Minimización de AFD’S

Eduardo Salas
Jorge Nicho
Alvaro Aguirre
Teoría de la computación
UTEC

June 5, 2020

Introducción

Podemos particionar los estados de cualquier AFD en grupos de mutuos indistinguibles estados. Los miembros de dos grupos diferentes son siempre distinguibles. Si reemplazamos esos grupos por un único estado, resulta en un AFD equivalente al inicial pero con una cantidad de estados mínimos.

Definición del problema

La construcción de un AFD no nos asegura el mejor rendimiento, ya que tiene dos o más estados que podrían ser un solo estado. Existen AFD’s que pueden ser minimizados y ofrecernos las siguientes ventajas:

1. Ejecución: Debido a que se podría reducir la cantidad de estados de un AFD, este no demoraría tanto en procesar una cadena, por lo cual disminuiría su tiempo de ejecución.

2. Costo: Cuanto mayor sea el número de estados mayor será el costo.

3. Complejidad: Tener un menor número de estados sin afectar el lenguaje de un AFD nos permite tener un diseño más limpio.

1 Estado del Arte

Dado un AFD M = (Q,$\sum$,δ , q0, F), se trata de encontrar un AFD M’ con L(M) = L(M’) y tal que M’ tenga el mínimo número de estados posible. El método consiste en encontrar todos los estados que son equivalentes, es decir, que son indistinguibles en el autómata. Por cada clase de estados equivalentes, el autómata mínimo necesitará un solo estado. Hemos hallado 3 algoritmos para lograr este objetivo: Hopcroft, Moore y Myhill Nerode.

1.1 Algoritmo de Hopcroft

Hopcroft desarrolló en 1970 un algoritmo de minimización que se ejecuta en tiempo O(nlogn)en un autómata de n estados. No se conoce otro algoritmo más rápido que funcione para autómatas en general.
A = (Q, i, F) autómata en el alfabeto A. Sea P una partición de Q.
Un divisor es un par (P, a), con P $\in$ P y a $\in$ A. El objetivo de un divisor es dividir otra clase de P.
El divisor (P, a) divide la clase R $\in$ P si $\varnothing$ (P $\cap$ R $\cdot$ a (R $\cdot$ a o equivalentemente si $\varnothing$ (a 1P $\cap$ R (R . Aquí a1P = q | q $\cdot$ a $\in$ P.
En cualquier caso, denotamos por (P, a)|R la partición de R compuesto por los conjuntos no vacíos de a 1P $\cap$ R and R \ a 1P. El divisor (P, a) divide R si (P, a)|R 6= R.

1.1.1 Ejemplo

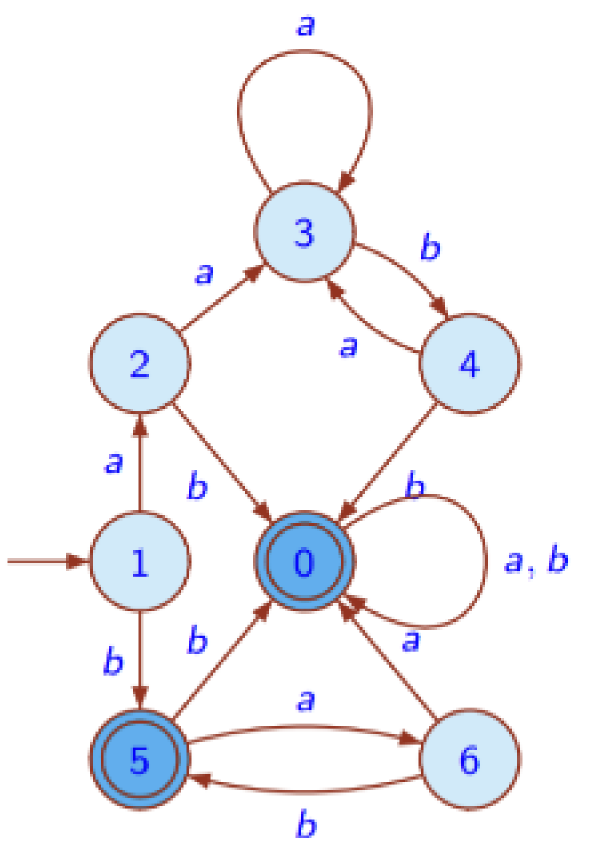


Figure 1: Ejemplo algoritmo de Hopcroft

P es la partición actual y W es el conjunto de espera.

1. Partición inicial P: 05|12346

2. Conjunto de espera W : (05, a), (05, b)
3. Eligiendo divisor : (05, a)
4. Estado divisor : a - 105 = 06
5. Primera clase en dividirse: 12346 $\rightarrow$ 1234|6
6. Divisores para agregar: (6, a) and (6, b)
7. Segunda clase en dividirse: 05 $\rightarrow$ 0|5
8. Divisores para agregar: (5, a) (or (0, a))
9. Divisores para reemplazar: (05, b) : by (0, b) and (5, b)
10. Nueva partición P: 0|1234|5|6
11. Nuevo conjunto de espera W: (0, b), (6, a), (6, b), (5, a), (5, b)

1.1.2 Pseudocódigo de hopcroft

```
P := {F, Q \ F};
W := {F, Q \ F};
while (W is not empty) do
     choose and remove a set A from W
     for each c in Σ do
          let X be the set of states for which a transition on c leads to a state in A
          for each set Y in P for which X ∩ Y is nonempty and Y \ X is nonempty do
               replace Y in P by the two sets X ∩ Y and Y \ X
               if Y is in W
                    replace Y in W by the same two sets
               else
                    if |X ∩ Y| <= |Y \ X|
                         add X ∩ Y to W
                    else
                         add Y \ X to W
          end;
     end;
end;
```

Figure 2: Pseudocódigo hopcroft

1.2 Algoritmo de Moore

El método para minimizar un autómata mediante el algoritmo de moore consiste en encontrar todos los estados que son indistinguibles entre sí y sustituirlos por un único estado. Para ello lo principal es averiguar qué estados lo son y cuáles no. El método para saber qué estados son indistinguibles es el siguiente:
a. Si hay algún estado inalcanzable eliminarlo.
b. (i := 0) Marcar todos los estados que pueden distinguirse con la cadena vacía (es decir, todos los finales se pueden distinguir de los no finales).
c. (i := i + 1) Marcar como distinguibles q y q' si con algún a $\in \sum$ tenemos δ(q, a) y δ(q' , a) dos estados que ahora son distinguibles.
d. Si en el paso anterior se han distinguido nuevos estados, entonces volver al paso.

1.2.1 Ejemplo

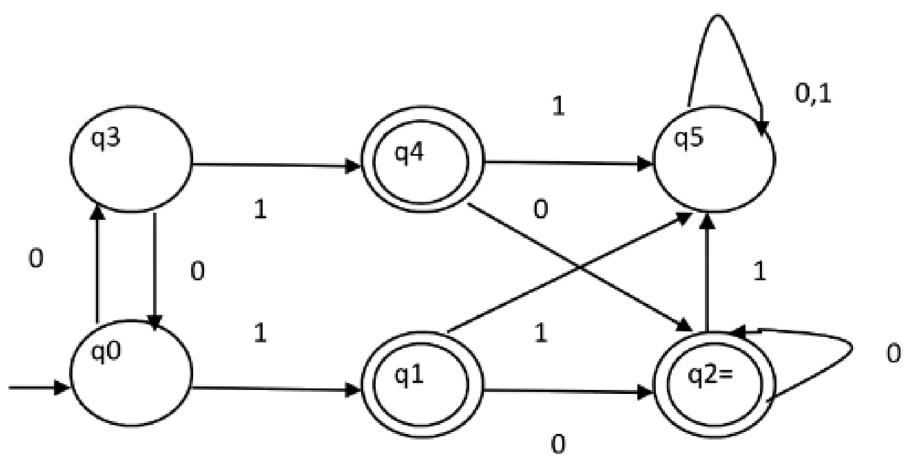


Figure 3: Autómata ejemplo

Paso 1: Separamos los estados de aceptación y los de no aceptación.

$$\mathbb{P}_0 = \{\{q_1, q_2, q_4\}, \{q_0, q_3, q_5\}\}$$

Figure 4: Paso 1

El conjunto de estados lo llamaremos Po siendo k = 0 ; quiere decir que tienen 0 transiciones iguales, solo están divididos por aceptación y no aceptación.

Paso 2: Separamos aquellos estados en cada conjunto de Po que tenga 1 estado de transición distinguible.

$$\delta(q_1, 0) = \delta(q_2, 0) = q_2 \ \ y \ \ \delta(q_1, 1) = \delta(q_2, 1) = q_5$$

Figure 5: Paso 2

Por lo tanto en el primer conjunto, todos los valores son indistinguibles. Quiere decir que este no será particionado en P1. Ahora veamos el segundo conjunto:

$$\delta(q_0, 0) = q_3 \ \ y \ \ \delta(q_3, 0) = q_0$$
$$\delta(q_0, 1) = q_1 \ \ y \ \ \delta(q_3, 1) = q_4$$

Figure 6:

Los movimientos de q0 con 0 y q3 con 0 son q3 y q0 que están en el mismo conjunto de partición en Po. Lo mismo sucede en q0 con 1 y q3 con 1, que son q1 y q4 , que también están en la misma partición en Po. Caso contrario sucede con q5, veamos:

$$\delta(q_0, 0) = q_3 \ \ y \ \ \delta(q_5, 0) = q_5 \ \ y \ \ \delta(q_0, 1) = q_1 \ \ y \ \ \delta(q_5, 1) = q_5.$$

Figure 7:

Vemos que no pertenecen al mismo conjunto en Po. Es por ello que P1 quedaría de la siguiente manera:

$$P_1 = \{\{q_1, q_2, q_4\}, \{q_0, q_3\}, \{q_5\}\}$$

Este es el caso final puesto que no se pueden realizar más particiones. El autómata quedaría de la siguiente manera.

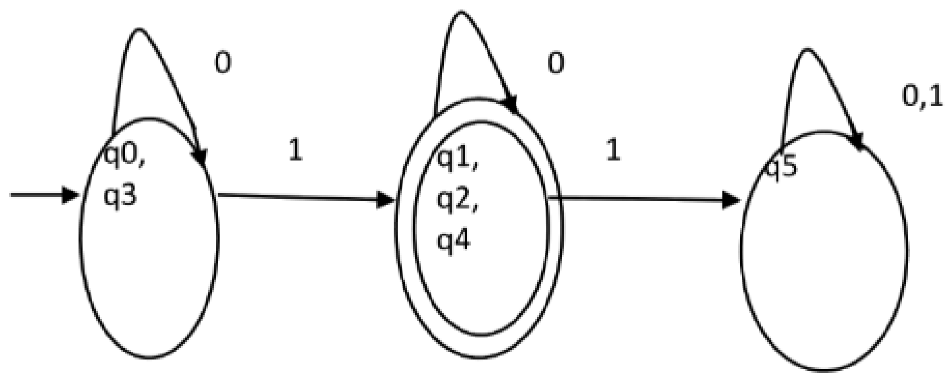
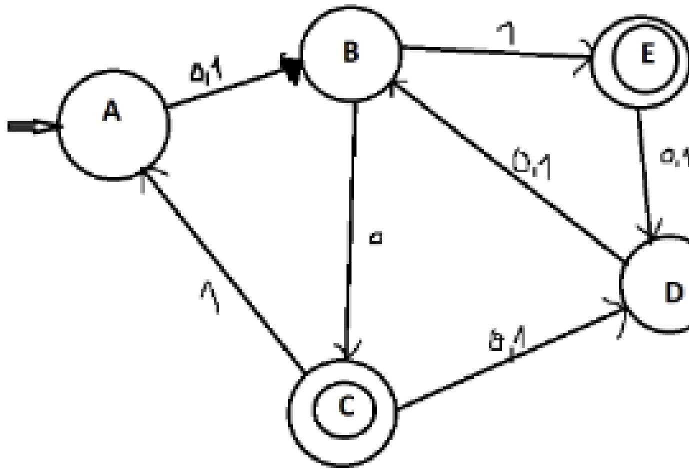
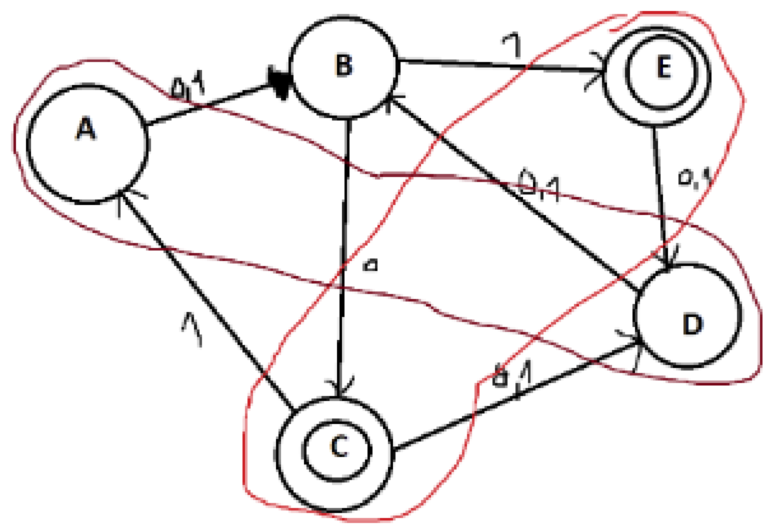


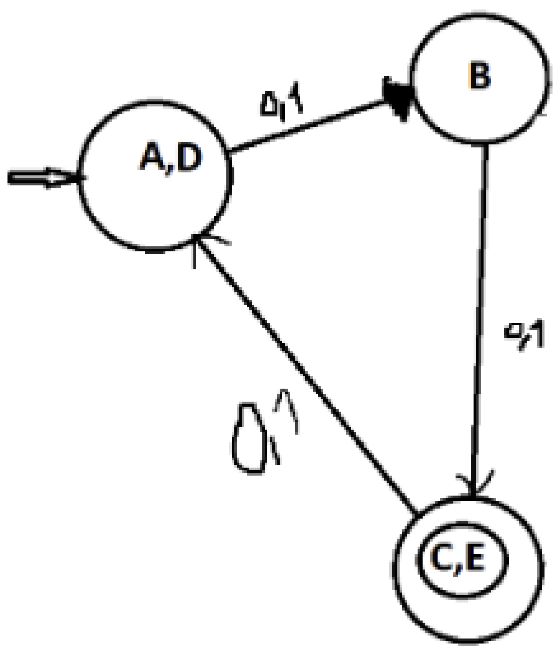
Figure 8: Aútomata final

1.3 Algoritmo de Myhill Nerode

Primero dibujamos una tabla para todos los pares de estados (P, Q). Luego marcamos los pares donde P sea un estado final y Q no lo sea o viceversa. Para los pares no marcados, analizamos sus estados de transición. Por ejemplo [δ(P,0), δ(Q,0)]. Si está marcado el nuevo par de estados a los que nos dirigimos con 0, entonces marcamos el par (P,Q). De lo contrario seguimos analizando los otros pares no marcados hasta que no se pueda marcar nada más. Finalmente combinamos todos los pares no marcados y formamos un único estado en el AFD minimizado.


A
B
C
D
E
0,1
1
0,1
0,1
0
1
0,1


A
B
C
D
E
0,1
1
0,1
0,1
0
1
0,1


A,D
B
C,E
0,1
0,1
0,1