



Proyecto de Lógica Computacional. (Versión B)

Por: Comas Castañeda Mauricio Santiago, Martínez
Osorio Benjamin y Medina Peralta Joaquin.

Planificación del proyecto

- Dividimos el proyecto en la investigación general de la programación lógica, usos y aplicaciones de la programación lógica y finalmente el código para el problema planteado.



Investigación general de la programación lógica.



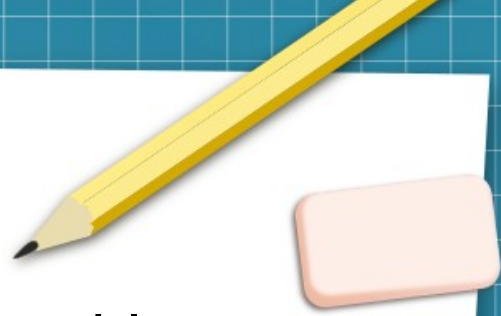
- Se buscaron la mayoría de referencias en el libro del mismo nombre escrito por Pascual Julián Iranzo y María Alpuente Frasnado.
- Gracias a esta investigación entendimos mas a fondo el paradigma de la programación lógica, basado en su mayoría por las Clausulas de Horn ya que gracias a su semántica operacional permite una implementación mas eficiente basada en un universo de discurso y sus interpretaciones, ademas concluimos por lo ya mencionado que este paradigma al poseer la capacidad de inferencia lógica abre las puertas a muchos usos y aplicaciones.

Investigación sobre aplicaciones de la programación lógica.



- Esta sección es en donde mas hallazgos encontramos, pues encontramos varios ejemplos de aplicaciones de la programación lógica, véase IA, gestión de Base de datos y sistemas expertos, para el ejemplo nos decantamos por los sistemas de traducción automática, en particular la plataforma Apertium de código abierto. Dicha plataforma utiliza como base un sistema de inferencia por reglas del lenguaje, pues sabemos que para crear un lenguaje se necesitan reglas que lo definan, para este trabajo se analizaron los módulos de analizador morfológico, transferencia léxica, transferencia estructural y el generador morfológico que componen el programa.

Código



- Finalmente para la parte del código se analizo el problema de el ratón borracho y en base a los requerimientos creamos nuestra base de conocimientos en prolog sobre como se movería el ratón para llegar a la salida así como los estados de muerte y borrachera que tomaba en casillas especificas.

Extracto de código en PROLOG.

- Aquí una muestra de una parte del código, con comentarios para entender las funciones:

```
68 % Movimiento en una dirección aleatoria
69 random_direction(D) :-
70     random_between(1, 4, R),
71     (R = 1 -> D = derecha;
72      R = 2 -> D = izquierda;
73      R = 3 -> D = arriba;
74      R = 4 -> D = abajo).
75
76 % Girar a la izquierda
77 girar_izquierda(derecha, arriba).
78 girar_izquierda(arriba, izquierda).
79 girar_izquierda(izquierda, abajo).
80 girar_izquierda(abajo, derecha).
81
82 % (si está en dirección de pared y no borracho) gira_izquierda hasta que la condición no se cumpla
83 % (si no está en dirección de pared y borracho) gira aleatorio
84 % Mantiene todo igual en otros casos
85 revisa_direccion(Raton, NRaton) :-
86     Raton = raton(X, Y, D, B),
87     (
88         viendo_pared(Raton), B = 0 -> girar_izquierda(D, ND), revisa_direccion(raton(X, Y, ND, B), NRaton);
89         not(viendo_pared(Raton)), B > 0 -> random_direction(ND), NRaton = raton(X, Y, ND, B);
90         NRaton = Raton
91     ).
92
93 % Avanzar al raton dependiendo de su dirección
94 avanza(X, Y, arriba, X, NY) :- NY is Y - 1.
95 avanza(X, Y, abajo, X, NY) :- NY is Y + 1.
96 avanza(X, Y, izquierda, NX, Y) :- NX is X - 1.
97 avanza(X, Y, derecha, NX, Y) :- NX is X + 1.
```

Conclusión

- Finalmente para la reflexión final el equipo concluyo que la programación lógica es sin duda una herramienta que aporta grandes beneficios en la resolución de problemas y es importante manejarla para adaptarse a los entornos nuevos que se presentan y a los campos laborales así como oportunidades que se brindan, pues aunque el paradigma es algo difícil de comprender para programadores que iniciaron con el clásico paradigma orientado a objetos es una poderosa herramienta que manejada correctamente puede hacer nuestro trabajo muchísimo mas sencillo, consideramos que PROLOG fue una buena introducción a este paradigma y esperamos seguir dominando este estilo de programación.