

# El Algoritmo para Reconocer Hileras en un Autómata Finito no Determinista

Astrid Carballo Salas<sup>1</sup>, Juan Pablo González Barrientos<sup>1</sup>, Eduardo Ramírez Acosta<sup>1</sup>, David Víquez Vílchez<sup>1</sup>,

<sup>1</sup>{astrid.carballo, juan.gonzalez, eduardo.ramirez, david.viquez}@ecci.ucr.ac.cr

**Resumen.** El fin primordial de los autómatas de cualquier tipo es evaluar y comprobar cadenas que le son insertadas, de manera que se compruebe mediante algunas reglas su validez o bien, se niegue la misma. Existen tipos de autómatas dependiendo de sus características, en este caso el interés se centra en los autómatas finitos no deterministas. Para validar hileras un autómata puede hacer uso de un algoritmo simple que consta de dos conjuntos de estados, y que avanza en la hilera que se quiere evaluar hasta llegar a un estado en que si bien la hilera se acepta, o se rechaza. Es importante el uso de este algoritmo para asegurarse que siempre el resultado sea correcto, ya que este es el fin de la existencia de algoritmos.

**Palabras claves:** Autómata, no determinista, validar hileras, algoritmo, cadenas.

## 1 Introducción

El presente artículo hace un análisis del algoritmo que se utiliza para validar hileras en un autómata finito no determinista. Define los conceptos más importantes que se deben conocer para entender este algoritmo, aplicarlo y analizarlo con profundidad para que quede claro cómo es que logra llegar a un resultado correcto la totalidad de las ocasiones en que se ejecuta [1].

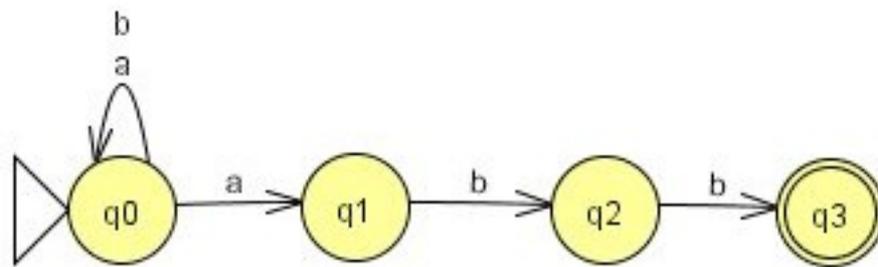
Para ampliar el conocimiento acerca de este algoritmo se hará uso del mismo con un ejemplo y de esta manera se podrá ahondar aún más en la utilidad y la validez que constituye. El principal objetivo es el de llegar a la mayor cantidad de personas posibles, para que utilicen este procedimiento como herramienta útil cuando sea requerido [1].

Es importante recalcar que este tipo de algoritmos pertenecen al área de las ciencias de la computación y de la informática concernientes a los compiladores y a los autómatas, sin embargo, no es solamente para esta área que este tipo de teoría se utiliza, por ejemplo se utiliza también a la hora de realizar traducciones entre distintos tipos de lenguaje [1].

## 2 Autómatas finitos no deterministas

Los autómatas finitos son modelos matemáticos que se utilizan para aceptar o rechazar entradas que se les proporcionan. En pocas palabras son sistemas que reciben cadenas y luego de procesarlas indican si es válida o no de acuerdo a la estructura que se le proporcionó al mismo autómata [1].

Los autómatas se pueden representar mediante diagramas de transición, en donde los nodos son estados.



**Fig. 1.** Diagrama de transición que representa un autómata [2].

Existen dos tipos de autómatas finitos, los deterministas y los no deterministas, para el enfoque de interés solo se tomarán en cuenta los autómatas finitos no deterministas. Un autómata finito no determinista (AFN) consiste en:

- Un conjunto finito de estados  $S$ .
- Un conjunto de símbolos de entrada  $\Sigma$ , el alfabeto de entrada. Se supone que  $\epsilon$ , representa a la hilera vacía y nunca forma parte de  $\Sigma$ .
- Una función de transición que proporciona, para todos los estados y símbolos en  $\Sigma$  unido con  $\epsilon$ , un conjunto de estados siguientes.
- Un estado inicial que pertenece a  $S$  ( $s_0$ ).
- Un conjunto  $F$  de estados, que pertenecen a  $S$  y que se reconocen como estados de aceptación o finales [1].

En la Fig. 1 se reconocen los siguientes elementos anteriormente explicados:

- $q_0$  es  $s_0$  o el estado inicial.
- $\{q_0, q_1, q_2, q_3\}$  representan elementos del conjunto de estados  $S$ .
- $\{a, b\}$  representan los símbolos de entrada  $\Sigma$ .
- $q_3$  es el único estado en este caso que pertenece a  $F$ , ya que es el único estado de aceptación [1].

Es posible representar un autómata finito no determinista por medio de la llamada tabla de transición. Una tabla de transición es aquella cuyas columnas son asignadas a

los símbolos de entrada del autómata, y al símbolo de la hilera vacía  $\epsilon$ , y sus filas corresponden a los estados del autómata [1].

Para asignar un valor a una entrada de la tabla se aplica la función de transición para el estado y el símbolo, en caso de que la función de transición no defina información para este par de elementos, se coloca el símbolo de conjunto vacío  $\emptyset$ .

**Tabla 1.** Transiciones para el autómata finito no determinista de la Fig. 1 [1].

Estado	a	b	c
0	{0,1}	{0}	$\emptyset$
1	$\emptyset$	{2}	$\emptyset$
2	$\emptyset$	{3}	$\emptyset$
3	$\emptyset$	$\emptyset$	$\emptyset$

Para que una cadena sea aceptada por un autómata finito no determinista (y en general para cualquier tipo de autómata), es necesario que siendo “y” la cadena por analizar, sea posible mediante el autómata, deletrear “y”, es decir, que exista un camino en el autómata desde el estado inicial hasta un estado de aceptación, de forma que los símbolos que pertenecen al camino sean los mismos que pertenecen a “y”, respetando el orden. El símbolo especial “ $\epsilon$ ” se utiliza para ser ignorado si pertenece al camino que acepta la hilera [1].

El siguiente ejemplo muestra como el autómata de la Fig. 1 acepta una cadena. Considérese la cadena “aabb”; ahora tomando en cuenta el estado inicial del autómata que en este caso es  $q_0$ , se observa que existe un camino del estado  $q_0$  a él mismo, con el primer símbolo de la hilera que sería “a”, de manera que se puede consumir ese símbolo y seguir en el estado  $q_0$ . Considerando el segundo símbolo de la hilera que nuevamente es “a”, se observa que existe otro camino que sale de  $q_0$  por medio del símbolo “a”, y que llega al estado  $q_1$ , con lo cuál se consume este símbolo del análisis y se avanza al estado  $q_1$  del autómata. El siguiente símbolo de la cadena que se analiza es “b”, y es notorio que el único camino que sale del estado  $q_1$  es por medio del símbolo “b” al estado  $q_2$  [1]. Finalmente el símbolo último de la hilera es “b” por medio del cuál se puede avanzar del estado  $q_2$  al  $q_3$  y consumir el último símbolo, y dado el caso que se consumió totalmente la hilera que se analizaba y que el estado actual es un estado de aceptación ( $q_3$ ), entonces el autómata validó con éxito la hilera [1].

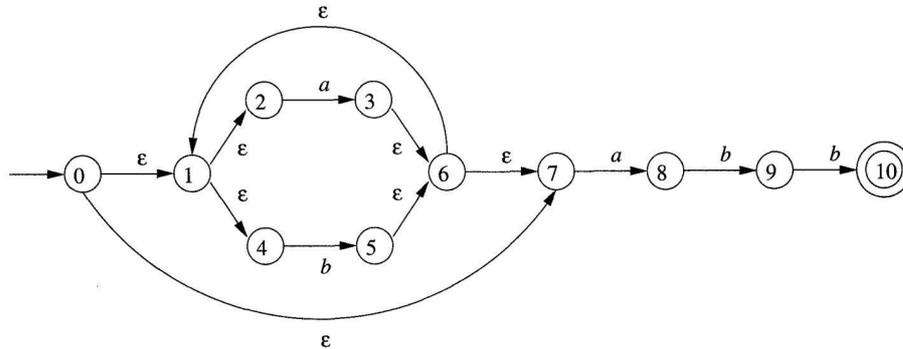
Surge la necesidad de definir el concepto de lenguaje definido o aceptado por un autómata, puesto que se refiere al conjunto total de hileras que el autómata acepta, es decir, que etiquetan cierto camino desde el estado inicial a un estado de aceptación. En el caso del autómata de la Fig. 1, cuyo alfabeto de entrada contiene a los símbolos {a,b}, el conjunto de hileras válidas o lenguaje, es el de todas las hileras que contengan estos símbolos y que terminen con la subhilera “abb” [1].

### 3 Algoritmo para aceptación de hileras

Un autómata finito no determinista puede representar un árbol de caminos posibles desde el estado inicial bastante extenso, inclusive es posible que ocurra que una hilera válida para el autómata también contenga otro camino en el autómata, el cuál no concluya con un estado de aceptación, por lo que una hilera válida podría ser erróneamente invalidada, por ejemplo considérese el ejemplo de la hilera "aabb": comenzando con el estado inicial  $q_0$  es posible trasladarse al estado  $q_0$  por medio del símbolo "a", luego es posible mediante la siguiente "a" volver al estado  $q_0$ , y por medio del siguiente par de símbolos "b" se pueden realizar dos movimientos idénticos que con las "a", ya que el estado  $q_0$  contiene transiciones hacia él mismo, por medio del símbolo "a" y el símbolo "b". Al finalizar este análisis es posible apreciar cómo la hilera sí define un camino en el autómata que no finaliza en un estado de aceptación, por lo tanto no es un camino válido, pero como ya se explicó anteriormente la hilera sí es válida en el autómata [2].

Es posible imaginar el problema que podría surgir en autómatas más grandes y con alfabetos más extensos a la hora de reconocer hileras, si no se cuenta con un mecanismo infalible, que funcione con cualquier entrada y que esté bien definido, en otras palabras, un algoritmo bien establecido [2].

Se definen dos conjuntos compuestos por estados, uno con el nombre de *núcleo* y el otro con el nombre de *cola*. En el primer paso se procede a llenar un primer par de conjuntos núcleo-cola; en el núcleo se inserta el estado inicial, y en la cola se agregan todos los estados a los que se es posible acceder desde el estado inicial por medio de la hilera vacía o " $\epsilon$ " y también se agregan todos los estados a los que es posible llegar por medio de " $\epsilon$ " desde los estados que ya pertenecen a la cola, siempre y cuando no sean parte ya de la misma cola. Una vez que se tiene este primer par de conjuntos núcleo-cola se procede a rellenar un segundo par del mismo tipo de la siguiente manera: para cada estado que pertenece al núcleo anterior y a la cola anterior, se verifica a cuáles estados se pueden acceder mediante el primer símbolo de la cadena que se quiere validar y se agregan al nuevo núcleo, luego se llena la nueva cola, de la misma manera que la anterior pero con el núcleo actual, esto es, con cada estado que pertenezca al núcleo actual se agregan a la cola actual los estados a los que puedo llegar por medio de " $\epsilon$ " desde ese estado y también se agregan los estados que son accesibles por medio de " $\epsilon$ " desde un estado que pertenece a la cola actual, pero que todavía no pertenezcan a esta cola. Es necesario realizar este paso para cada uno de los símbolos de la hilera. En el momento en que hayan producido los núcleos y las colas para cada símbolo de la hilera, se verifica si el último par contiene (ya sea en su núcleo o en su cola), un estado de aceptación, si esta condición se cumple, es porque la hilera es válida. Si en algún momento no se logra avanzar más porque el núcleo quedó vacío, o si no se cumple la condición de que el último contenga un estado de aceptación es porque la hilera no es válida para ese autómata, o en otras palabras, no la contiene su lenguaje.



**Fig. 2.** Autómata finito no determinista [2].

Considérese el autómata de la Fig. 2, para lograr una mayor claridad se procede a explicar como funciona el algoritmo para validar hileras con este autómata cuando intenta validar la hilera “aabbabb”. Como lo indica el algoritmo, el primer paso consiste en formar un par de conjuntos de estados núcleo-cola, y en este caso se llenan respectivamente con el estado inicial y con todos los estados accesibles desde este estado por medio del símbolo “ $\epsilon$ ” [2].

$$\begin{aligned} \text{Núcleo} &= \{0\} \\ \text{Cola} &= \{1,2,4,7\} \end{aligned} \tag{1}$$

Los conjuntos anteriores se definen de esta manera porque el estado inicial de este autómata es “0” (por esta razón pertenece al primer núcleo). El conjunto cola contiene al estado “1” y el estado “7” porque existen dos enlaces diferentes entre el estado “0” y ambos, por medio del símbolo “ $\epsilon$ ”, lo cuál concuerda con el algoritmo; el estado “2” se agrega a la cola ya que es posible acceder por medio de “ $\epsilon$ ” desde el estado “1”, que ya pertenece a la cola, al estado “2”, y de igual manera ocurre con el estado “4” que también se agrega a la cola por tener un enlace desde el estado “1” a él por medio de “ $\epsilon$ ” [2].

El segundo par núcleo-cola se rellena de la siguiente manera:

$$\begin{aligned} \text{Núcleo} &= \{3,8\} \\ \text{Cola} &= \{6,7,1,2,4\} \end{aligned} \tag{2}$$

El motivo por el que ocurre este relleno es el siguiente: se toma en cuenta el par anterior y el primer símbolo de la hilera que es “a”; se observa que su núcleo contiene al estado “0”, sin embargo por medio del símbolo “a” y desde el estado “0”, no existe ningún enlace, por lo que se procede a analizar la cola, y se observa que contiene al estado “1”, el cuál en este caso no aporta ninguna información importante puesto que desde él solo existen enlaces por medio del símbolo “ $\epsilon$ ” y no por medio del símbolo “a”, por lo que se continúa en la cola hacia el estado “2” y se nota que desde el estado “2”, es posible acceder al estado “3” por medio del símbolo “a”, el cuál es el objeto de análisis actual, y esta es la razón de que en el núcleo actual se

presente el estado 3. Aunque se haya insertado un estado en el núcleo, la cola anterior aún contiene estados que no han sido analizados, verbigracia el “4”, pero este tampoco aporta ningún estado nuevo al núcleo que se está rellenando puesto que a partir del estado “4”, en el autómata, no existe otro enlace que no sea por medio del símbolo “b”, y como el símbolo que se está analizando es “a” y no “b”, este estado no interesa por el momento. El último estado que se debe considerar de la cola es el estado “7”, y como se observa en el autómata es posible trasladarse desde él hasta el estado “8” por medio del símbolo “a” el cuál es el símbolo que se está analizando en la hilera, y es por esto que el núcleo actual contiene entre sus estados a “8”. Continuando con el algoritmo se procede a rellenar la cola actual; el primer estado que se agrega es “6” ya que es posible llegar desde el estado “3” que pertenece al núcleo actual, al estado “6” por medio de “ε”. El estado “7” forma parte de la cola puesto que es posible llegar desde el estado “6” al estado “7” por medio de “ε” y el estado “6” pertenece a la cola actual (es importante recalcar que “7” no se hubiera agregado si ya hubiera sido agregado con anterioridad). Desde el estado “6” que pertenece a la cola actual es posible llegar por medio de “ε” al estado “1” por lo que este estado también forma parte de la cola actual. Los estados “2” y “4” forman parte del conjunto de estados denominado como cola actual, puesto que es posible llegar a ellos desde el estado “1” que ya es parte de la cola actual [2].

El tercer par núcleo-cola se completa de la siguiente manera:

$$\begin{aligned} \text{Núcleo} &= \{8,3\} \\ \text{Cola} &= \{1,2,4,6,7\} \end{aligned} \quad (3)$$

Se explica ahora el porqué de esta composición de conjuntos: se toma el siguiente símbolo de la hilera que se está analizando, que en este caso es nuevamente el símbolo “a”. Por medio del núcleo anterior se observa que contiene a “8”, el cuál sólo se enlaza por medio de “b” y no de “a” por lo que no es de interés en este momento, y por otro lado el estado “3” que también pertenece al núcleo anterior solo posee un enlace por medio del símbolo “ε”, por lo que tampoco aporta en este caso ninguna información importante para el algoritmo en este momento.

Por medio de la cola anterior se observa que su primer estado “6”, solo contiene enlaces por medio de “ε”, a diferencia de “7” desde el cuál es posible llegar al estado “8” por medio de “a” y dado que este es el símbolo que se está analizando, se procede a agregar a “8” al núcleo actual. Los estados “1” y “4” no aportan estados porque el estado “1” solo contiene enlaces por medio de “ε” y el estado “4” por medio de “b”, en cambio el estado “2” aporta al núcleo actual el estado “3” porque es posible acceder a él por medio de “a” desde “2” [2].

Una vez que se tiene listo el núcleo de esta iteración del algoritmo se procede a llenar la cola, y dado el caso que el núcleo es idéntico a uno que se obtuvo con anterioridad se puede proceder a trasladar la cola de este, para ser más preciso, es válido asumir que dos núcleos iguales van a poseer colas idénticas, esto ayuda a ahorrar tiempo a la hora de ejecutar el algoritmo y lo simplifica de gran manera cuando el algoritmo produce muchos núcleos repetidos [2].

El cuarto par de conjuntos se completan de la siguiente manera:

$$\begin{aligned} \text{Núcleo} &= \{9,5\} \\ \text{Cola} &= \{6,7,1,2,4\} \end{aligned} \quad (4)$$

Esta conformación surge por las siguientes razones: se prosigue con el siguiente símbolo por analizar que sería en este caso “b”, una vez aclarado esto se continúa analizando el conjunto núcleo-cola anterior, comenzando con el núcleo, que como se puede observar contiene al estado “8”, y en este caso se nota que existe un enlace en el autómata que sale desde “8” hasta “9” por medio del símbolo “b” y como este es el símbolo que se está analizando, se procede a agregar “9” al núcleo actual. Continuando con el núcleo anterior el estado “3” solo contiene un estado con “ε” por lo que no se agrega nada por causa de “3”. Continuando con la cola anterior se nota que el estado “1” solo contiene enlaces de tipo “ε”, por lo que no se agrega nada. Prosiguiendo se toma ahora el estado “2” que es el siguiente de la cola anterior y se observa que solo contiene un enlace por medio del símbolo “a”, el cuál no es actualmente el objeto de análisis, y por esta razón el estado “2” no aporta un estado nuevo. El estado siguiente (“4”) sí aporta un estado nuevo, el estado “5”, ya que se observa que existe un enlace entre “4” y “5” por medio del símbolo “b” y por esto es que el estado “5” forma parte del núcleo actual. Los estados “6” y “7” que son los restantes de la cola anterior que no se han analizado, no aportan estados nuevos al núcleo puesto que “6” solo contiene un enlace con “ε” y “7” por medio de “b”.

Para la cola actual se observa que, analizando el núcleo actual, desde “9” no es posible llegar a ningún estado por medio de “ε” sin embargo desde “5” sí lo es, y por esto se agrega a la cola el estado “6”. Como se acaba de agregar el estado “6” a la cola actual, se debe también tomar en cuenta los enlaces que existen por medio de “ε” saliendo desde “6” hasta cualquier estado que no sea parte de la cola, en este caso “7” y “1” y esta es la razón de que sean parte de la cola [2]. Continuando con el algoritmo desde “7” no se puede llegar a ningún otro estado por medio de “ε”, sin embargo desde “1” existe la posibilidad de acceder a los estados “2” y “4” por medio de “ε” y por causa de esto son parte de la cola [2].

Los siguiente pares núcleo-cola quedar completos de la siguiente manera siguiendo las reglas ya expuestas:

Con la siguiente “b”:

$$\begin{aligned} \text{Núcleo} &= \{10, 5\} \\ \text{Cola} &= \{6, 7, 1, 2, 4\} \end{aligned} \quad (5)$$

Con la siguiente “a”:

$$\begin{aligned} \text{Núcleo} &= \{8, 3\} \\ \text{Cola} &= \{1, 2, 4, 6, 7\} \end{aligned} \quad (6)$$

Con la siguiente “b”:

$$\begin{aligned} \text{Núcleo} &= \{9, 5\} \\ \text{Cola} &= \{1, 2, 4, 6, 7\} \end{aligned} \quad (7)$$

Con la última “b”:

$$\begin{aligned} \text{Núcleo} &= \{ 10, 5 \} \\ \text{Cola} &= \{ 1, 2, 4, 6, 7 \} \end{aligned} \quad (8)$$

Como es notable, en el último núcleo se encuentra el estado 10, que pertenece al conjunto de estados de aceptación del autómata, por ende se puede concluir con toda seguridad que la hilera es válida para este autómata [2].

Si en algún momento durante el algoritmo hubiera sido imposible avanzar porque el núcleo estuviera vacío o porque en la última letra que se analizó no se produjo una cola o un núcleo en cuyos estados se encontrara el estado de aceptación "10", hubiera sido un indicativo de que la hilera no es válida o que no pertenece al lenguaje de este autómata.

Es intuitivo el porqué funciona el algoritmo presentado, pues va construyendo la totalidad de caminos que pueden formarse para la hilera que se quiere aceptar o rechazar, por lo que es válido concluir que no interfiere el hecho de que exista un camino o varios (o ninguno) que contenga o delecteen a la hilera que se quiere evaluar, pues el algoritmo explora todas las posibilidades que ofrece el autómata, y el hecho de que se exija que el estado de aceptación se encuentre en el núcleo o la cola del último paso, indica que por algún camino se puede llegar a él, respetando el orden y los símbolos que establece la hilera [2].

## 4 Conclusiones

Un autómata es un modelo matemático que se utiliza para validar hileras, de manera que se puedan incluir o no en un lenguaje, que es el conjunto total de hileras que acepta un autómata. Para validar una hilera un autómata toma la hilera que se desea analizar y descubre si en él mismo existe algún camino que delectee a esa hilera.

Es importante mencionar que no sólo para la teoría de compiladores se utilizan los autómatas y sus temáticas, sino que en otros campos de las ciencias de la computación e informática son de invaluable ayuda para resolver problemas, verbigracia, la traducción entre diferentes lenguajes de programación.

Existen diferentes tipos de autómatas, entre los cuáles se cuentan los autómatas finitos no deterministas que cumplen una serie de características que los identifican, verbigracia el hecho de que pueden contener el símbolo "ε", que representa a la hilera vacía.

Puede ocurrir que un autómata contenga dos o más caminos que delecteen a la hilera que se quiere evaluar, por lo que surge la necesidad de un algoritmo establecido para evaluar hileras, de manera que se pueda tener siempre un resultado correcto.

El algoritmo que se utiliza hace uso de dos conjuntos de estados, que descubren todos los posibles caminos que existen para la hilera que se quiere validar, si tal camino no existe o si no termina en un estado de aceptación, la hilera no es válida.

## Referencias

1. Aho, A.V., Lam, M., Sethi, R., Ullman, J.D.: *Compiladores: Principios, Técnicas y Herramientas*. Addison-Wesley, México D.F. (2008)
2. Sitio en línea del curso Autómatas y Compiladores, impartido por el profesor Adolfo Di Mare Hering, <http://www.di-mare.com/pub/AyC>