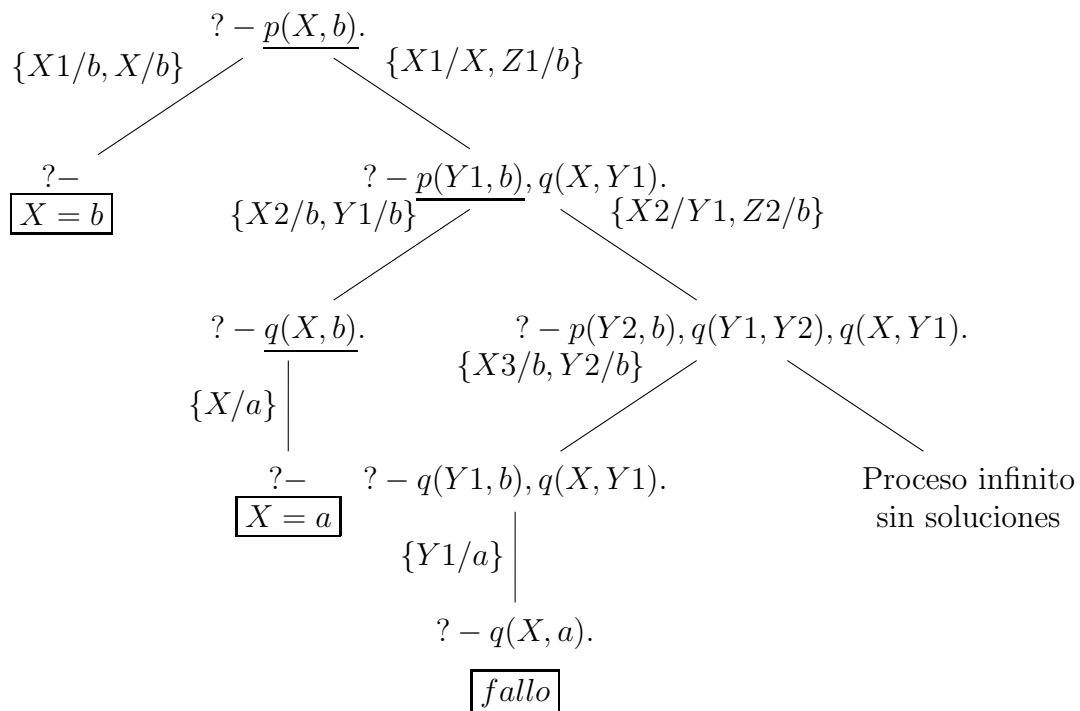


Al encontrarse con un fallo, Prolog retrocede automáticamente hasta el siguiente nodo pendiente de cálculo, que resultará ser $?- p(Y3,b), q(Y2,Y3), q(Y1,Y2), q(X,Y1)$. Llegados a este punto no merece la pena seguir desarrollando el árbol, puesto que se

aprecia que se trata de un proceso infinito. En efecto, el nuevo nodo tendrá dos hijos, de forma que:

- La rama de la izquierda, al cabo de unos pasos, acabará con un nodo fallo. Esto es debido a que los dos últimos sub-objetivos de la consulta, $q(Y1, Y2)$, $q(X, Y1)$, siempre darán lugar a un fallo, ya que $q(Y1, Y2)$ solo puede ser cierto mediante la sustitución $Y1/a$, sustitución que necesariamente hace imposible que se cumpla el siguiente sub-objetivo, ya que este pasaría a ser $q(X, a)$.
- La rama de la derecha repetirá el proceso indefinidamente, ya que la regla C2 es recursiva por la izquierda y vuelve a poner en cabeza del objetivo un nuevo predicado p .

En definitiva, el árbol de Resolución correspondiente al programa y a la consulta dados es el siguiente:



Respuestas de Prolog

Ante una consulta, Prolog construye el árbol de Resolución en profundidad por la izquierda, con retroceso, y facilita las soluciones según las va encontrando. Por lo tanto, ante la consulta dada, y a la vista del árbol anterior, la respuesta de Prolog será la siguiente:

```

X=b ;                % primera solución encontrada
X=a ;                % segunda solución encontrada
ERROR: Out of local stack % computación infinita
  
```

© 2022 Ana Pradera Gómez

Algunos derechos reservados

Este documento se distribuye bajo la licencia “Atribución-CompartirIgual 4.0 Internacional” de Creative Commons, disponible en

<https://creativecommons.org/licenses/by-sa/4.0/deed.es>