



TECNOLÓGICO
NACIONAL DE MÉXICO®



TECNOLÓGICO NACIONAL DE MEXICO

INSTITUTO TECNOLÓGICO DE
IZTAPALAPA

INTEGRANTES:

GUTIERREZ ARELLANO RAFAEL

181080022

ISC-6AM

LENGUAJES Y AUTOMATAS I

M.C. ABIEL TOMÁS PARRA HERNÁNDEZ

SEP 2020 / FEB 2021



ACTIVIDAD SEMANA 6

GUTIERREZ ARELLANO RAFAEL

Gramáticas regulares

Las gramáticas formales definen un lenguaje describiendo cómo se pueden generar las cadenas del lenguaje.

Una gramática formal es una cuadrupla $G = (N, T, P, S)$ donde:

- N es un conjunto finito de símbolos no terminales
- T es un conjunto finito de símbolos terminales $N \cap T = \emptyset$
- P es un conjunto finito de producciones Cada producción de P tiene la forma $\alpha \rightarrow \beta$, $\alpha = \varphi A \rho$ y $\beta = \varphi \omega \rho$ $\varphi, \omega, \rho \in (N \cup T)^*$ y A es S ó $A \in N$
- S es el símbolo distinguido o axioma $S \notin (N \cup T)$

Restringiendo los formatos de producciones permitidas en una gramática, se pueden especificar cuatro tipos de gramáticas (tipo 0, 1, 2 y 3) y sus correspondientes clases de lenguajes.

Gramáticas regulares (Tipo 3)

Generan los lenguajes regulares (aquellos reconocidos por un autómata finito). Son las gramáticas

más restrictivas. El lado derecho de una producción debe contener un símbolo terminal y, como máximo, un símbolo no terminal. Estas gramáticas pueden ser:

- Lineales a derecha, si todas las producciones son de la forma

Existe un algoritmo que permite obtener una gramática regular que genera un lenguaje regular dado a partir del autómata finito que reconoce ese lenguaje. Los pasos a seguir son los siguientes:

- 1) Asociar al estado inicial el símbolo distinguido S .
- 2) Asociar a cada estado del autómata (menos el estado inicial) un símbolo no terminal. Si al estado inicial llega algún arco asociar también un símbolo no terminal (además del símbolo distinguido). No asociar símbolo no terminal a aquellos estados finales de los que no salen arcos.



- 3) Para cada transición definida $\delta(e_i, a) = e_j$, agregar al conjunto de producciones, la producción $A \rightarrow aB$, siendo A y B los símbolos no terminales asociados a e_i y e_j respectivamente. Si e_j es un estado final, agregar también la producción $A \rightarrow a$. Si e_j es el estado inicial (tiene dos símbolos asociados, el distinguido y uno no terminal), utilizar el símbolo no terminal (de esta manera se evita que el símbolo distinguido aparezca a la derecha de una producción).
- 4) Si el estado inicial es también final agregar la producción $S \rightarrow \epsilon$

Autómata finito determinista.

Un autómata finito determinista es un autómata finito que además es un sistema determinista; es decir, para cada estado en que se encuentre el autómata, y con cualquier símbolo del alfabeto leído, existe siempre no más de una transición posible desde ese estado y con ese símbolo.

Este tipo de autómatas admite su definición de dos maneras bien diferentes:: Como autómatas traductores o reconocedores. La definición como autómatas traductores continua a la definición de las máquinas secuenciales, y se los podría definir como una subclase de estas, ya que los autómatas finitos tendrían como limitante no poder iniciar desde cualquier estado como lo hacen en las máquinas secuenciales.

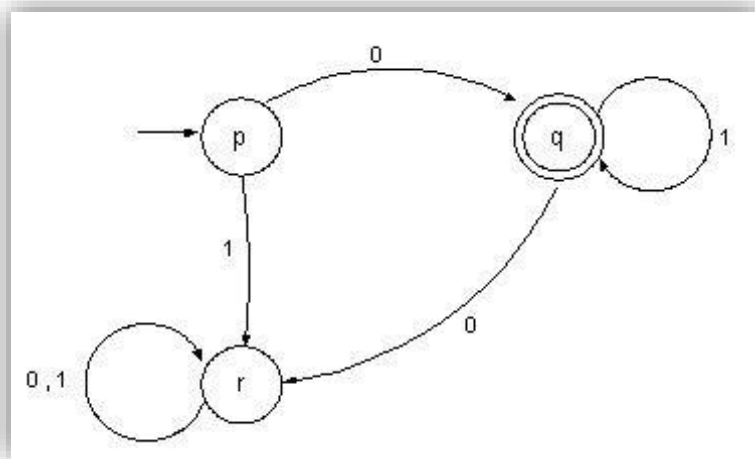
La forma que adoptaremos para la definición de los autómatas finitos deterministas es como autómatas reconocedores, ya que se ajusta con los contenidos de la informática teórica y utilización que se les da dentro del diseño de los analizadores léxicos.

Estos autómatas sólo se limitarán a aceptar o no una determinada cadena recibida en la entrada, por lo tanto, podemos decir que la salida de los mismos solo tendrá dos valores posibles aceptar o no aceptar a la palabra de entrada.

Una característica importante de este tipo de autómatas es el determinismo, lo cual significa que estando en un estado y recibiendo una entrada del exterior el autómata tendrá la posibilidad de transitar a uno y solo un estado del conjunto de estados posibles.



Con respecto al conjunto de estados ahora se pueden clasificar en tres tipos: Estado Inicial, que es por donde comenzará la ejecución de la máquina; Estados finales o de aceptación que será un subconjunto del conjunto de estados por los que transitó la máquina, y si cuando se hayan terminado de procesar todos los símbolos de entrada y no reste ningún símbolo por leer, la máquina quede posicionada en uno de estos estados de aceptación, se concluirá que la cadena procesada será aceptada por el autómata. y Estados Intermedios, que tienen comportamiento idéntico a los definidos en las máquinas secuenciales.



Autómata finito no determinista.

Un autómata finito no determinista es un autómata finito que, a diferencia de los autómatas finitos deterministas, posee al menos un estado $q \in Q$, tal que para un símbolo $a \in \Sigma$ del alfabeto, existe más de una transición δ posible.

La definición formal de AFND se basa en la consideración de que a menudo según los algoritmos de transformación de expresiones y gramáticas regulares a AF terminan obteniéndose autómatas con transiciones múltiples para un mismo símbolo o transiciones vacías. Independientemente que sean indeseables, sobre todo para la implementación material, fundamentalmente mecánica, de los autómatas finitos, son imprescindibles durante la modelación de analizadores lexicográficos de los elementos gramaticales de los lenguajes de programación, llamados tókens, como literales numéricos, identificadores, cadenas de texto, operadores, etc.

Los AFND son definiciones no tan deseables dentro de los lenguajes regulares porque dificultan su implementación tanto mecánica como informática; aunque en la mayoría de las transformaciones a lo interno de los LR (expresiones regulares a AF, gramáticas regulares a AF) conducen a AFND. Los AFND, por tanto, son imprescindibles en el análisis lexicográfico y el diseño de los lenguajes de programación.