Lenguajes ID

Teorías del a computación

Docente: Santiago Salazar Fajardo

NRC: 8041

Presentado por:

Brian Steevens Zambrano Chaparro

ID: 518329

Colombia - Bogotá D

DESCRIPCION DEL PROBLEMA

El Problema está basado en encontrar las palabras que comienzan en los dos primeros dígitos de su ID “51”, que no terminen en el último digito de su ID “9” y la cantidad de símbolos que componen cada palabra es múltiplo de 3 0 de 7.

El alfabeto asignado es {1,2,3,4,5,6,7,8,9,0} .

MODELO

Teoría de modelos finitos

La teoría de modelos finitos es la parte de la teoría de modelos más cercanas al álgebra universal. Al igual que otras partes del álgebra universal, y a diferencia con otras áreas de la teoría de modelos, está relacionada principalmente con álgebras finitas, o más generalmente, con una σ-estructura finita para signaturas σ que pueden contener símbolos relacionales como en el siguiente ejemplo:

La signatura estándar para grafos es σgrph={E}, donde E es un símbolo de relación binaria.

Un grafo es una σgrph-estructura que satisface las proposiciones {\displaystyle \forall u\forall v(uEv\rightarrow vEu)} {\displaystyle \forall u\forall v(uEv\rightarrow vEu)} y {\displaystyle \forall u\neg (uEu)} {\displaystyle \forall u\neg (uEu)}.

Un σ-homomorfismo es una aplicación que conmuta con las operaciones y preserva relaciones de σ. Esta definición lleva a la noción usual de homomorfismo de grafos, que tiene la propiedad interesante que un homomorfismo biyectivo no necesita tener inverso. Las estructuras también forman parte del álgebra universal, después de todo, algunas estructuras algebraicas tales como grupos ordenados admiten una relación binaria del tipo < "menor que". Lo que distingue a un modelo finito de un ágebra universal es el uso de proposiciones lógicas más generales (como el ejemplo anterior) en lugar de identidades (en un contexto de teoría de modelos la identidad t=t' se escribe como una proposición {\displaystyle \forall u_{1}u_{2}\dots u_{n}(t=t')} {\displaystyle \forall u_{1}u_{2}\dots u_{n}(t=t')}.)

La lógica empleada en una teoría de modelos finitos generalmente es más expresiva que una lógica de primer orden, o la lógica estándar para la teoría de modelos más general o las estructuras infinitas.

Modelos para teorías lógicas de primer orden

Este artículo se enfoca en teoría finitaria de modelos de primer orden de estructuras infinitas. La teoría de modelos finitos, la cual se concentra en estructuras finitas, diverge significativamente del estudio de estructuras infinitas tanto en los problemas estudiados como en las técnicas usadas. La teoría de modelos en lógicas de orden superior o lógicas infinitarias está obstaculizada por el hecho de que la completitud no se cumple para estas lógicas. Actualmente existe un número importante de resultados sobre las propiedades de los sistemas lógicos tanto de primer orden como de segundo orden.

Debe tenerse presente que dada una teoría lógica de primer orden generalmente existe más de un modelo para dicha teoría, y dichos modelos usualmente no son isomorfos. Eso significa que los axiomas de una determinada teoría caracterizan en realidad aspectos de diferentes tipos de estructuras. Muchas veces esto es un resultado buscado. Por ejemplo, la teoría de grupos y sus axiomas definitorios admiten diversos modelos (cada grupo matemático de hecho es un modelo es un modelo de dicha teoría). En otras ocasiones como en el intento de formalizar los números reales mediante una teoría de primer orden se buscaba que esencialmente existiera un modelo único, sin embargo, el teorema de Löwenheim-Skolem permite ver que existen diversos modelos no isomorfos, entre ellos los números reales convencionales, pero también los números hiperreales constituyen otro modelo no isomorfo al anterior que también satisface los mismos axiomas y teoremas que los números reales.

La existencia de un modelo permite establecer la consistencia de una teoría. La existencia de diferentes modelos puede permitir establecer la independencia de algunos axiomas. Esencialmente eso es lo que puede establecer la teoría de modelos aplicada a la teoría de conjuntos axiomática, por ejemplo.

Modelos de ZFC

La existencia de diferentes modelos posibles para los axiomas de Zermelo-Fraenkel (ZFC) ha permitido establecer la independencia del axioma de elección y de la hipótesis del continuo de otros axiomas de la teoría de conjuntos (los principales resultados se deben a Paul Cohen (1963) y Kurt Gödel (1938).

Se ha probado que tanto el axioma de elección como su negación son consistentes con los axiomas de Zermelo-Fraenkel de la teoría de conjuntos. Y la hipótesis del continuo, es lógicamente independiente, de los axiomas de Zermelo-Fraenkel y el axioma de elección. Estos resultados son ejemplos de aplicaciones de la teoría de modelos a la teoría axiomática de conjuntos.

Modelos para la teoría de los números reales

Un ejemplo de los conceptos de la teoría de modelos es la teoría de los números reales. Comenzamos con un conjunto de individuos, donde cada individuo es un número real y un conjunto de relaciones y/o funciones como $\{ \times, +, -, ., 0, 1 \}$. Si hacemos una pregunta "$\exists$ y (y × y = 1 + 1)" en este lenguaje, entonces está claro que la sentencia es verdadera para reales, ya que existe tal número real y, a saber la raíz cuadrada de 2. Para los números racionales, sin embargo, la sentencia es falsa. Una proposición similar, "$\exists$ y (y × y = 0 − 1)", es falsa en los reales, pero es verdadera en los números complejos, donde i × i = 0 − 1.

Teoría de la demostración

La teoría de modelos puede emplearse como herramienta en la teoría de la demostración que se ocupa de lo que se puede probar con sistemas matemáticos dados, y cómo estos sistemas se relacionan entre sí. En principio la teoría de la demostración se ocupa de la complejidad sintáctica de las teorías a diferencia de la teoría de modelos que se ocupa principalmente de las posibilidades semánticas de la teoría.

PRUEBAS

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package menu;

import java.util.Scanner;
import javax.swing.JOptionPane;

/**
 *
 * @author bzamb
 */
import lenguaje.metodo;
public class menu {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Scanner leer=new Scanner(System.in);
        metodo mt=new metodo();
        String palabra="";
        System.out.println("Ingrese palabra: ");
        palabra=leer.next();//Entrada de la palabra
        System.out.println("Respuesta:");
        System.out.println(mt.lenguaje(palabra)); //imprime el resultado
    }
```

enu.menu » main » leer

: - Lenguaje (run)

```
run:
Ingrese palabra:
5123432
Respuesta:
Pertenece al lenguaje
BUILD SUCCESSFUL (total time: 11 seconds)
```

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package menu;

import java.util.Scanner;
import javax.swing.JOptionPane;

/**
 *
 * @author bzamb
 */
import lenguaje.metodo;
public class menu {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Scanner leer=new Scanner(System.in);
        metodo mt=new metodo();
        String palabra="";
        System.out.println("Ingrese palabra: ");
        palabra=leer.next();//Entrada de la palabra
        System.out.println("Respuesta:");
        System.out.println(mt.lenguaje(palabra)); //imprime el resultado
```

enu.menu 〉 main 〉 leer 〉

: - Lenguaje (run)

```
run:
Ingrese palabra:
312321432
Respuesta:
3
No pertenece al lenguaje
BUILD SUCCESSFUL (total time: 3 seconds)
```

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package menu;

import java.util.Scanner;
import javax.swing.JOptionPane;

/**
 *
 * @author bzamb
 */
import lenguaje.metodo;
public class menu {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Scanner leer=new Scanner(System.in);
        metodo mt=new metodo();
        String palabra="";
        System.out.println("Ingrese palabra: ");
        palabra=leer.next();//Entrada de la palabra
        System.out.println("Respuesta:");
        System.out.println(mt.lenguaje(palabra)); //imprime el resultado
    }
```

menu.menu » main » leer

ut - Lenguaje (run)

```
run:
Ingrese palabra:
2321423
Respuesta:
2
No pertenece al lenguaje
BUILD SUCCESSFUL (total time: 2 seconds)
```

CODIGO FUENTE

Menu

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package menu;

import java.util.Scanner;
import javax.swing.JOptionPane;

/**
 *
 * @author bzamb
 */
import lenguaje.metodo;
public class menu {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        Scanner leer=new Scanner(System.in);
        metodo mt=new metodo();
        String palabra="";
        System.out.println("Ingrese palabra: ");
        palabra=leer.next();//Entrada de la palabra
        System.out.println("Respuesta:");
        System.out.println(mt.lenguaje(palabra)); //imprime el resultado
    }

}
```

Método

```java
/**
 *
 * @author bzamb
 */
public class metodo {

    public String lenguaje(String palabra){
        int tam=palabra.length();
        boolean ver=true;
        String num=String.valueOf(tam);
        for(int i =0;i<tam;i++){//Confirma que pertenece al lenguaje
            for(int j =0;j<tam;j++){
                if(palabra.charAt(i)==j){

                }

            }
        }
        if(tam%3==0||tam%7==0){//Pertenece o no pertenece al lenguaje
            for(int i =0;i<tam;i++){//Recorre la entrada
                if(palabra.charAt(0)!='5'||palabra.charAt(1)!='1'||palabra.charAt(palabra.length()-1)=='9'){//Condiciones para pertenecer al lenguaje
                    System.out.println(palabra.charAt(i));// En caso de que la entrada imprime el error
                return "No pertenece al lenguaje";// En caso de que la entrada no cumpla con los parametros pertenece al lenguaje
            }


        }
        }
        else{
            return "No pertenece al lenguaje";// En caso de que la entrada no cumpla con ser multiplo de 3 o 7
        }
        return "Pertenece al lenguaje";// En caso de que la entrada cumpla con los parametros pertenece al lenguaje
    }

}
```

REFERENCIAS

https://es.wikipedia.org/wiki/Teor%C3%ADa_de_modelos