

Instituto Politécnico Nacional

ESCUELA SUPERIOR DE COMPUTACIÓN

BUSCADOR DE PALABRAS

Práctica 4

Alumno:

Alcántara Covarrubias Erik

Profesor:

Juarez Martinez Genaro

Grupo: 4CM6

1 Introducción

Un autómata finito no determinístico es aquel que puede estar en varios estados a la vez, al igual que un autómata finito determinístico tiene un conjunto finito de estados, un conjunto finito de símbolos de entrada, un estado inicial y un conjunto de estados finales. También tiene funciones de transición.

Un autómata finito no determinístico se representa esencialmente como un autómata finito determinístico:

$$A = \{Q, \Sigma, \delta, q_0, F\} \quad (1)$$

donde:

- Q es un conjunto finito de estados.
- Σ es un conjunto finito de símbolos de entrada.
- q_0 , un elemento de Q , es el estado inicial.
- F , un subconjunto de Q , es el conjunto de estados finales (o de aceptación).
- δ la función de transición, es una función que toma como argumentos un estado de Q y un símbolo de entrada de Σ y devuelve un subconjunto de Q . Observe que la única diferencia entre un AFN y un AFD se encuentra en el tipo de valor que devuelve δ : un conjunto de estados en el caso de un AFN y un único estado en el caso de un AFD.

2 Marco teórico

Primero denominaremos palabras clave a las palabras que buscaremos, en este caso es se procede a diseñar un autómata finito no determinista que va a indicar, mediante un estado final, que ha encontrado una de las palabras clave. El texto de un documento se introduce carácter a carácter en este AFN, el cual reconoce a continuación las apariciones de las palabras clave en dicho texto. Existe una forma simple para que un AFN reconozca un conjunto de palabras clave.

- Hay un estado inicial con una transición a sí mismo para cada uno de los símbolos de entrada, por ejemplo, todos los caracteres ASCII imprimibles si estamos examinando texto. Intuitivamente, el estado inicial representa una “conjetura” de que todavía no hemos detectado una de las palabras clave, incluso aunque hayamos encontrado algunas de las letras de una de esas palabras.
- Para cada palabra clave $a_1a_2\dots a_k$, existen k estados, por ejemplo, q_1, q_2, \dots, q_k . Existe una transición desde el estado inicial a q_1 para el símbolo a_1 , una transición desde q_1 a q_2 para el símbolo a_2 , etc. El estado q_k es un estado de aceptación e indica que se ha encontrado la palabra clave $a_1a_2\dots a_k$.

Ya teniendo el AFN, las reglas para poder construir el AFD desde el AFN son los siguientes:

- Si q_0 es el estado inicial del AFN, entonces $\{q_0\}$ es uno de los estados del AFD.
- Suponemos que p es uno de los estados del AFN y se llega a él desde el estado inicial siguiendo un camino cuyos símbolos son $a_1a_2\dots a_m$. Luego uno de los estados del AFD es el conjunto de estados del AFN constituidos por:
 - q_0 .
 - p .
 - Cualquier otro estado del AFN al que se pueda llegar desde q_0 siguiendo un camino cuyas etiquetas sean un sufijo de $a_1a_2\dots a_m$ es decir, cualquier secuencia de símbolos de la forma $a_ja_{j+1}\dots a_m$.

3 Explicación del problema e implementación del algoritmo

3.1 Problema a resolver

Programar el autómata finito determinístico que reconozca las palabras: web, webpage, website, webmaster, ebay, page, site

1. Diseñar el NFA.
2. Realizar la conversión a DFA mostrando todo el proceso a través de los subconjuntos y tablas.
3. El programa deberá de leer un archivo de texto, podría ser de una página web.
4. El autómata deberá de identificar cada palabra reservada con el DFA, contarlas e indicar dónde las encontró.
5. En un archivo imprimir la evaluación del autómata por cada carácter que lea y cambio de estado, es decir, toda la historia del proceso.
6. En otro archivo enumerar, contar y anotar donde están las palabras encontradas.
7. Tener una opción para ver el autómata, es decir, hay que graficarlo.

3.2 Implementación

4 Pruebas de escritorio y sus grafos

5 Conclusión

6 Bibliografía

- Introduction to Automata Theory, Languages, and Computation John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman Pearson/Addison Wesley, 2nd edition
- Software Foundation, P. (2022, 13 enero). 3.10.2 Documentation. Documentación de Python. Recuperado 15 de febrero de 2022, de <https://docs.python.org/es/3/>