Minimización de Autómatas Finitos Deterministas

Sebastián Ruiz Blais
ruizble@yahoo.com
Estudiante de pregrado,
Escuela de Ciencias de Computación e Informática
Universidad de Costa Rica

Resumen:
El algoritmo de minimización de autómatas finitos deterministas permite simplificar el manejo de autómatas. Se presenta aquí los detalles de este algoritmo y un ejemplo concreto. Se busca facilitar el entendimiento del algoritmo, en particular en estudiantes de cursos de compiladores o en profesionales que busquen la referencia al tema.

1) Introducción:

Cuando se trabaja en compiladores, se requiere utilizar gramáticas y autómatas asociados a éstas. Existen herramientas que permiten simplificar la utilización de los autómatas. En particular, para los autómatas finitos deterministas (AFD), es de suma utilidad aplicar el algoritmo de minimización, ya que de ésta forma se obtiene un autómata equivalente, pero mucho más eficiente.

Se brindará una explicación paso por paso del funcionamiento del algoritmo de minimización de AFD, en un primer momento. Luego, se procederá a aplicar el algoritmo en un ejemplo práctico.

2) Explicación del algoritmo de Minimización:

El algoritmo de minimización consiste en cinco pasos, aunque éstos a veces deben ser repetidos:

- Partición inicial
- Partición de nuevos conjuntos
- Verificar la nueva partición
- Eliminar los estados que se encuentren en el mismo grupo
- Remover estados sumidero y no alcanzables desde símbolo inicial

De los pasos anteriores, el segundo y el tercero forman un ciclo, mientras que los demás son procesos mecánicos que sólo se deben aplicar una vez. A continuación, detallaremos cada uno de estos pasos.

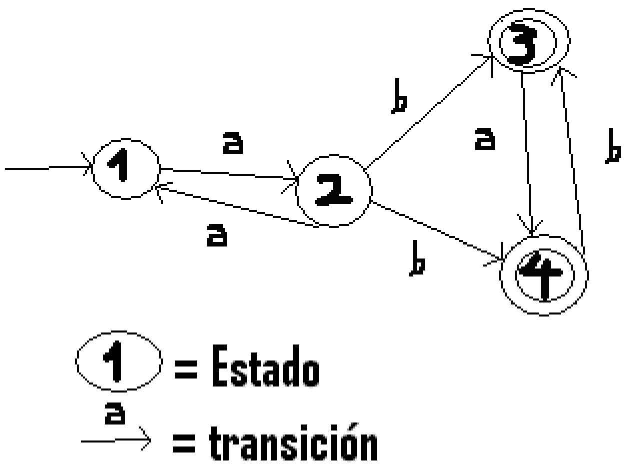
A partir de un autómata determinista (AFD), ya sea una tabla con los valores o su dibujo, se determinan dos conjuntos: uno que contiene a todos los estados finales y otro que contiene a los demás estados. Por ejemplo, en la tabla de abajo, los estados finales son aquellos que tienen “(f)” después del número (3 y 4) y los estados no finales son los demás (1 y 2). Para el dibujo, los estados finales son los que tienen doble círculo

(3 y 4), mientras que los no finales son los que tienen círculo sencillo (1 y 2). A este proceso se le conoce como partición inicial.

Estados \ Transiciones	a	b
1(i)	2	
2	1	4
3(f)	4	3
4(f)		3

(i)= estado inicial
(f)= estado final

Tabla de representación de un autómata finito determinista



Dibujo de representación de un autómata finito determinista

A partir de cada estado en cada uno de los conjuntos que se tiene, se estudia cuáles transiciones tienen, a cuáles estados se dirigen y a cuáles conjuntos pertenecen estos estados de llegada. Si dos estados del mismo conjunto se dirigen a estados de conjuntos distintos por medio de una de las transiciones, entonces estos estados deben separarse, formando así una nueva partición. A esto se le llama partición de nuevos conjuntos. Asimismo, si uno de los estados del conjunto tiene una transición hacia otro estado y los demás estados del conjunto no van a ninguna parte con el terminal de la transición, entonces el que sí la tiene debe separarse.

En estrecha relación con el paso anterior se encuentra el paso de verificación de la nueva partición. Este indica que se debe repetir la fase de partición hasta que los conjuntos ya no se puedan separar más. Los conjuntos no se pueden separar cuando quedan con únicamente un estado o cuando todos los estados del conjunto tienen transiciones hacia estados de los mismos conjuntos. Cuando ya no se puede realizar más particiones a los conjuntos, se sigue con el cuarto paso. Resulta de suma importancia considerar que cuando se separan dos estados de un conjunto, esto puede afectar las futuras particiones.

Se dispone ahora de una partición final. Si todos los estados están en conjuntos distintos, no hay que hacer nada. Si, por el contrario, hay varios estados en uno o varios de los conjuntos, estos estados deben fusionarse. La justificación es sencillamente que todos los estados de un conjunto cumplen una misma función y el autómata no se ve afectado si se unen. Para realizar esta fusión, se determina arbitrariamente un estado representante por cada conjunto. En la tabla del autómata, se reemplazan los demás estados del conjunto por el representante. Luego, se eliminan las filas repetidas.

Finalmente, se debe remover aquellos estados no finales que sólo tienen transiciones hacia ellos mismos o, en otras palabras, aquellos a partir de los cuales no se puede llegar a ningún estado final. A estos se les denomina estados sumideros. Asimismo, se debe eliminar los estados a los que no se puede llegar desde el estado inicial.

3) Ejemplo de Minimización de un Autómata Determinista

Se tiene la siguiente tabla de un autómata.

	1	0
A	B	G
B	B	C
C	A	E
D	D	G
[E]	E	E
F	G	F
G	G	G

El estado A es inicial y el estado E es final. Se aplicará el algoritmo de minimización descrito anteriormente.

Primer paso. Se separan los estados en estados finales y no finales. Tenemos los conjuntos: {A B C D F G} y {E}.

Segundo paso. A excepción de C, todos los estados tienen transiciones hacia los estados del primer conjunto {A B C D F G}. Como C se dirige a E por medio de 0, separamos a este estado. Nos quedan entonces los conjuntos: {A B D F G} {C} {E}. Los estados C y E han quedado solos en sus conjuntos y entonces no pueden ser separados más.

Se vuelve a realizar el paso 2. B se dirige a C mediante 0. Como C se encuentra en un conjunto diferente al de los demás estados de llegada a partir del conjunto {A B D F G}, hay que separar a B. Esto nos da: {A D F G} {B} {C} {E}. B ya no puede ser particionado.

Se vuelve a realizar el paso 2. A se dirige a B mediante 1 y B está en un grupo distinto del conjunto de llegada de {D F G}. Por lo tanto, A debe separarse del grupo. Nos queda: {D F G} {A} {B} {C} {E}. A no puede separarse más.

Tercer paso. {D F G} no se puede separar más ya que todos sus estados se dirigen a un mismo conjunto de estados ({D F G}).

Cuarto paso. Para el único conjunto con más de un estado {D F G}, se escoge el representante D. En la tabla, reemplazamos por D donde sea que aparezca F o G. Tenemos la siguiente tabla:

	1	0
A	B	D
B	B	C
C	A	E
D	D	D
[E]	E	E
D	D	D
D	D	D

Como resultado, se tiene tres filas idénticas. Por lo tanto, se elimina dos de ellas, obteniendo la siguiente tabla:

	1	0
A	B	D
B	B	C
C	A	E
D	D	D
[E]	E	E

Ahora, el autómata que se tiene no tiene estados sumidero y todos los estados son alcanzables a partir del estado inicial A. El autómata ha sido minimizado.

4) Conclusiones

El uso del algoritmo de minimización es de gran utilidad para disminuir el número de estados de un autómata determinista. Sin embargo, esto no asegura que el número de transiciones entre estados se reduzca. El algoritmo es bastante sencillo de aprender, aunque es necesario realizar mucha práctica para poder dominarlo. Para facilitar la obtención de un autómata mínimo se recomienda partir de un autómata simple si se da la oportunidad de crearlo a partir de una expresión regular. Entre más estados tenga un autómata, más complejo resultará y será más difícil de manejar.

Bibliografía:

-Aho, Alfred V & Sethi, Ravi & Ullman, Jeffrey D.: ***Compilers: Principles, Techniques and Tools***, Addison Wesley. 1979.
-Di Mare. Herramientas de curso Autómatas y Compiladores: http://www.di-mare.com/pub/AyC/