



Escuela Superior de Cómputo

TEORÍA DE LA COMPUTACIÓN

PRÁCTICA 5: *Buscador Ansi-C*

Autor:

Rodrigo Gerardo Trejo Arriaga

Octubre 2023

Práctica 5:

Buscador Ansi-C

La conversión de Autómatas Finitos No Deterministas (AFND) a Autómatas Finitos Deterministas (AFD) es un proceso importante en la teoría de la computación. Permite transformar un autómata no determinista en uno determinista, lo que facilita su implementación y comprensión. Aquí explicaremos el procedimiento básico para llevar a cabo esta conversión.

Los AFND son más expresivos y flexibles, pero su determinismo puede dificultar su implementación. Por otro lado, los AFD son más simples de diseñar y entender, pero pueden no ser capaces de reconocer ciertos lenguajes. La conversión de AFND a AFD busca combinar la expresividad de los primeros con la simplicidad de los segundos.

El procedimiento general para convertir un AFND en un AFD es el siguiente:

1. **Construcción de Conjuntos Potencia:** Para cada conjunto de estados alcanzables en el AFND mediante transiciones epsilon (transiciones sin entrada), se debe construir un conjunto potencia. Esto generará un nuevo estado en el AFD.
2. **Transiciones del AFD:** Cada conjunto potencia se convierte en un estado del AFD. Las transiciones desde un conjunto de estados en el AFND se convierten en transiciones desde un estado en el AFD. Para cada símbolo de entrada, se calcula el conjunto de estados alcanzables y se asocia con una transición en el AFD.
3. **Estados de Aceptación:** Un estado en el AFD se marca como estado de aceptación si contiene al menos un estado de aceptación del AFND.
4. **Estado Inicial:** El estado inicial del AFD se obtiene a partir del conjunto potencia que contiene el estado inicial del AFND.
5. **Simplificación:** Es posible que el AFD resultante tenga estados redundantes. Se pueden aplicar técnicas de simplificación, como la eliminación de estados inaccesibles y la unión de estados equivalentes, para obtener un AFD más compacto.

1. Ejemplo de Conversión

A continuación, se presenta un ejemplo simple de conversión de un AFND a un AFD:

AFND:

	<i>a</i>	<i>b</i>
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\{q_1\}$	$\{q_2\}$
q_2	$\{q_2\}$	$\{q_0, q_2\}$

AFD:

	<i>a</i>	<i>b</i>
$\rightarrow p_0$	p_0	p_1
p_1	p_0	p_2
p_2	p_2	p_1

Este es un ejemplo simplificado con tres estados y dos símbolos de entrada. En la conversión, se generaron nuevos estados y se establecieron las transiciones correspondientes. El estado inicial se obtuvo del conjunto potencia que contiene el estado inicial del AFND.

I. Instrucciones

A continuación se presentan las instrucciones para diseñar un Autómata Finito Determinista (DFA) y realizar su conversión, lectura de un archivo de texto, identificación de palabras reservadas, y otras tareas:

1. **Diseño del NFA:** Comience diseñando un Autómata Finito No Determinista (NFA) que represente las reglas de reconocimiento para palabras reservadas. Este NFA servirá como base para la conversión a DFA.
2. **Conversión a DFA:** Realice la conversión del NFA a un Autómata Finito Determinista (DFA). Muestre todo el proceso a través de los subconjuntos y tablas de transición. Asegúrese de explicar cómo se obtienen los nuevos estados y las transiciones.
3. **Lectura de un Archivo de Texto:** El programa debe ser capaz de leer un archivo de texto que contenga el código fuente o texto a analizar. Puede ser un archivo de una página web u otro origen.
4. **Identificación de Palabras Reservadas:** Utilice el DFA diseñado para identificar cada palabra reservada en el archivo de texto. Registre cuántas veces se encuentra cada palabra y anote su posición (x, y) en el archivo.
5. **Registro de Evaluación:** Mientras el DFA analiza el archivo, registre en otro archivo la evaluación del autómata por cada carácter leído y cambio de estado. Esto incluye toda la historia del proceso de reconocimiento.
6. **Graficar el DFA:** Genere una representación gráfica del DFA para visualizar su estructura. Esto facilitará la comprensión de su funcionamiento y permitirá una mejor depuración si es necesario.

II. Proceso de conversión

Se dibujó el autómata no determinista a mano y en equipo

Posteriormente se realizó la conversión a determinista mediante conjuntos en la tablita como se vio en clase

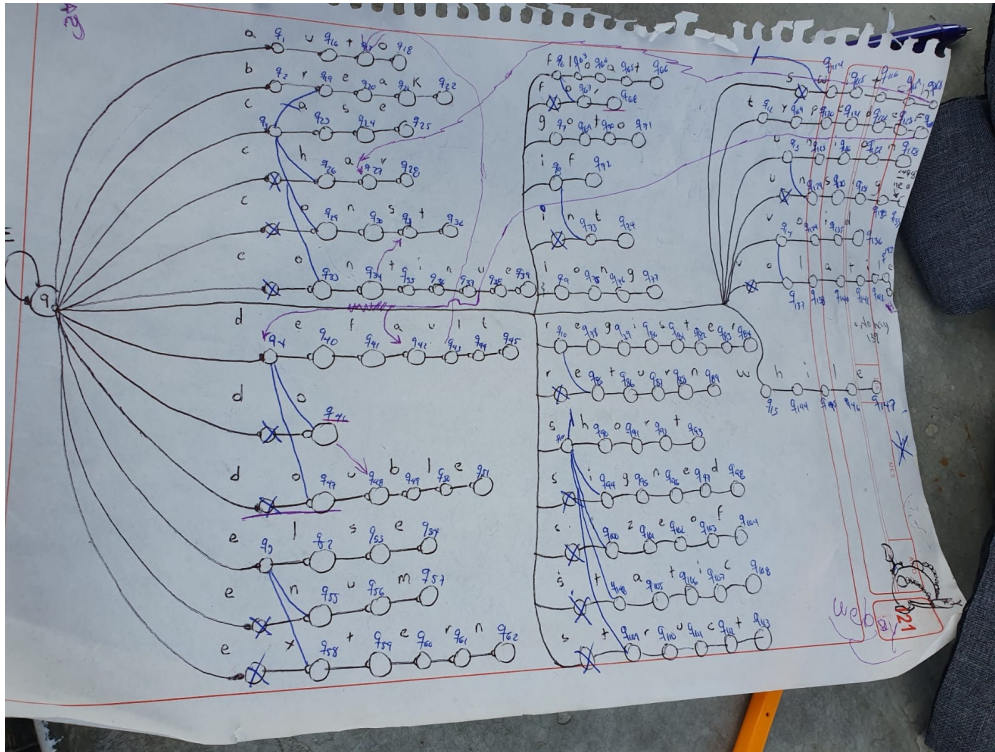


Figura 1: Diagrama del autómata no determinista

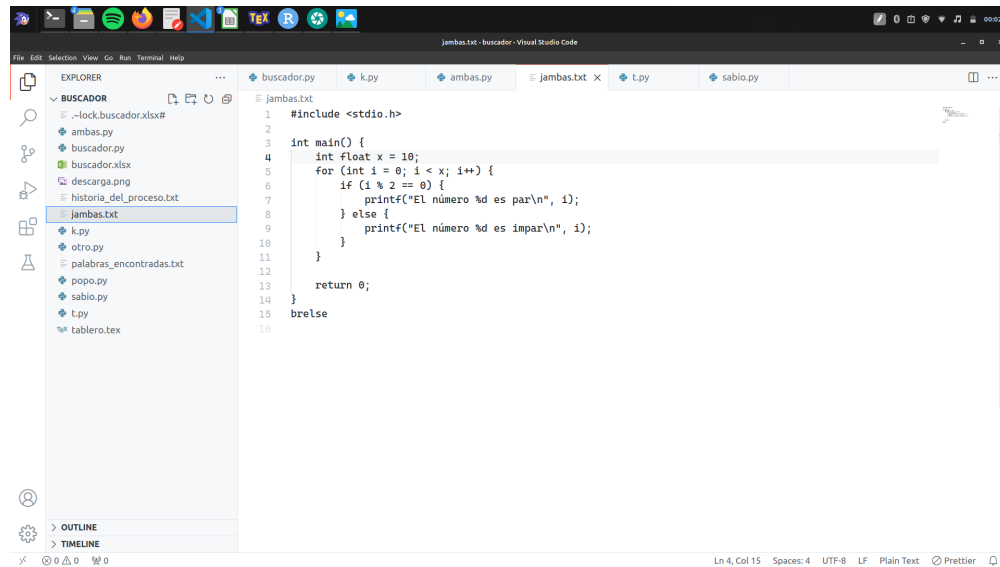
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	q	y	e	g	u	g	a	b	g	o	o	o	o	o	o	o	o	o	o	o	o
2	q14.00	q0	q14.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0	q15.00	q0	q0.00	q16.00	q0.07	q0.00	q0	q0	q0	q0
3	q13.00	q0	q14.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0	q18.00	q0.00	q0	q0.07	q0.00	q0	q0	q0	q0	q0
4	q0.012	q0	q14.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0.019	q0	q0.00	q0	q0.07	q0.00	q0	q0	q0	q0	q0
5	q11.00	q0	q14.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0	q0.00	q0	q0.00	q0	q0.07	q0.00	q0	q0	q0	q0
6	q10.00	q0	q14.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0	q0.00	q0	q0.00	q0	q21.00.07	q0.00	q0	q0	q0	q0
7	q0.00	q0	q14.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0	q0.00	q0.022	q0	q0.07	q0.023.00	q0	q0	q0	q0	q0
8	q0.00	q0	q14.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0.020.00	q0	q0.00	q0	q0.07	q0.00	q0	q0	q0	q0	q0.027.00
9	q0.07	q0	q14.00.028	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0	q0.00	q0	q0.00	q0	q0.07	q0.00	q0	q0	q0	q0
10	q0.00	q0.029	q14.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0	q0.00	q0	q0.00	q0	q0.07	q0.00	q0	q0	q0	q0
11	q0.00	q0	q14.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0	q0.00	q0	q0.00	q0	q0.07	q0.00	q0	q0	q0	q0
12	q0.04	q0	q14.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0	q0.00	q0	q0.00	q0	q0.07	q0.00	q0	q0	q0	q0
13	q0.00	q0	q14.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0	q0.00	q0	q0.00	q0	q0.07	q0.00	q0	q0	q0	q0
14	q0.00	q0	q14.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0	q0.00	q0	q0.00	q0	q0.07	q0.00	q0	q0	q0	q0
15	q0.00	q0	q14.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0	q0.00	q0	q0.00	q0	q0.07	q0.00	q0	q0	q0	q0
16	q0.00	q0	q14.00	q13.00	q0.012	q0	q11.00.035.00	q10.00	q0	q0.00	q0	q0.00	q0	q0.00	q0	q0.07	q0.00	q0	q0	q0	q0
17	q138.00	q0	q30.014.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0	q0.00	q0	q0.00	q0	q0.07	q0.00	q0	q0	q0	q0
18	q15.00	q0	q14.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0	q0.00	q0	q0.00	q0	q0.07	q0.00	q0	q0	q0	q0
19	q16.00	q0	q14.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0	q0.00	q0	q0.00	q0	q0.07	q0.00	q0	q0	q0	q0
20	q17.03.00	q0	q14.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0.041.00.00	q0	q0.00	q0	q0.07	q0.00	q0	q0	q0	q0	q0
21	q18.00	q0	q14.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0	q0.00	q0	q0.00	q0	q0.07	q0.00	q0	q0	q0	q0
22	q0.019	q0	q14.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0.05.00.043	q0	q0	q0.00	q0.044	q0	q0.07	q0.00	q0	q0	q0
23	q20.00.012	q0	q14.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0.019	q0	q0.00	q0	q0.07	q0.045.00.00	q0	q0	q0	q0	q0
24	q21.00.07	q0	q40.014.00.013.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0	q0.00	q0	q0.00	q0	q0.07	q0.00	q0	q0	q0	q0
25	q0.022	q0	q14.00	q13.00.004	q0.012	q0	q11.00	q10.00	q0	q0.00	q0.020.00	q0	q0.00	q0	q0.07	q0.00	q0.048	q0	q0	q0	q0.027.00
26	q0.023.00	q0.029	q14.00	q13.00	q0.012	q0	q11.00.049	q10.00	q0	q0.00	q0	q0.00	q0	q0.00	q0	q0.07	q0.00	q0	q0	q0	q0
27	q0.024.04	q0	q14.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0	q0.00	q0.052	q0	q0.07	q0.00	q0	q0	q0	q0	q0
28	q0.025	q0	q14.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0	q0.00	q0.051	q0	q0.07	q0.00	q0	q0	q0	q0	q0
29	q0.026	q0	q14.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0	q0.00	q0	q0.00	q0	q0.07	q0.00	q0	q0	q0	q0
30	q0.027.00	q0	q14.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0	q0.00	q0	q0.00	q0	q0.07	q0.00	q0	q0	q0	q0
31	q14.00.028	q0	q14.00	q13.00.003	q0.012	q0	q11.00	q10.00	q0	q0.00	q0.015.00	q0	q0.00	q0.016.00	q0	q0.07	q0.00	q0.054.00	q0	q0	q0
32	q0.029	q0	q14.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0	q0.00	q0	q0.00	q0	q0.07	q0.00	q0	q0	q0	q0
33	q0.030	q0	q14.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0	q0.00	q0	q0.00	q0	q0.07	q0.00	q0	q0	q0	q0
34	q11.01.00	q0	q14.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0.027.00.03	q0	q0	q0.00	q0.07	q0.00	q0	q0	q0	q0	q0
35	q0.032	q0	q14.00	q13.00	q0.012	q0	q11.00	q10.00	q0	q0.00	q0	q0.00	q0.038	q0	q0.07	q0.00	q0	q0	q0	q0	q0

Figura 2: Diagrama del autómata no determinista

III. Resultados

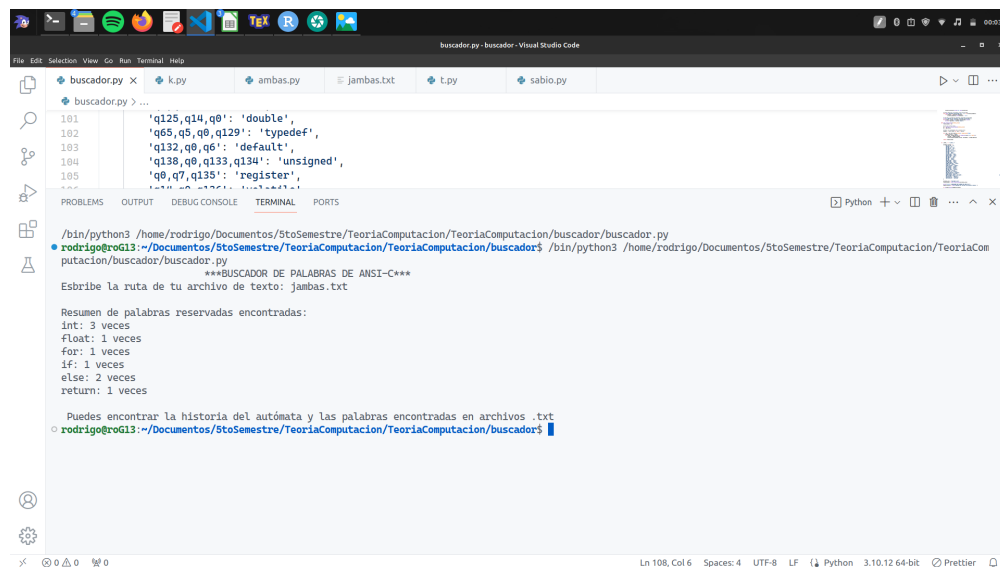
IV. Código de Implementación

```
1
2 import openpyxl
3 import os
4
5 def es_palabra_clave(palabra, transiciones):
6     estado_actual = 'q0'
7     historia_temp = []
8     for caracter in palabra:
9         # Si el caracter no es alfabético, se reinicia al estado q0
10        if not caracter.isalpha():
11            historia_temp.append(f"Caracter leido: '{caracter}'\tEstado
                actual: {estado_actual}\tEstado siguiente: {'q0'}")
12            estado_actual = 'q0'
13            continue
14
15        # Procesa el caracter con las transiciones del DFA
16        if (estado_actual, caracter) in transiciones:
17            estado_siguiente = transiciones[(estado_actual, caracter)]
18            historia_temp.append(f"Caracter leido: '{caracter}'\tEstado
                actual: {estado_actual}\tEstado siguiente: {estado_siguiente}"
                )
19            estado_actual = estado_siguiente
20        else:
21            return False, [], None
22        return estado_actual in estados_finales, historia_temp,
            estado_actual
23
24 def identificar_palabras_clave_en_texto(archivo, transiciones):
25     historia = [] # Almacenar la historia del proceso
26     palabras_encontradas = {} # Almacenar palabras reservadas y sus
        posiciones
27     detalles_palabras = [] # Almacenar detalles de las palabras
        encontradas
28
29     with open(archivo, 'r') as archivo_texto:
30         for num_linea, linea in enumerate(archivo_texto, start=1):
31             palabras = linea.split()
32             for num_palabra, palabra in enumerate(palabras, start=1):
33                 es_clave, historia_temp, estado = es_palabra_clave(palabra,
                    transiciones)
34                 historia.extend(historia_temp)
35
36         if es_clave:
```



```
1 #include <stdio.h>
2
3 int main() {
4     int float x = 10;
5     for (int i = 0; i < x; i++) {
6         if (i % 2 == 0) {
7             printf("El número %d es par\n", i);
8         } else {
9             printf("El número %d es impar\n", i);
10        }
11    }
12    return 0;
13 }
14
15 brelse
16
```

Figura 3: Texto a analizar con el autómata



```
101 'q125,q14,q0': 'double',
102 'q65,q5,q0,q129': 'typedef',
103 'q132,q0,q6': 'default',
104 'q138,q0,q133,q134': 'unsigned',
105 'q0,q7,q135': 'register',
...
***BUSCADOR DE PALABRAS DE ANSI-C***
Escribe la ruta de tu archivo de texto: jambas.txt

Resumen de palabras reservadas encontradas:
int: 3 veces
float: 1 veces
for: 1 veces
if: 1 veces
else: 2 veces
return: 1 veces

Puedes encontrar la historia del autómata y las palabras encontradas en archivos .txt
rodrigo@rogl3:~/Documentos/StoSemestre/TeoriaComputacion/TeoriaComputacion/buscador$
```

Figura 4: Palabras reconocidas

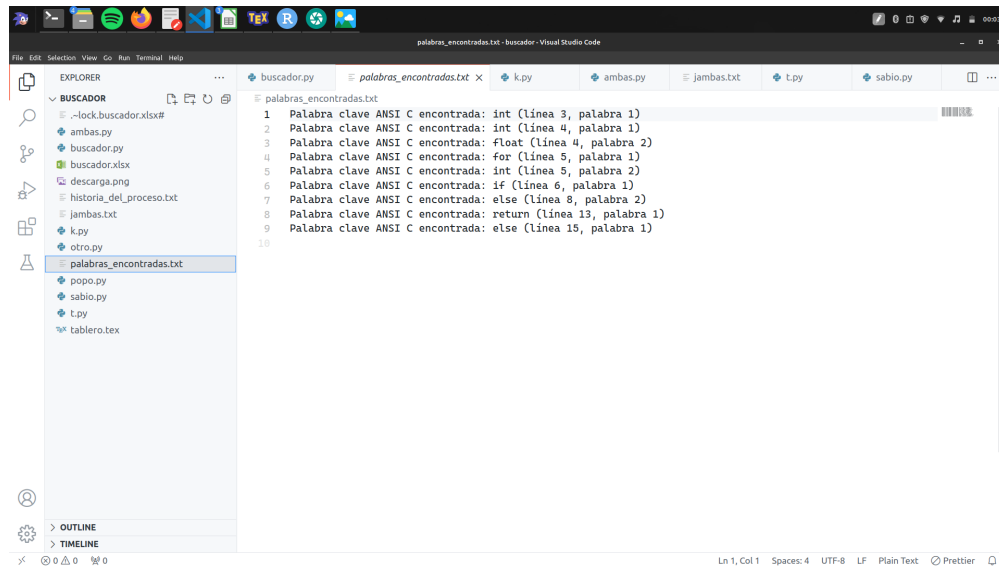


Figura 5: Posiciones de las palabras dentro del archivo analizado

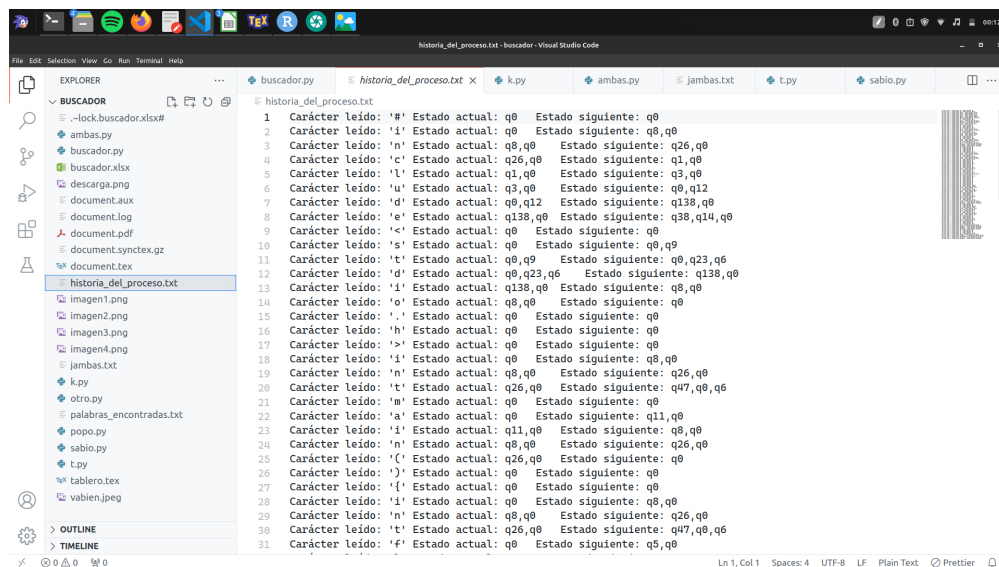


Figura 6: Historia de ejecución

```
37 detalles = f"Palabra clave ANSI C encontrada: {estados_finales[
    estado]} (línea {num_linea}, palabra {num_palabra})"
38 detalles_palabras.append(detalles)
39 palabras_encontradas[estados_finales[estado]] =
    palabras_encontradas.get(estados_finales[estado], 0) + 1
40
41 # Guarda la historia del proceso en un archivo
42 with open('historia_del_proceso.txt', 'w') as historia_archivo:
43     historia_archivo.write('\n'.join(historia))
44
45 # Guarda detalles de palabras clave encontradas
46 with open('palabras_encontradas.txt', 'w') as archivo_palabras:
47     for detalle in detalles_palabras:
48         archivo_palabras.write(detalle + '\n')
49
50 # Imprime el resumen de palabras reservadas encontradas
51 print("\nResumen de palabras reservadas encontradas:")
52 for palabra, conteo in palabras_encontradas.items():
53     print(f"{palabra}: {conteo} veces")
54
55 def leer_transiciones(archivo_xlsx):
56     transiciones = {}
57
58     # Abre el archivo Excel
59     wb = openpyxl.load_workbook(archivo_xlsx)
60     ws = wb.active
61
62     # Obtén los encabezados de la primera fila
63     headers = [cell.value for cell in ws[1]]
64
65     for row in ws.iter_rows(min_row=2, values_only=True):
66         estado_inicial = row[0]
67         for index, estado_destino in enumerate(row[1:-1], start=1):
68             entrada = headers[index]
69             transiciones[(estado_inicial, entrada)] = estado_destino
70
71     return transiciones
72
73
74 if __name__ == "__main__":
75
76     estados_finales = {
77         'q5,q27,q0': 'if',
78         'q38,q0': 'do',
79         'q47,q0,q6': 'int',
80         'q56,q0,q7': 'for',
81         'q67,q0': 'enum',
82         'q14,q69,q0': 'else',
```



```
83         'q70,q0': 'goto',
84         'q73,q0': 'auto',
85         'q13,q85,q0': 'long',
86         'q138,q0,q86': 'void',
87         'q88,q14,q0': 'case',
88         'q91,q0,q7': 'char',
89         'q0,q96': 'union',
90         'q97,q0': 'break',
91         'q0,q102,q6': 'short',
92         'q105,q0,q6': 'float',
93         'q14,q0,q106': 'while',
94         'q108,q0,q23,q6': 'const',
95         'q112,q0': 'extern',
96         'q138,q0,q133': 'signed',
97         'q5,q0,q115': 'sizeof',
98         'q1,q0,q116': 'static',
99         'q0,q117,q6': 'struct',
100        'q37,q0,q118': 'switch',
101        'q0,q120': 'return',
102        'q125,q14,q0': 'double',
103        'q65,q5,q0,q129': 'typedef',
104        'q132,q0,q6': 'default',
105        'q138,q0,q133,q134': 'unsigned',
106        'q0,q7,q135': 'register',
107        'q14,q0,q136': 'volatile',
108        'q137,q14,q0': 'continue'
109    }
110
111    archivo_xlsx = "buscador.xlsx"
112    transiciones = leer_transiciones(archivo_xlsx)
113
114    print("\t\t\t ***BUSCADOR DE PALABRAS DE ANSI-C***")
115    nombre_archivo = input("Esbríbe la ruta de tu archivo de texto: ")
116
117    if os.path.exists(nombre_archivo):
118        identificar_palabras_clave_en_texto('jambas.txt', transiciones)
119        print("\n Puedes encontrar la historia del aut mata y las
120              palabras encontradas en archivos .txt")
121
122    else:
123        print(f"El archivo {nombre_archivo} no existe.")
```