

Hernandez Martínez José Angel de Jesús

1/mayo/2025

Hands-on 3

Compiladores

Implementación de Analizadores Sintácticos

Índice

Ejemplo para resolver expresiones regulares nivel básico.	3
Ejemplo para resolver expresiones regulares nivel medio y avanzado.....	4
Conclusiones	6
Referencias	6

Ejemplo para resolver expresiones regulares nivel básico.

Código:

```
def es_valida(expr):
    operadores = set("+-*/")
    par_abiertos = 0
    ultima = ""

    for c in expr:
        if c == " ":
            continue
        elif c.isdigit():
            ultima = "num"
        elif c in operadores:
            if ultima != "num" and ultima != "par_cierra":
                return False
            ultima = "op"
        elif c == "(":
            par_abiertos += 1
            ultima = "par_abre"
        elif c == ")":
            par_abiertos -= 1
            if par_abiertos < 0:
                return False
            if ultima == "op" or ultima == "par_abre":
                return False
            ultima = "par_cierra"
        else:
            return False # carácter inválido

    if par_abiertos != 0:
        return False
    if ultima in ["op", "par_abre"]:
        return False
    return True

# Ciclo para ingresar expresiones manualmente desde la terminal
while True:
    entrada = input("Ingresa una expresión (o escribe 'salir' para terminar): ")

    if entrada.lower() == 'salir':
        print("Saliendo...")
        break
```

```

if es_valida(entrada):
    print(" Expresión válida")
else:
    print(" Expresión inválida")

```

```

ry/OneDrive/Escritorio/compilers/analizador_sintatico.py
Ingresa una expresión (o escribe 'salir' para terminar): 3*4/2
Expresión válida
Ingresa una expresión (o escribe 'salir' para terminar): 0/0+3*6/0
Expresión válida
Ingresa una expresión (o escribe 'salir' para terminar): 3
Expresión válida
Ingresa una expresión (o escribe 'salir' para terminar): 3*
Expresión inválida
Ingresa una expresión (o escribe 'salir' para terminar): salir
Saliendo...

```

Ilustración 1 Funcionamiento analizador nivel básico

Ejemplo para resolver expresiones regulares nivel medio y avanzado.

```

def es_valida(expr):
    operadores_matematicos = set("+-*/")
    par_abiertos = 0
    ultima = ""
    i = 0

    while i < len(expr):
        c = expr[i]

        if c == " ":
            i += 1
            continue
        elif c.isdigit():
            ultima = "num"
        elif c == "(":
            par_abiertos += 1
            ultima = "par_abre"
        elif c == ")":
            par_abiertos -= 1
            if par_abiertos < 0:

```

```

        return False
    if ultima in ["op", "par_abre"]:
        return False
    ultima = "par_cierra"
elif c in operadores_matematicos:
    if ultima != "num" and ultima != "par_cierra":
        return False
    ultima = "op"
elif expr[i:i+3] == "AND":
    if ultima != "num" and ultima != "par_cierra":
        return False
    ultima = "op"
    i += 3
    continue
elif expr[i:i+2] == "OR":
    if ultima != "num" and ultima != "par_cierra":
        return False
    ultima = "op"
    i += 2
    continue
elif expr[i:i+3] == "NOT":
    if ultima != "num" and ultima != "par_cierra" and ultima != "":
        return False
    ultima = "op"
    i += 3
    continue
else:
    return False # Caracter inválido

i += 1

if par_abiertos != 0:
    return False
if ultima in ["op", "par_abre"]:
    return False
return True

# Entrada del usuario
print("Introduce expresiones para validar. Escribe 'salir' para terminar.")
while True:
    expr = input(">> ")
    if expr.lower() == "salir":
        break
    if es_valida(expr):

```

```
print(" Expresión válida\n")
else:
    print(" Expresión inválida\n")
```

```
Introduce expresiones para validar. Escribe 'salir' para terminar.
>> 1 AND 2 * (3-2)
    Expresión válida

>> 3 AND 4 + (2-4)
    Expresión válida

>> 3 OR 5 - 45
    Expresión válida

>> 3 or 6 0
    Expresión inválida
```

Ilustración 2 Funcionamiento analizador sintáctico intermedio y avanzado

Conclusiones

Fue una actividad que fue sencilla gracias a las explicaciones del archivo proporcionado por el profesor y las investigaciones anteriores para así poder hacer de manera eficiente el que compile el código.

Referencias

Mendez, J. A. (marzo de 2025). *classroom*. Obtenido de classroom: [analizador-sintactico.pdf](#)