Mecanismo de obtención de respuestas de Prolog

Esta sección pretende dar una descripción informal del procedimiento utilizado por Prolog para responder a las preguntas, generalmente denominado reducción de metas, porque finaliza cuando se genera la meta vacía.

Para reducir una meta, Prolog intenta demostrar que la meta es una consecuencia lógica de las cláusulas del programa. Si la meta contiene variables, Prolog proporciona los elementos del dominio que la hacen cierta, según el programa.

Para ello, Prolog:

- Selecciona un literal de la meta.
- Intenta unificarle con las cabezas de las cláusulas del programa, en el orden dado por éste
  - Si unifica con un hecho, simplemente elimina el literal seleccionado de la meta, almacenando la substitución encontrada
  - Si unifica con una regla, reemplaza el literal de la meta por el cuerpo de la regla, considerando la substitución encontrada
  - Si no unifica con la cabeza de ninguna cláusula, se produce un fallo en la búsqueda

Prolog realiza una búsqueda primero en profundidad, implementada mediante **backtracking** para reducir las metas. El orden de la búsqueda viene dado por el orden de las cláusulas. Además, los literales de la meta se reducen por orden, seleccionando siempre el primero por la izquierda.

Si nuestro programa es:

```
padre(juan, antonio).
padre(juan, luis).
padre(luis, pedro).
abuelo(X, Y) :- padre(X, Z), padre(Z, Y).
```

y presentamos la pregunta `?-abuelo(juan, X).` , la secuencia de metas que se van generando es:

1. abuelo(juan ,G1), que es la propia pregunta renombrando las variables.

2. padre(juan,Z1), padre(Z1,Y1)

   obtenida al reducir `abuelo(juan ,X)` con `abuelo(X1, Y1) :- padre(X1, Z1), padre(Z1, Y1)`, donde se han renombrado las variables de la 4ª cláusula (X1,Y1,Z1) y se realizan las substituciones `juan/X1, Y1/G1`.

3. padre(antonio, Y1)

   obtenida al reducir padre(juan, Z1) con `padre(juan, antonio)`, obteniéndose la substitución `antonio/Z1`, que se aplica al resto de la meta.

4. padre(juan, Z1), padre(Z1, Y1)

   esta meta se obtiene por que la meta anterior (3.) falla. Eso hace que Prolog recupere la meta anterior (bactracking). Al mismo tiempo, Prolog descarta la primera cláusula para reducir esta meta, pues ya la ha utilizado y ha obtenido un fallo.

5. padre(luis, Y1)

   obtenida al reducir padre(juan,Z1) con `padre(juan, luis)` , con el unificador `luis/Z1`.

6. meta vacía

   obtenida al reducir padre(luis,Y1) con `padre(luis, pedro)` , con la sustitución `pedro/Y1` .

La representación gráfica de la secuencia de metas, las cláusulas utilizadas y las sustituciones empleadas ilustra con más claridad el tipo de búsqueda que realiza Prolog:

La composición de las substituciones obtenidas (concretamente `pedro/Y1` y `Y1/G1` pues las demás no se necesitan) proporciona `pedro/X` (recordar que X se renombró con G1), que es la respuesta que daría Prolog.

