

Conversión de un Autómata Finito No Determinista (AFND) a Autómata Finito Determinista (AFD)

Un autómata finito no determinista (abreviado AFND) es un autómata finito que, a diferencia de los autómata finitos deterministas (AFD), posee al menos un estado $q \in Q$, tal que para un símbolo $a \in \Sigma$ del alfabeto, existe más de una transición $\delta(q,a)$ posible. Todo AFND puede ser convertido en un AFD equivalente.

En un AFND puede darse cualquiera de estos dos casos:

- Que existan transiciones del tipo $\delta(q,a)=q_1$ y $\delta(q,a)=q_2$, siendo $q_1 \neq q_2$;
- Que existan transiciones del tipo $\delta(q, \epsilon)$, siendo q un estado no-final, o bien un estado final pero con transiciones hacia otros estados.

Cuando se cumple el segundo caso, se dice que el autómata es un autómata finito no determinista con transiciones vacías o transiciones ϵ (abreviado AFND- ϵ). Estas transiciones permiten al autómata cambiar de estado sin procesar ningún símbolo de entrada. Considérese una modificación al modelo del autómata finito para permitirle ninguna, una o más transiciones de un estado sobre el mismo símbolo de entrada.

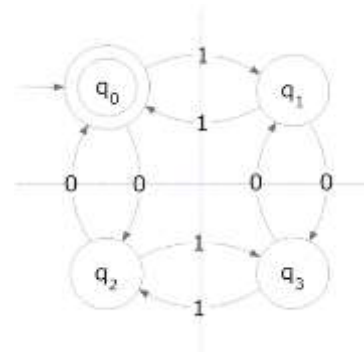
El siguiente es un ejemplo sencillo para ilustrar estos pasos:

Supongamos que tienes el siguiente AFND:

AFND = $(Q, \Sigma, \delta, q_0, F)$

Donde:

- Q : Conjunto de estados.
- Σ : Alfabeto.
- δ : Función de transición.
- q_0 : Estado inicial.
- F : Conjunto de estados finales.



La máquina comienza en el estado inicial especificado y lee una cadena de caracteres pertenecientes al alfabeto. El autómata utiliza la función de transición de estados T para determinar el siguiente estado, usando el estado actual y el símbolo que acaba de leer o la cadena vacía. Sin embargo, "el estado siguiente de un AFND no solo depende de el evento de entrada actual, sino que también en un número arbitrario de los eventos de entrada posterior. Hasta que se producen estos acontecimientos posteriores no es posible determinar en qué estado se encuentra la máquina". Cuando el autómata ha terminado de leer, y se encuentra en un estado de aceptación, se dice que el AFND acepta la cadena, de lo contrario se dice que la cadena de caracteres es rechazada. Tanto para un AFND como para un autómata finito determinista (AFD) se puede aceptar el mismo lenguaje. Por lo tanto, es posible convertir un AFND existente en un AFD para el desarrollo de una máquina tal vez

más simple. Esto puede llevarse a cabo utilizando la construcción del conjunto potencia, que puede conducir a un aumento exponencial en el número de estados necesarios.

Hay muchas formas de implementar una AFND:

- Convertir al equivalente AFD: en algunos casos esto puede causar un aumento exponencial en el tamaño del autómata, y así un espacio auxiliar proporcional al número de estados en el AFND (como el almacenamiento del valor del estado requiere en la mayoría de un bit por cada estado en el AFND).
- Mantener un conjunto de datos de todos los estados en que la máquina podría estar en la actualidad. Al consumir el último carácter de entrada, si uno de estos estados es un estado final, la máquina acepta la cadena. En el peor de los casos, esto puede requerir espacio adicional proporcional al número de estados en el AFND; si la estructura del conjunto usa un bit por estado del AFND, entonces esta solución es exactamente equivalente a la anterior.
- Crear múltiples copias. Por cada n forma de la decisión, el AFND crea hasta $n-1$ copias de la máquina. Cada uno de ellos entrara en un estado independiente. Si, al momento de consumir el último símbolo de la entrada, al menos una copia del AFND está en un estado de aceptación, el AFND lo aceptará. (Esto también requiere un almacenamiento lineal con respecto al número de estados del AFND, ya que puede haber una máquina por cada estado del AFND).