Instituto Politécnico Nacional
Escuela Superior de Cómputo
Ingeniería en Sistemas Computacionales

Práctica 3: Implementación del algoritmo AFN ->AFD.

Compiladores

Profesor: M.C Rafael Norman Saucedo Delgado
Alumno: Iris Yonitzi Marínez González Boleta: 2015090419
irismartinezgonzalez@gmail.com

26 de enero de 2021

Índice

1. Introducción

Como ya se vió en clase, el algoritmo de los subconjuntos tiene como objetivo tranformar un autómata finino no determinista en un autómata determinista, es decir, resive como entrada un autómata AFN y entrega como salida uno del tipo AFD.

Citando el libro de su método es el siguiente:

Construye una tabla de transición Dtran para D. Cada estado de D es un conjunto de estados del AFN, y construimos Dtran para que D pueda simular "en paralelo" todos los posibles movimientos que N puede realizar sobre una cadena de entrada dada. Nuestro primer problema es manejar las transiciones épsilon de N en forma apropiada.

Dichas transiciones son tratados con funciones auxiliaries:

- $E-cerradura(s)$: Conjunto de estados del AFN a los que se puede llegar desde el estado s del AFN, sólo en las transiciones $E(épsilon)$.
- $E-cerradura(T)$: Conjunto de estados del AFN a los que se puede llegar desde cierto estado s del AFN en el conjunto T, sólo en las transiciones $E; = U_s$ en T $E - cerradura(s)$.
- $mover(T, a)$:Conjunto de estados del AFN para los cuales hay una transición sobre el símbolo de entrada a, a partir de cierto estado s en T.

El libro indica que el algorimo de la construcción de subconjuntos se ve de la siguiente forma:

```
while ( hay un estado sin marcar T en Destados ) {
    marcar T;
    for ( cada símbolo de entrada a ) {
        U = E-cerradura(mover(T, a));
        if ( U no está en Destados )
            agregar U como estado sin marcar a Destados;
        Dtran[T, a] = U;
    }
}
```

De lo anterior el objetivo de la práctica es siguiente:
Utilizar los conocimientos sobre el paradigma orientado a objetos, los autómatas finitos, y el algorimo de los Subconjuntos vistos en clase, para diseñar e implementar dicho algoritmo retomando las clases de la Práctica 1.

2. Desarrollo

Como se retomó el análisis de la práctica 1 *Clases AFN y AFD* para esta solo se agregó una clase más llamada *SubConjunto*, y un método a la clase *AFN*, *convertirAFD()*.

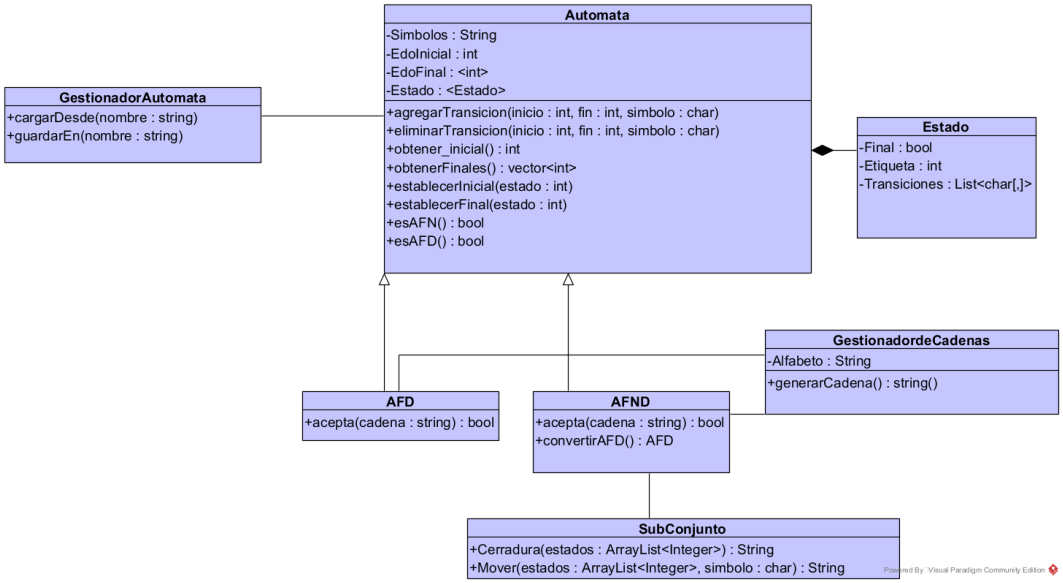

Figura 1: Diagrama de clases

Al realizar la práctica primero hice un pseudocódigo de lo que serían las funciones que necesita el algoritmo con base en los tributos de mis clases y de las funciones que indica el propio libro de consulta, posteriormente se pasó a código creando el packete correspondiente a la clase y sus métodos.

Para la clas Estado creé un método auxiliares el cual ayuda a tratar de forma óptima la información del AFN :

- $ArrayList < Integer > buscarDestino(charsimbolo)$: este método tiene como función buscar los destinos o los estados destinos dado un simbolo segun la estructura del autómata.

Esté método facilitó la implementación de las funciones$E - cerradura(s)$ y $mover(T, a)$ que requiere el algoritmo.

Haciendo pruebas unitarias de dichas funciones implementadas tenemos los siguientes resultados:

- El autómata de prueba es el siguiente:

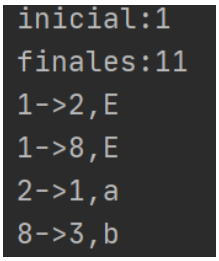

Figura 2: Autómata de prueba

- El conjunto de prueba es el siguiente:

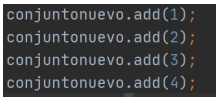

Figura 3: Conjunto de prueba

- El resultado es el siguiente:

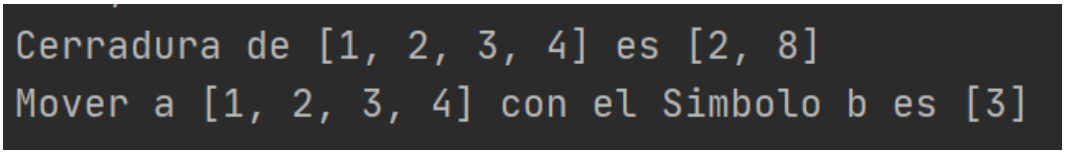

Figura 4: Resultado

3. Conclusiones

Después de la realización de esta práctica, me dí cuenta que puedo óptimizar ciertas partes del código que se implemento, y que de me debo de poner al corriente y profuncizar no solo con el lenguaje, sino también el orientado a objetos qeu aún me falta mucho.

Esto me lleva a hacerme una pregunta muy importante: *¿Qué hice en mi curso de Programacón Orienteda a Objetos.*

4. Bibliografía y Referencias

- AHO, ALFRED V. COMPILADORES. PRINCIPIOS, TÉCNICAS Y HERRAMIENTAS. Segunda edición PEARS ON EDUCACIÓN, México, 2008