

Lógica Computacional: Practica 5

José Miguel Toledo Reyes

11 de mayo de 2020

1. Realización

Para la realizacion de la practica, fue necesario analizar a profundidad los ejercicios que se nos plantearon, ya que al estar codificando, obtuve algunos errores relacionados con la manera en la que estaba interpretando los problemas.

El primer problema.El cual trata acerca de una carrera de la formula 1, lo resolví siguiendo la estructura que se uso para resolver el problema de la cebra, ya que era muy similar y solamente habia que cambiar algunos datos para obtener la salida que se requería. Al resolver este problema con ayuda de prolog fue bastante facil, ya que me evito de la engorrosa necesidad de tener que estar analizando condicion por condicion, Prolog se encargo de ello, solamente yo debia de codificar correctamente las condiciones que me daba el problema para que Prolog pudiera interpretarlas y darme una respuesta.

El segundo problema,no fue tan dificil de entenderlo ya que,en lo personal ya había trabajado con ese teorema en la materia de grafica y juegos, por lo que solo fue necesario seber como inplementarlo en Prolog.

La primer implementacion que di para este problema,funcionaba perfectamente para el caso de ejemplo que se nos proporciono, y para algunos otros que se me habían ocurrido, pero la probarlo para el mapa de la ciudad de mexico mi implementación no fuicionaba de la manera en la que se esperaria, ya que se quedaba procesando los datos hasta que ocurría un desborde de memoria y no daba respuesta alguna.

Asi que opte por moficar mi implementacion para que trabajara con una conjunto de pares el cual fuera almacenando las parejas de regiones y colores que se hallan ido generando antes de hacer una llamada recursiva,Esto me permitio tambien poder comprobar que las regiones adyacentes de un mapa que se iban agregando al conjunto de pares efectivamente tueviesen un color distinto y asi mismo si la condicion no se cumplia, se procederia a decechar la rama en la que se estaba trabajando y a hacer un backtraking para poder encontrar algún otro camino que satisfaga la condición.

¿La Ciudad de México puede ser coloreada con menos de 4 colores?

La respuesta es no, si ejecutamos el programa coloracion con los siguientes datos obtenemos lo siguiente:

- `swipl carrera.pl` compilamos

```
?-coloracion(cdmx,[azul,rojo,amarillo,cafe],S).  
S = [r16-azul, r15-rojo, r14-cafe, r13-rojo, r12-azul, r11-amarillo, r10-azul,  
r9-azul, ... - ...|...]
```

Notemos que le hemos proporcionado a el programa los datos del mapa que se va a colorear, y la cantidad de colores que se utilizaran, en este caso son 4.

La respuesta que obtenemos es practicamente una lista la cual nos indica que el mapa de la ciudad de mexico se pudo colorear con los 4 colores que le proporcionamos, y ademas que cada una de las regiones que son adyacentes entre si del mapa estan pintadas por un color diferente.

pero ahora si le proporcionamos los siguientes datos:

- `?-coloracion(cdmx,[azul,rojo,amarillo],S).`
`false.`

Podemos ver que la respuesta que obtenemos es false, lo cual nos indica que el mapa de la ciudad de México no se pudo pintar con 3 colores que proporcionamos, y hacer que se cumpliera lo que indica el teorema de los 4 colores.

por lo que puedo concluir que el mapa de la ciudad de mexico no se puede pintar con 3 colores y hacer que se cumpla lo que indica el teorema de los 4 colores.

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 encuentran los archivos carrera.pl y coloracion.pl, y ejecutar los siguientes comandos:

- Compilación y ejecución del archivo **carrera.pl** :

```
swipl carrera.pl
```

Esto hará que se compile el programa y sea posible su ejecución.

Para poder ver la respuesta que arroja el programa respecto a lo que pedía el problema 1, hay que escribir:

```
?- respuesta([Nombre de alguna variable empezando en mayuscula]).
```

Ejemplo:

```
?- respuesta(Carrera).
```

Esto nos arrojará una lista en donde se podrá observar como queda organizada la información de cada corredor, siguiendo las condiciones que se nos especificaron en el problema.

- Ejecución **coloracion.pl**:

```
swipl coloracion.pl
```

Esto hará que se compile el programa y sea posible su ejecución.

Así mismo al igual que con el anterior archivo, para ejecutar cualquiera de las funciones del programa solo basta escribir:

```
?- coloracion(cdmx, [Lista de colores], S)
```

Esto nos arrojará una respuesta la cual nos dirá si es que fue posible colorear el mapa de la ciudad de México siguiendo el teorema de los 4 colores con el número de colores que proporcionamos.

3. Conclusiones

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

En general resolver los problemas anteriores con Prolog fue de cierta manera sencillo ya que me ahorre la tarea de estar pensando y analizando cada posible solución que podía llegar a tener el problema, Prolog directamente me arrojaba una respuesta dependiendo de

la cantidad y la correctud de la informacion que le proporcionaba, y con base esa respuesta solo tenia que verificar que Prolog estaba interpretando la informacion de manera correcta.