

Programación Lógica

Consulta 1 : Backtracking y Unificación

Héctor F. JIMÉNEZ S.
hfjimenez@utp.edu.co
PGP KEY ID: 0xB05AD7B8

Kevin Alexander MORENO H.
kevinamh24@gmail.com

Fecha de Entrega: 21 de Febrero, 2017
Profesor: Cesar Jaramillo

1 OBJETIVOS

Consultar sobre *que como* y *para que* se utiliza :

1. Backtracking
2. Unificación

1.1 BACKTRACKING

El *backtracking* es una técnica especial utilizada en algunos paradigmas de programación con el fin de realizar una serie de pasos recursivos sobre un conjunto de posibles soluciones p , donde la idea es encontrar la mejor combinación posible en un momento determinado, algunos consideran que este tipo de técnicas tienen cuestiones prácticas sobre grafos, ya que al buscar de manera recursiva una solución, y encontrar fallo se dice que se construye un grafo dirigido, a este también se le considera como algoritmo un uso especial sobre búsqueda en profundidad. Durante la búsqueda, si se encuentra una alternativa incorrecta, la búsqueda de la solución debe retroceder, devolverse hasta el paso anterior y tomar la siguiente alternativa para buscar otra solución, cuando se han terminado las posibilidades intentadas del conjunto de soluciones p , se vuelve a la elección anterior y se toma la siguiente opción como se menciona anteriormente, pero si no hay más alternativas la búsqueda falla por que no existe una posible solución.

Para hacer uso de backtracking típicamente se utiliza recursividad en un conjunto de soluciones, el backtracking se usa en problemas en los que se tienen muchas soluciones y se desea encontrarlas, hay muchos ejemplos de problemas aplicables como encontrar la salida en un laberinto, resolver sudokus dinámicamente, el problema de las 8 reinas, entre otros son aplicaciones del backtracking.

1.2 UNIFICACIÓN PROGRAMACIÓN LÓGICA

Como sabemos la programación lógica en Prolog consiste en un conjunto de sentencias y hechos que son afirmaciones verdaderas de la forma

$$A : -B$$

, esto implica que el hecho **A** es cierto si es cierto el hecho **B**. Ahora estos hechos y sentencias se convierten en nuestro conjunto de reglas que le permitirán resolver ciertas incógnitas o dudas que se le presenten a nuestro programa de Prolog, para ello el hace uso de dos conceptos :

- Unificación
- Resolución

Nosotros para esta tarea hablaremos sobre la unificación; como sabemos las instrucciones se ejecutan normalmente en orden secuencial, es decir, una a continuación de otra, en el mismo orden en que están escritas, así que el interprete de Prolog es el que va resolviendo y verificando los hechos, y sentencias. El proceso de unificación intenta casar un predicado con otro, es decir comprobar si son absolutamente iguales, cuando es posible hacer sustituciones, éstas se realizan de manera que los predicados que se están unificando se tornen completamente iguales y proporcionen un resultado de ÉXITO. La unificación se realiza, para cada predicado, de izquierda a derecha, y para cada conjunto de predicados, de arriba a abajo. Se pueden unificar variables con constantes, siempre que la variable no esté instanciada. Si la variable está instanciada, el hecho de hacer unificación entre ambas se corresponde con la situación de unificación de dos constantes. Para que dos constantes se puedan unificar ambas han de ser iguales. Esta funcionalidad, unida a la posibilidad de crear términos complejos con estructuras anidadas, resulta un mecanismo muy potente y la principal ventaja a la hora de utilizar Prolog. En Prolog, decimos que dos términos distintos `termino1` y `termino2` unifican en los siguientes casos:

1. si `termino1` y `termino2` son constantes unifican si y solo si son el mismo átomo o el mismo número.
2. si `termino1` es una variable y `termino2` es cualquier tipo de término unifican, y además, `termino1` se instancia a `termino2`. En sentido contrario funciona exactamente igual. Y si ambos términos son variables, unifican porque cada uno se instancia con el otro y comparten valores

1.3 BIBLIOGRAFÍA

1. Introduccion a Prolog. <http://www.sc.ehu.es/jiwhehum2/prolog/Temario/Tema1.pdf>
2. Prolog: unificación y recursión, MRC <http://vitojph.github.io/ling/mrc/Prolog-Unificacion-Recursion.pdf>
3. Prolog: Lenguaje y Conceptos http://www.uhu.es/nieves.pavon/pprogramacion/temario/tema1/tema1.html#_Toc495042877