

Trabajo 1: Identificación de Funciones Matemáticas

Alumno: Ronald Wilder Incacutipa Muñuico

Docente: Ing. Torres Cruz Fred

Programación Numérica – FINESI
Universidad Nacional del Altiplano

Descripción del Problema

Usamos un programa en Python capaz de reconocer si una expresión matemática ingresada por el usuario es una **función** o no, además, identificar las variables utilizadas. El sistema debe contar la cantidad de variables distintas, así como el número de operaciones, incluyendo las multiplicaciones implícitas ($5x, xy$).

Especificaciones

- Entrada: cadena de texto con la función.
- Salida: listado de variables, número de variables distintas, total de operaciones y si es una funcion o no.
- Restricciones: máximo 3 variables; operaciones consideradas: $+, -, *, /, .$

Ejemplo

$$f(x, y, z) = 9xy + 2x + 3y + z$$

Salida esperada:

Variables encontradas: x, y, z

Número de variables: 3

Numero de operaciones: 15

Es funcion

Código en Python

```
1 import tkinter as tk
2 from tkinter import ttk, messagebox
3 import sympy as sp
4
5 def analizar():
6     expresion = entrada.get().strip()
7
8     if not expresion:
9         messagebox.showwarning("Error", "Por favor ingresa una expresión matemática.")
10        return
11
12
13    expresion = expresion.replace("^", "**")
14
15    try:
16        if "=" in expresion:
17            izquierda, derecha = expresion.split("=")
18            izquierda, derecha = izquierda.strip(), derecha.strip()
19            lhs = sp.sympify(izquierda)
20            rhs = sp.sympify(derecha)
21            eq = sp.Eq(lhs, rhs)
22        else:
23
24            lhs, rhs = sp.symbols("y"), sp.sympify(expresion)
25            eq = sp.Eq(lhs, rhs)
26
27
28        variables = list(eq.free_symbols)
29        num_vars = len(variables)
30
31
32        y = sp.Symbol("y")
33        if y in variables:
34            soluciones = sp.solve(eq, y)
35        else:
36
37            var_dep = variables[0]
38            soluciones = sp.solve(eq, var_dep)
39
40
41        operaciones = sum(expresion.count(op) for op in "+-*/**")
42
43
44        resultado_texto = f"Expresión: {expresion}\n"
45        resultado_texto += f"Variables encontradas: {'', ' '.join(str(v) for v in variables)}\n"
46        resultado_texto += f"Número de variables: {num_vars}\n"
47        resultado_texto += f"Número de operaciones: {operaciones}\n"
48
49        if len(soluciones) == 1:
```

```

50         resultado_texto += f" Es función: {soluciones[0]}"
51     elif len(soluciones) > 1:
52         resultado_texto += f" NO es función (múltiples salidas: {soluciones})"
53     else:
54         resultado_texto += " No se pudo determinar si es función."
55
56     resultado.set(resultado_texto)
57
58     except Exception as e:
59         messagebox.showerror("Error", f"No se pudo analizar la expresión.\n{e}")
60
61
62 ventana = tk.Tk()
63 ventana.title("Analizador de funciones")
64 ventana.geometry("500x400")
65 ventana.configure(bg="#f4f6f7")
66
67
68 estilo = ttk.Style()
69 estilo.configure("TButton", font=("Arial", 12), padding=6)
70 estilo.configure("TLabel", font=("Arial", 12))
71
72
73 titulo = tk.Label(
74     ventana, text=" Analizador de Funciones",
75     font=("Arial", 16, "bold"), bg="#f4f6f7", fg="#2c3e50"
76 )
77 titulo.pack(pady=10)
78
79
80 entrada = ttk.Entry(ventana, font=("Consolas", 14))
81 entrada.pack(pady=10, ipadx=20, ipady=5)
82
83
84 btn = ttk.Button(ventana, text="Analizar", command=analizar)
85 btn.pack(pady=10)
86
87
88 resultado = tk.StringVar()
89 label_resultado = tk.Label(
90     ventana, textvariable=resultado,
91     font=("Consolas", 12), bg="#ecf0f1", fg="#2c3e50",
92     relief="groove", justify="left", anchor="w"
93 )
94 label_resultado.pack(fill="both", expand=True, padx=20, pady=10)
95
96
97 ventana.mainloop()
98

```

Resultados



Figure 1: Captura de pantalla del analizador de funciones.

Conclusión

El programa diseñado cumple con el reconocimiento de funciones, permitiendo extraer de forma automática las variables y operaciones involucradas. Se usó la librería **Tkinter** para la parte gráfica y estructuras básicas de Python para el análisis.