Sintaxis y Semántica del Lenguaje

Contenidos

Introducción a Autómatas y Lenguajes

Aplicación JFLAP para generación y testing de AF

Dado un lenguaje, como generamos un AFD

Resolución de ejercicios fuera del TP 3

Resolución de ejercicios TP 3

Autómatas finitos

Definición:

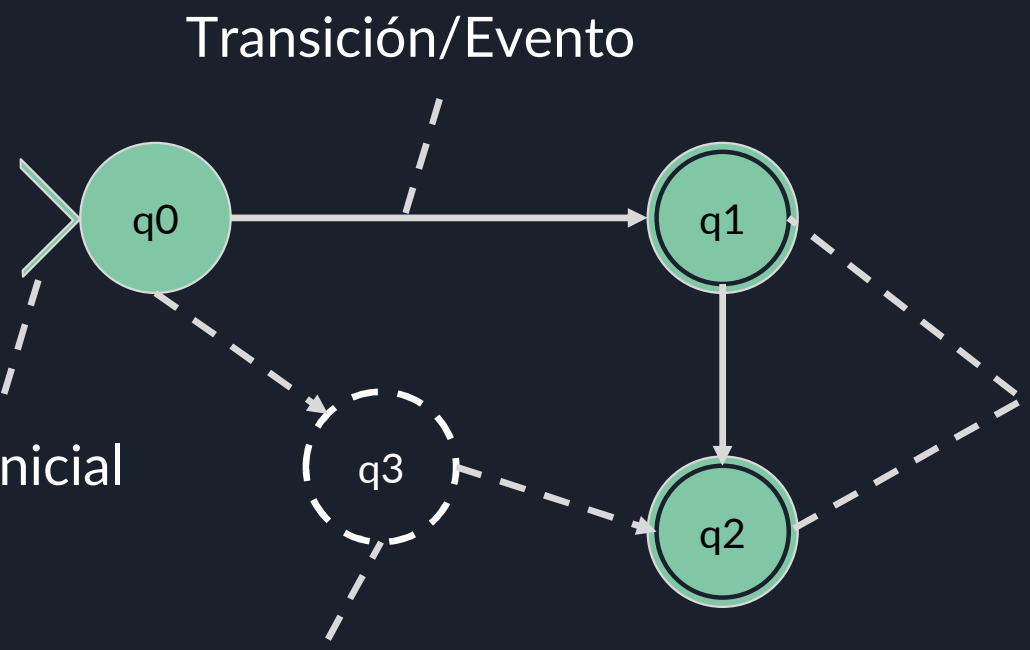
Las máquinas de estados más sencillas son los autómatas finitos. Estos, son un un modelado gráfico de una máquina abstracta que permite representar una serie de acciones o eventos y de estados.

Tiene forma de grafo dirigido. Vamos a verlo gráficamente →

Autómatas finitos

Un grafo se compone de ***vértices*** y de ***aristas*** que los conectan. En un grafo dirigido, las aristas cuentan con una dirección (tal como indica la palabra). Es decir, se recorren en determinado sentido.

Autómatas finitos: Características



¿Que indica cada parte? →

Autómatas Finitos: Características

Las ***transiciones*** ***o*** ***eventos*** representan situaciones instantáneas que marcan un cambio de un estado a otro.

Un ***estado inicial*** marca el inicio del recorrido. Siempre se empieza a recorrer a partir de este estado

Un ***estado final*** (pueden tener más de uno) marca que la **secuencia** ingresada es **válida**.

Autómatas Finitos: Propósito

Sirven para reconocer ***secuencias de acciones legales o inválidas*:**

Si al recorrer el grafo con la secuencia de entrada se llega a un ***estado final***, la secuencia es válida. Si se llega a un estado ***no final*** es inválida.

Al camino recorrido se lo llama ***Trayectoria.***

Lenguajes

Un lenguaje L es un **conjunto de palabras** y está **definido sobre un alfabeto ∑**.
Como los lenguajes son conjuntos pueden ser vacíos, de un solo elemento, de muchos elementos. En realidad, de infinitos elementos.

Ejemplos:
L={ca} es un lenguaje
L={ca, sa} es otro lenguaje

En todo los casos el lenguaje L es un subconjunto de $\sum^*$.
$\sum^*$ = {Ɛ, ca, sa, casa, saca, sasa, casasa, sacasa,....}

Autómatas y Lenguajes

Los autómatas finitos pueden aplicarse sobre cualquier lenguaje formal L, ya sea para validar si alguna palabra w pertenece a él, o para generar palabras que lo hagan.

Se puede modelizar ***asignando, a cada transición, una letra del alfabeto*** sobre el que está definido. A su vez, cada nodo representa una ***regla de formación***.

Entonces, para verificar la validez de la palabra, comenzamos verificando que el caracter ingresado pertenezca al alfabeto. Luego se valida si la posición que ocupa el carácter en la palabra respeta las reglas de definición del mismo.

Autómatas y Lenguajes

Entonces, ingresando uno a uno los caracteres de la palabra, el autómata intenta encontrar una trayectoria para la misma (que haya transiciones que me lleven de qn a qm para cada carácter). Si al finalizar de recorrer el grafo con la palabra, se llega a un estado final, es considerada legal.

Autómatas Finitos: Clasificación

Los autómatas finitos tienen un conjunto de estados y su "control" pasa de uno a otro en base a las entradas.
Podemos clasificar los autómatas, según su control, en **determinísticos (AFD)** y **no determinísticos (AFN)**.

Un autómata es tiene un control "determinista" cuando, después de leer una secuencia de entradas, puede estar ***únicamente*** en un estado. Por lo tanto, el control será ***no determinista*** si puede estar en más de un estado a la vez.

AFD - Definición Formal

Un AFD es una ***quíntupla*** (K, S, F, ∑, δ) donde:

K es el conjunto de todos los estados posibles

S es el estado inicial

F es el conjunto de estados finales

∑ es el alfabeto dado sobre el que se construye el lenguaje L

δ es la función de transición, que a partir de un estado y un símbolo del alfabeto se llega a un nuevo y único estado

AFD - Particularidades

Los estados deben ser ***excluyentes entre sí***. Es decir, estoy en un nodo ó en otro, no puedo estar con una misma secuencia en dos nodos distintos.

Cada nodo debe tener ***una transición por cada letra del alfabeto***. Si tengo un alfabeto conformado por ‘a’, ‘b’ y ‘c’, todos los nodos deben tener una “flecha” con ‘a’, una con ‘b’, y una con ‘c’.

Ejercicio

Diseñar el AFD que acepte palabras en el alfabeto ∑={a,b} que empiecen con la letra ‘a’.

K =
S =
F =
∑ =
δ =

Ejercicio

Diseñar el AFD que acepte palabras en el alfabeto ∑={a,b,c} que terminen con ‘bb’.

K =
S =
F =
∑ =
δ =

Ejercicio

Diseñar el AFD que acepte palabras en el alfabeto ∑={a,b,c} que no contengan la subcadena ‘ac’.

K =
S =
F =
∑ =
δ =

Ejercicio

Diseñar el AFD que acepte palabras en el alfabeto ∑={a,b,c} que acepte el lenguaje L={w/w contiene cantidad par de ‘a’}.

K =
S =
F =
∑ =
δ =

JFLAP

Instalar:

Paquete de Java: https://javadl.oracle.com/webapps/download/AutoDL?BundleId=242029_3d5a2bb8f8d4428bbe94aed7ec7ae784

JFLAP: http://www.jflap.org/jflaptmp/july27-18/JFLAP7.1.jar