



Instituto Politécnico Nacional



Escuela Superior de Cómputo ESCOM

Compiladores

Profesor Rafael Norman Saucedo Delgado

*“Práctica. Algoritmo de minimización de
número de estados de un AFD”*

Ángel David Ortega Prado

3CV16

Fecha: 08/11/2021

Algoritmo de minimización de estados de un AFD

Objetivo

Comprender el algoritmo de minimización de estados de un AFD y programar dicho algoritmo (Java como caso personal).

Introducción y desarrollo.

Existen distintos métodos para la minimización de estados de un AFD, para nuestro caso utilizaremos determinado algoritmo, que nos resulta de forma más sencilla implementarlo y por lo visto en clase.

Pueden haber muchos AFD's que reconozcan el mismo lenguaje sin importar el número de estados con los que cuenten, transiciones (para cada estado se derivan n número de transiciones de acuerdo con su alfabeto).

Sin embargo, si implementamos un analizador léxico como un AFD, por lo general es preferible que el AFD tenga el menor número de estados posible, puesto que cada estado requiere entradas en la tabla para describir el analizador léxico.

En cuanto al nombre de cada estado, se le puede otorgar poca importancia. Decimos que dos autómatas tienen nombre de estados equivalentes si uno puede transformarse en el otro solo con cambiar los nombres de los estados. Esto quiere decir que podemos reducir el número de los estados de uno o de otro, de tal forma que los dos acepten el mismo alfabeto y que se puedan obtener mismas salidas sean cuales sean.

El AFD mínimo existe y es único, es decir.

Dado un AFD $M = (Q, \Sigma, \delta, q_0, F)$, existe un AFD M' con $L(M) = L(M')$ y tal que M' tiene el mínimo número de estados de entre todos los AFD que reconocen $L(M)$.

El método de minimización de número de estados de un AFD se puede realizar a partir de varios pasos:

1. Empezar con una partición inicial Π con dos grupos, F y $S - F$, los estados de aceptación y de no aceptación de D .

2. Aplicar el procedimiento de la figura 3.64 para construir una nueva partición Π nueva.

al principio, dejar que $\Pi \text{ nueva} = \Pi$;

for (cada grupo G de Π) {

 particionar G en subgrupos, de forma que dos estados s y t
 se encuentren en el mismo subgrupo, si y sólo si para
 todos los símbolos de entrada a , los estados s y t tienen
 transiciones sobre a hacia estados en el mismo grupo
 de Π ;

 /* en el peor caso, un estado estará en un subgrupo por sí
 solo */ sustituir G en $\Pi \text{ nueva}$ por el conjunto de todos los
 subgrupos formados;

}

3. Si $\Pi \text{ nueva} = \Pi$, dejar que $\Pi \text{ final} = \Pi$ y continuar con el paso (4). De no ser así, repetir el paso (2) con $\Pi \text{ nueva}$ en vez de Π .
4. Elegir un estado en cada grupo de $\Pi \text{ final}$ como el representante para ese grupo.

Programación en Java

Se crearán alrededor de 6 clases

Una clase llamada *MinimizacionEstados* tendrá el método main donde solamente se van a instanciar e invocar métodos y objetos de otras clases correspondientes.

La clase en donde se obtienen las entradas y datos correspondientes para el AFD (como el número de estados, estado inicial, estados finales y alfabeto) tiene como nombre DatosAFD.