

Reporte: Práctica 1

Barrera Pérez Carlos Tonatihu
Profesor: Saucedo Delgado Rafael Norman
Compiladores
Grupo: 3CM6

1 de septiembre de 2017

Índice

1. Introducción	1
2. Desarrollo	1
3. Resultados	2
4. Conclusiones	2
5. Referencias	2

1. Introducción

Los autómatas finitos no determinista y determinista nos permiten identificar si una cadena (secuencia de símbolos) pertenece a un determinado lenguaje regular. Las principales diferencias entre estos dos tipos de autómatas son las siguientes.

- Autómata finito no determinista: Un mismo símbolo puede conducir a diferentes estados desde un mismo estado, además, la cadena vacía es válida.
- Autómata finito determinista: El autómata tiene por cada estado y por cada símbolo sólo una transición hacia otro estado.

La descripción formal de un autómata determinista A es la siguientes.

$$A = (Q, \Sigma, \delta, q_0, F)$$

Donde:

- Q es un conjunto de estados finitos.
- Σ es un conjunto finito de símbolos de entrada.
- δ es una función de transición que toma como argumentos un estado y un símbolo y retorna un nuevo estado. En un autómata no determinista la función devuelve un conjunto de estados que pertenecen a Q .
- q_0 es un estado inicial que pertenece al conjunto de estados finitos Q .
- F es un conjunto de estados finales que pertenecen al conjunto Q .

Todas estas características fueron importantes en esta práctica ya que al tenerlas en consideración se pudieron modelar los dos tipos de autómatas mediante el uso de programación orientada a objetos. Y por ende poder representar cualquier autómata finito a partir de los elementos que lo componen.

2. Desarrollo

En esta práctica se usó el paradigma orientado a objetos para poder modelar cualquier autómata finito determinista o no determinista con base a sus componentes y con esto poder evaluar cualquier cadena y poder determinar si pertenece a algún determinado lenguaje. Se programaron dos clases, una para el autómata determinista y otro para el no determinista.

El código de la clase AFN fue:

El código de la clase AFD hereda muchas propiedades y métodos de la clase AFN modificando el método que se encarga de la evaluación de la cadena.

3. Resultados

hola

4. Conclusiones

Fue indispensable conocer los componentes de este tipo de autómatas además de saber el como trabajan ya que sin este conocimiento seria imposible el poder modelarlos en una clase. La parte difícil de esta práctica fue el AFN ya que al poder ir a varios estados desde una mismo con el mismo símbolo por lo que aumenta su complejidad al aumentar el numero de caminos que se pueden tomar. Así que el revisar trabajos anteriores en este tema fue importante para poder solucionar este problema.

5. Referencias