



Mi Universidad

MAPA CONCEPTUAL

Sofia Pereyra Orantes

Evaluación Perezosa/Fundamentos de programación lógica

Unidad 3 y 4

Programación Lógica

Ojeda Trujillo Juan José

Ingeniería en Sistemas Computacionales

Cuatrimestre 8

Comitan de Dominguez, Chiapas a 5 de Abril de 2025

EVALUACIÓN PEREZOSA/FUNDAMENTOS DE LA PROGRAMACIÓN LÓGICA

LA ESTRATEGIA DE EVALUACIÓN PEREZOZA

Estrategia de programación que retrasa la evaluación de expresiones hasta que se necesite su valor

TECNICAS

TECNICAS DE BACKTRAKING(PROLOG)

Si una alternativa falla el flujo retorna hasta la última decisión e intenta otra

TECNICAS DE GUARDIAS(GUARDS)

Si una alternativa falla el flujo retorna hasta la última decisión e intenta otra

TECNICA DE APRENDIZAJE REFORZADO

Recordar conexiones exitosas y aumentar su prioridad así como considerar las decisiones en el contexto del estado mutable

EJEMPLOS

En una clase de matemáticas: Profesor: "¿Cuál es la respuesta a este ejercicio?" Estudiante: "La respuesta es 15." Profesor: "¿Seguro? Bueno, te doy el punto."

En una revisión de código: Desarrollador 1: "Revisé el código, todo parece estar bien." Desarrollador 2: "¿Probaste todas las funcionalidades?" Desarrollador 1: "No, pero no parece que haya errores, así que lo doy por bueno."

```
def evaluar_perezosamente(condiciones):
    # Evaluación perezosa: retorna el primer valor True que encuentre.
    for condicion in condiciones:
        if condicion:
            return "Condición verdadera encontrada."
    return "No se encontró ninguna condición verdadera."

# Lista de condiciones (puede tener valores True o False)
condiciones = [False, False, True, False]

# Ejecutando la evaluación perezosa
resultado = evaluar_perezosamente(condiciones)
print(resultado)
```

REPASO DE LA LOGICA PRIMER ORDEN

Expresar hechos y reglas de manera declarativa. En lugar de decirle a la computadora cómo hacer algo (como en la programación imperativa), se le dice qué se debe hacer.

EJEMPLO

es_mayor_de_edad(Persona) :-
edad(Persona, Edad), Edad >= 18.

UNIFICACIÓN

Es el proceso de encontrar una sustitución de variables que haga que dos términos (o fórmulas) se conviertan en idénticos.

Un término en lógica de primer orden puede ser un predicado, constante, variable o una función.

RESOLUCIÓN

Técnica de inferencia utilizada en lógica de primer orden para derivar nuevas cláusulas a partir de otras, utilizando un proceso de unificación y luego aplicando una regla que "elimina" las variables comunes entre las dos cláusulas

CLAUSULA DE HORN

Disyunción de literales en la que, al menos, uno de ellos es negativo, y puede tener un máximo de un literal positivo.

PROGRAMACIÓN LÓGICA CON CLAUSULA DE HORN

```
import java.util.*;

public class ClausulasDeHorn {
    // Clase Persona para almacenar información sobre las relaciones familiares
    static class Persona {
        String nombre;
        boolean esMadre;
        boolean esPadre;

        Persona(String nombre) {
            this.nombre = nombre;
            this.esMadre = false;
            this.esPadre = false;
        }

        void setMadre() {
            this.esMadre = true;
        }

        void setPadre() {
            this.esPadre = true;
        }
    }

    public static void main(String[] args) {
        // Crear personas
        Persona maria = new Persona("Maria");
        Persona juan = new Persona("Juan");
        Persona pedro = new Persona("Pedro");

        // Establecer relaciones familiares
        maria.setMadre(); // Maria es madre de Juan
        juan.setPadre(); // Juan es padre de Pedro

        // Verificar si Maria es abuela de Pedro
        if (esAbuela(maria, juan, pedro)) {
            System.out.println(maria.nombre + " es abuela de " + pedro.nombre);
        } else {
            System.out.println(maria.nombre + " no es abuela de " + pedro.nombre);
        }

        // Método que simula una cláusula de Horn para verificar si alguien es abuela
        static boolean esAbuela(Persona x, Persona z, Persona y) {
            return x.esMadre && z.esPadre; // Maria es madre de Juan y Juan es padre de Pedro
        }
    }
}
```

RESOLUCIÓN SLD

Deriva nuevos hechos o respuestas a partir de un conjunto de hechos y reglas

SEMANTICA DE PROGRAMACIÓN LÓGICA

Significado y las reglas de interpretación de los programas escritos en lenguajes de programación lógicosignificado y las reglas de interpretación de los programas escritos en lenguajes de programación lógica

REPRESENTACIÓN CLAUSADA DEL CONOCIMIENTO



PROGRAMACIÓN LÓGICA CON NUMEROS Y ARBOLES

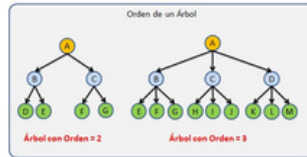
son

NUMEROS

los números pueden ser representados como enteros o reales. La manipulación de números suele implicar operaciones matemáticas, comparaciones y recursión.

ARBOLES

puede representarse como una estructura recursiva. Un árbol tiene un nodo raíz y subárboles que se representan como otros árboles.



```
// Definición de la clase Nodo para el árbol binario
class Nodo {
    int valor;
    Nodo izquierdo, derecho;
    // Constructor
    public Nodo(int item) {
        valor = item;
        izquierdo = derecho = null;
    }
}

// Definición de la clase Árbol Binario
class ArbolBinario {
    Nodo raiz;
    // Constructor
    public ArbolBinario() {
        raiz = null;
    }

    // Método para recorrer el árbol en orden (In-Order Traversal)
    void recorrerEnOrden(Nodo nodo) {
        if (nodo != null) {
            recorrerEnOrden(nodo.izquierdo); // Recorrer subárbol izquierdo
            System.out.print(nodo.valor + " "); // Imprimir el valor del nodo
            recorrerEnOrden(nodo.derecho); // Recorrer subárbol derecho
        }
    }

    // Método para iniciar el recorrido en orden desde la raíz
    void recorrer() {
        recorrerEnOrden(raiz);
    }

    // Método para insertar un nuevo valor en el árbol binario
    void insertar(Nodo raiz, int valor) {
        // Si el árbol está vacío, el nuevo nodo es la raíz
        if (raiz == null) {
            raiz = new Nodo(valor);
            return raiz;
        }

        // De lo contrario, recorrer el árbol
        if (valor < raiz.valor) {
            raiz.izquierdo = insertar(raiz.izquierdo, valor);
        } else if (valor > raiz.valor) {
            raiz.derecho = insertar(raiz.derecho, valor);
        }

        // Devolver el nodo sin cambios
        return raiz;
    }

    // Método para insertar un valor en el árbol
    void insertarValor(int valor) {
        raiz = insertar(raiz, valor);
    }
}

public class Main {
    public static void main(String[] args) {
        ArbolBinario arbol = new ArbolBinario();

        // Insertar valores en el árbol
        arbol.insertarValor(50);
        arbol.insertarValor(30);
        arbol.insertarValor(20);
        arbol.insertarValor(40);
        arbol.insertarValor(70);
        arbol.insertarValor(60);
        arbol.insertarValor(80);

        // Realizar un recorrido en orden
        System.out.print("Recorrido en orden del árbol binario:");
        arbol.recorrer(); // Salida esperada: 20 30 40 50 60 70 80
    }
}
```

ESPACIOS DE BUSQUEDA

es

Conjunto de todas las posibles soluciones o estados que el motor de inferencia debe explorar para resolver una consulta. Este espacio de búsqueda depende de los hechos, las reglas y las consultas definidas en el programa.

El motor de inferencia utiliza técnicas como la unificación, la resolución y el backtracking para explorar este espacio y encontrar respuestas

un

EJEMPLO

```
% Hechos
padre(juan, pedro).
padre(juan, ana).
padre(ana, maria).
padre(pedro, juan).
```

```
% Regla: Un abuelo es alguien que es padre de un hijo, que a su vez
es padre de otro.
abuelo(X, Y) :- padre(X, Z), padre(Z, Y).
```

```
% Consulta
?- abuelo(juan, maria).
```

- Consulta: La consulta es `?- abuelo(juan, maria)`, lo que pregunta si Juan es abuelo de María.
- Aplicación de la Regla: Para resolver esto, Prolog aplicará la regla `abuelo(X, Y) :- padre(X, Z), padre(Z, Y)` con `X = juan` y `Y = maria`. La consulta se convierte en:

`padre(juan, Z), padre(Z, maria)`

CONSULTA DE UNA BASE DE CLAUSULAS

es

Forma de representar el conocimiento en la lógica, utilizando cláusulas y especialmente las cláusulas de Horn

un

EJEMPLO

```
% Hechos
padre(juan, pedro).
padre(juan, ana).
padre(ana, maria).
padre(pedro, juan).
```

```
% Regla: Un abuelo es alguien que es padre de un hijo,
que a su vez es padre de otro.
abuelo(X, Y) :- padre(X, Z), padre(Z, Y).
```

```
% Consulta
?- abuelo(juan, maria).
```

CONTROL DE BUSQUEDA EN PROGRAMAS LÓGICOS

es

Forma en que el motor de inferencia explora el espacio de soluciones posibles para una consulta, utilizando técnicas como backtracking, unificación y resolución.

MANIPULACIÓN DE TERMINOS

es

Los términos son los bloques básicos de los programas lógicos, y pueden ser constantes, variables, o estructuras como listas o árboles.

forma de representar el conocimiento en la lógica, utilizando cláusulas y especialmente las cláusulas de Horn

uso
estructurar hechos y reglas que describen el mundo.



- Se divide en 4:
1. Átomos
 2. Números
 3. Variables
 4. Estructuras

BIBLIOGRAFÍA

Prezi, A. I. S. M. O. (s. f.). La estrategia de evaluación perezosa y Técnicas de programación funcional perezosa. prezi.com. <https://prezi.com/p/4-jachvnyj/la-estrategia-de-evaluacion-perezosa-y-tecnicas-de-programacion-funcional-perezosa/>

Abril, D. B. (2023). Lógica de primer orden. La Máquina Oráculo. <https://lamaquinaoraculo.com/matematicas/logica-de-primer-orden/>

Prezi, M. R. L. O. (s. f.). Consulta de Cláusulas en Programación Lógica y Funcional. prezi.com. <https://prezi.com/p/irg5yfn-yrfq/consulta-de-clausulas-en-programacion-logica-y-funcional/>

Del Carmen Leon Moreno, L. (2019, 28 mayo). 4.5 Espacios de búsqueda. <https://lucydelcarmenleonmoreno.blogspot.com/2019/05/45-espacios-de-busqueda.html>

Figueroa, A. B. L. (s. f.). 4.6 Programacion Logica con numeros, listas y arboles. <https://alexisbladimirlopezfigueroa.blogspot.com/2019/05/46-programacion-logica-con-numeros.html>