Imagen que contiene Logotipo

Descripción generada automáticamenteImagen que contiene Texto

Descripción generada automáticamente

Programas de prolog

Fundamentos de Inteligencia Artificial

Alonso Pérez Víctor Ernesto

Castro Elvira Diego

Daphne Sofia González Cano

Nancy Galicia Cocoletzi

Lizeth Hernández Rodríguez

UPIIT: Unidad Profesional Interdisciplinaria en Ingeniería Campus Tlaxcala Instituto Politécnico Nacional, Tlaxcala, Tlaxcala, México 9000

Ingeniera en Inteligencia Artificial

24 de abril de 2023

**Abstrac**

La programación lógica es un tipo de programación que se basa en la lógica de primer orden, es decir, se declaran qué cosas hay que hacer y el motor de ejecución decide cuál es la mejor forma de realizarlas, los programas lógicos obtienen resultados a través de la verdad (si es verdadero o no) y de la deducción lógica (de una cláusula lógica es una consecuencia del programa).

**Marco teórico**

**¿Qué es la programación lógica?**

La programación lógica es un tipo de programación que se basa en la lógica de primer orden, es decir, se declaran qué cosas hay que hacer y el motor de ejecución decide cuál es la mejor forma de realizarlas. En la lógica de programación se desarrollan una serie de instrucciones o sentencias en una secuencia determinada para lograr un objetivo concreto. Se organizan de forma coherente todas esas instrucciones para poder alcanzar el objetivo fijado de la forma más rápida y eficiente.

Imagen que contiene Diagrama

Descripción generada automáticamente

Los programas lógicos obtienen resultados a través de la verdad (si es verdadero o no) y de la deducción lógica (de una cláusula lógica es una consecuencia del programa).

Se basa en la lógica proposicional o de primer orden (utilizando un lenguaje y alfabeto de primer orden y una serie de normas y reglas de inferencia). Busca que una computadora sea capaz de deducir a partir de una serie de premisas.

**¿Qué es prolog?**

Se trata de un lenguaje de programación lógico e interpretado usado habitualmente en el campo de la inteligencia artificial, se ideo a principios de los años 70 en la universidad de Aix-Marsella. Los programas en Prolog se componen de cláusulas de Horn que constituyen reglas del tipo "modus ponendo ponens", es decir, "Si es verdad el antecedente, entonces es verdad el consecuente". No obstante, la forma de escribir las cláusulas de Horn es al contrario de lo habitual. Primero se escribe el consecuente y luego el antecedente. El antecedente puede ser una conjunción de condiciones que se denomina secuencia de objetivos. Cada objetivo se separa con una coma y puede considerarse similar a una instrucción o llamada a procedimiento de los lenguajes imperativos. En Prolog no existen instrucciones de control. Su ejecución se basa en dos conceptos: la unificación y el backtracking.

**¿Qué son los predicados en prolog?**

Los predicados son los elementos ejecutables en ProLog. Se utilizan para expresar propiedades de los objetos, y relaciones entre ellos, predicados. En ProLog se conocen como hechos. Se debe tener en cuenta que:

* Los nombres de todos los objetos y relaciones deben comenzar con una letra minúscula.
* Primero se escribe la relación o propiedad: predicado
* Y los objetos se escriben separándolos mediante comas y encerrados entre paréntesis: argumentos
* Al final del hecho debe ir un punto (".")

**¿Qué es una variable en prolog?**

Se utilizan para expresar propiedades de los objetos, y relaciones entre ellos, predicados. En ProLog se conocen como hechos. Se debe tener en cuenta que:

-Los nombres de todos los objetos y relaciones deben comenzar con una letra minúscula.

-Primero se escribe la relación o propiedad: predicado

-Y los objetos se escriben separándolos mediante comas y encerrados entre paréntesis: argumentos.

-Al final del hecho debe ir un punto (".")

**¿Qué es una base de conocimientos?**

Para responder a las preguntas o consultas formuladas por el programador, Prolog consulta una base de conocimiento. Esta base conocimiento representa el programa como tal, programa que se compone únicamente de cláusulas que, con el uso de la lógica, me expresan el conocimiento deseado por el programa.

La base de conocimiento o el programa se guarda en un archivo con la extensión ***'.pl'***, archivo que puede ser abierto y a partir de esto poderle hacer consultas a mi programa.

**Recursión y Prolog**

Prolog carece de mecanismos iterativos, por lo prolog utiliza recurison para:

* representar información (estructuras recursivas)
* procesar la información (relaciones recursivas)

**Ejemplo 1: Sumatoria**

La **sumatoria o sumatorio** se emplea para representar la suma de muchos o infinitos sumandos.

Explicaciones y ejemplos de sumatoria - 1

La expresión se lee: "***sumatoria*** *de Xi, donde i toma los valores de 1 a n*".

* La operación **sumatoria** se expresa con la letra griegra sigma mayúscula **Σ**.
* i es el valor inical llamado límite inferior.
* n es el valor final llamado líimite superior.

Si la sumatoria abarca la totalidad de los valores, su expresión se puede simplificar:

Explicaciones y ejemplos de sumatoria - 2

Se debe notar que, aunque el término sumatorio se refiere a un operador matemático útil para expresar cierto tipo de suma, no substituye este término a la palabra suma. Se dice: «la suma de dos y tres es cinco», y no «el sumatorio de dos y tres es cinco». Por la misma razón, decir que se realizará, por ejemplo, el sumatorio (o la sumatoria) de unos votos, es notoriamente un disparate. Los operadores de suma son útiles para expresar sumas de forma analítica; esto es, representar todos y cada uno de los sumandos en forma general mediante el «i-ésimo» sumando.

**Ejemplo 2: La Torre de Hanoi**

Las Torres de Hanói es un rompecabezas o juego matemático inventado en 1883 por el matemático francés Édouard Lucas. Este juego de mesa individual consiste en un número de discos perforados de radio creciente que se apilan insertándose en uno de los tres postes fijados a un tablero. El objetivo del juego es trasladar la pila a otro de los postes siguiendo ciertas reglas, como que no se puede colocar un disco más grande encima de un disco más pequeño.

|  |
| --- |
| **Algoritmo** Torres de Hanoi |
| **Entrada:** Tres pilas de números *origen*, *auxiliar*, *destino*, con la pila *origen* ordenada  **Salida:** La pila *destino*   1. **si** *origen* == { 1 } **entonces**    1. **mover** el disco 1 de pila origen a la pila destino (insertarlo arriba de la pila destino)    2. **terminar** 2. **si no**    1. *hanoi*( { 1 , … , n − 1 } ,*origen*,*destino*, *auxiliar*)     //mover todas las fichas menos la más grande (*n*) a la varilla auxiliar 3. **mover** disco *n* a *destino*                //mover la ficha grande hasta la varilla final 4. *hanoi* (*auxiliar*, *origen*, *destino*)          //mover todas las fichas restantes, 1...*n*–1, encima de la ficha grande (*n*) 5. **terminar** |

**Ejemplo 3: sucesión de Fibonacci**

**En matemáticas, la sucesión de Fibonacci (a veces mal llamada serie de Fibonacci) es la sucesión infinita de números naturales.**

0,1,1,2,3,5,8,13,21,34,55,89,144,233,377…

La sucesión comienza con los números 0 y 1, y a partir de estos, cada elemento es la suma de los dos anteriores. A los elementos de esta sucesión se les llama números de Fibonacci. Esta sucesión fue descrita en Europa por Leonardo de Pisa, matemático italiano del siglo XIII también conocido como Fibonacci.

Tiene numerosas aplicaciones en ciencias de la computación, matemáticas y teoría de juego. También aparece en configuraciones biológicas, como por ejemplo en las ramas de los árboles, en la disposición de las hojas en el tallo, en la flora de la alcachofa y en el arreglo de un cono.

El concepto fundamental de la sucesión de Fibonacci es que cada elemento es la suma de los dos anteriores. En este sentido, la sucesión puede expandirse al conjunto de los números enteros como de manera que la suma de cualesquiera dos números consecutivos es el inmediato siguiente.

***Sumatoria de N números***

*Código****:***

sumatoria(0, 0).

sumatoria(N, Resultado) :-

N > 0,

N1 is N - 1,

sumatoria(N1, R1),

Resultado is N + R1.

*Explicación del código:* Este código implementa el cálculo de la suma de los números naturales desde 0 hasta un número dado, la primer línea de código define un predicado de la sumatoria con dos argumentos N y Resultado, N representa el número hasta el que se debe sumar y Resultado, pues precisamente el valor que queremos encontrar. La segunda línea establece una regla base para la suma de cero que da igual a cero, en la tercer línea establecemos una regla recursiva para calcular la suma de números desde 1 hasta N, verificando que N sea mayor a O, si es así se resta 1 a N y se nombra a la función sumatoria con N1 como el nuevo valor de N, el valor de esta llamada se almacena en R1, luego se calcula el resultado sumando N y R1y se almacena en la variable Resultado.

*Ejecución del código:*

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

***Fibonacci***

fibonacci(0, 0).

fibonacci(1, 1).

fibonacci(N, Resultado) :-

N > 1,

N1 is N - 1,

N2 is N - 2,

fibonacci(N1, R1),

fibonacci(N2, R2), Resultado is R1 + R2.

Para poder ejecutar de forma correcta el código, primero es necesario poder colocar los casos bases de la sucesión, en este caso consta de dos; el primero corresponde a el caso base de 0, el cual regresa como resultado 0 y el segundo caso base corresponde a el 1 que en este caso regresa como resultado 1. Con la declaración de los casos base se pasó al diseño del algoritmo recursivo de la sucesión. Al algoritmo Fibonacci ingresa el valor N y resultado es la salida de la sucesión, Se comprueba que N es mayor a 1, si se cumple entonces se genera N1 que es igual a N-1 y también se genera N2 que es igual a N-2, estos dos van a ingresar de nuevo al algoritmo hasta que lleguemos a los casos base, es en este momento que cada uno de los fragmentos de la entrada inicial se van sumando hasta que lleguemos a las dos partes de la primera partición. La suma de estas partes se va almacenando dentro de la variable resultado y finalmente proyectada en pantalla. Obtuvimos los siguientes resultados.

Texto

Descripción generada automáticamenteInterfaz de usuario gráfica, Texto

Descripción generada automáticamente

***Torres de Hanoi***

hanoi(1, X, Y, \_):-

write('Mueve el disco de '),

write(X),

write(' a '),

write(Y),

nl.

hanoi(N, X, Y, Z):-

N>1,

M is N-1,

hanoi(M, X, Z, Y),

hanoi(1, X, Y, \_),

hanoi(M,Z,Y,X).

Explicación del código:

Para la ejecución del código, se hace la consulta

hanoi(<numeroDeDiscos>, <NombreDeLaTorre1>, <NombreDeLaTorre2>, <NombreDeLaTorre3>).

En el código, iniciamos por declarar el caso base, es decir la parte que se llamará a recursión mientras se cumpla la condición planteada en la segunda parte del código. Para este problema, el caso base se ejecuta, mientras N sea mayor a 1, es decir hasta que no se hayan movido todos los discos.

M fungirá como un contador descendente, que irá guardando el valor decremental de N. Este valor se utiliza para llamar al caso base, cambiando.

Ejecutándolo nos da el siguiente resultado:

Texto

Descripción generada automáticamenteTabla, Excel

Descripción generada automáticamente

**Conclusiones:**

A manera de conclusión, podemos comentar que los predicados lógicos nos permiten trasladar hechos de la vida real a un lenguaje que nos permite generar relaciones lógicas para de ahí inferir sí estas proposiciones son reales o falsas, esa es otra de sus ventajas, que el mundo de conocimiento lo simplifica a verdadero o falso, o bien, a 1 y 0, eso matemáticamente hablando, ahora, para trasladarlo a programación se utiliza la programación lógica que básicamente se enfoca en predicados, reglas, relaciones y base de conocimiento, el lenguaje idóneo para todo esto es Prolog, esto lo rectificamos al implementar los ejemplos que se plasmaron en el documento.

**Referencias:**

Roberto Vázquez Burguillo, 02 de febrero, 2016 Sucesión de Fibonacci. Economipedia.com

<https://economipedia.com/definiciones/sucesion-de-fibonacci.html>

Torres de Hanói. (2023, 6 de marzo). Wikipedia, La enciclopedia libre.

<https://es.wikipedia.org/w/index.php?title=Torres_de_Han%C3%B3i&oldid=149713329>

Qué significa sumatoria en Matemáticas (s.f.) Superpro.

<https://www.superprof.es/diccionario/matematicas/estadistica/sumatoria.html>