Lógica Computacional: Practica 6

José Miguel Toledo Reyes
Omar Fernando Gramer Muñoz

9 de junio de 2020

1. Realización

La práctica se realizó en conjunto entre los dos miembros del equipo, esto con el fin de estandarizar el estilo del código y mantener una mejor coherencia en el formato de la práctica.

El algoritmo implementado en Prolog verifica si las entradas del tablero de 9x9 recibidas son una solución válida del sudoku. Si faltan entradas del tablero entonces prolog realizará una búsqueda para encontrar las posibles soluciones (sí es que las hay).

El algoritmo recibe como entrada una lista llamada `Filas` la cual consiste de listas de longitud 9 que representan las entradas del tablero (como si se tratase de un arreglo bidimencional de 9x9).

1. Se verifica que efectivamente la entrada recibida sea un tablero de 9x9.
2. Se unen todas las entradas de las filas en una única lista llamada `Tablero`, la cual es utilizada para establecer que el dominio de los elementos del tablero deben ser números naturales entre el 1 al 9.
3. Se hace uso de las funciones de prolog `maplist` y `all_distinct` para verificar que en cada lista perteneciente a `Filas` no haya elementos repetidos. Con ello se verifica que no existan elementos repetidos en las filas del tablero.
4. Posteriormente se usa otra función de prolog llamada `transpose`, esta función recibe una lista de listas y con ellas calcula la 'matriz' transpuesta, la cúal regresa como otra lista de listas. En el algoritmo esta función recibe como entrada a `Filas` y asigna el resultado a `Columnas`.
5. Análogo al paso 3, se verifica que cada lista perteneciente a `Columnas` no tenga elementos repetidos y con ello se verifica que no existan elementos repetidos en las columnas del tablero.
6. Finalmente se identifica a cada fila con nombres `F1, F2, ... , F9` .Una vez identificadas se llama tres veces a la función bloques, esta función debe recibir como entrada tres filas contiguas y verifica que para cada cuadrante de 3x3 pertenecientes a ese sector de filas no hayan elementos repetidos.

2. Ejecución del programa

Para poder compilar el programa es necesario tener instalado swi-prolog compiler, el cual puede obtenerse en la siguiente liga..

```
https://www.swi-prolog.org/Download.html
```

Una vez hecho lo anterior, para correr el programa, es necesario establecerse el directorio de trabajo en la carpeta en donde se encuentra el archivo sudoku.pl y ejecutar el comando:

```
swipl sudoku.pls
```

Esto hará que se compile el programa y sea posible su ejecución. Para ejecutar el programa solo basta escribir:

```
?-  sudoku (_,_,_,_,_,_,_,_,.....).
```

Como parametros se recibiran un total de 81 valores los cuales representaran a las 81 casillas que posee el sudoku de 9x9. En los parametros se puede especificar el valor de las casillas de sudoku o ignorarlo y esperar a que prolog nos devuelva un posible valor,esto se hace escribiendo el simbolo _.

Con la entrada recibida,prolog tratará de darnos una posible solución para el sudoku, si fue posible encontrarla se nos regresara como respuesta un tablero de 9x9 en donde se indicara el valor que debe de tomar cada casilla para que cumpla con las condiciones del juego, de otro modo se nos regresará false.

3. Conclusiones

Esta práctica nos ayudó a conocer mas a fondo las características del lenguaje Prolog, y a entender con mas detalle como es que funciona la programación declarativa.

En general resolver un sudoku con prolog fue de cierta manera sencillo ya que al utilizar las tecnicas de la recusión y el backtracking.Nos ahorro la tarea de estar analizando cada posible solución que podia llegar a tener el problema y revisar si es que cumplia o no con las condiciones que tiene un sudoku,prolog directamente se encarga de hacerlo usando las condiciones que le definimos en un principio, y en base a lo que fuese decidiendo nos dará una respuesta dependiendo de la cantidad y la correctud de la informacion que se le proporcione.