Modelado de propiedades con autómatas finitos

Parte II: Autómata finito no determinista

Teoría de Lenguajes

2020

Introducción

Esta es la segunda parte de dos documentos cuyo objetivo es brindar ejemplos explicados paso a paso sobre el modelado de propiedades usando autómatas finitos.

En esta segunda parte veremos un ejemplo utilizando el modelo de **autómata finito no determinista** (*AFND*). Pero primero, veamos qué nos aporta ese modelo.

El no-determinismo

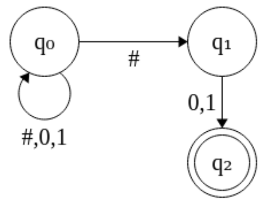
Decimos que un autómata presenta no-determinismo si su función de transición es $\delta : Q \times \Sigma \to 2^Q$. O sea, si dado un estado y un símbolo de entrada, la transición puede darse a más de un estado.

Observen que ahora la salida de la función de transición es un **conjunto** de estados (que puede ser vacío) y no un estado particular.

¿Cómo *pensamos* el no-determinismo?

Como ya mencionamos, la diferencia entre un AFND y un AFD es la definición de su **función de transición**. Las transiciones de los AFND las *pensaremos* de la misma manera que en un AFD, solamente que en algún momento tendremos más de un camino para tomar.

Por ejemplo, veamos el siguiente ejemplo sobre $\Sigma = \{\#, 0, 1\}$.



El AFND de la imagen cuenta con su función de transición definida como:

- $\delta(q_0, 0) = \{q_0\}$
- $\delta(q_0, 1) = \{q_0\}$
- $\delta(q_0, \#) = \{q_0, q_1\}$
- $\delta(q_1, 0) = \{q_2\}$
- $\delta(q_1, 1) = \{q_2\}$

Cuando el AFND lee un #, podríamos pensar como si ahora hubiera dos *caminos* de ejecución, uno que se mantiene en q_0 y otro que pasa a q_1. Por tanto, cuando venga un 0 o un 1, habrá uno de esos *caminos* que quedará nuevamente en el estado q_0 y el otro irá al estado q_2.

Pero...¿cuándo una tira es aceptada por un AFND?

En el teórico se vió que se puede modificar la función de transición para que, en lugar de símbolos, admita tiras. A esa función le llamamos $\hat{\delta}$ y está definida como $\hat{\delta} : Q \times \Sigma^* \to 2^Q$.

Por tanto, diremos que una tira w es **reconocida** por un AFND si

$$\exists q_i \in F : q_i \in \hat{\delta}(q_o, w)$$

Así que, en lenguaje natural, un AFND acepta una tira cuando:

1. Se lee **toda** la tira
2. Uno de esos *caminos* queda en un **estado final**.

Nota: hay que prestar especial atención al diseñar el AFND. Para afirmar que una tira **es reconocida** basta con que uno de esos caminos llegue a un estado final. Por el contrario, una tira **no es reconocida** si es rechazada en absolutamente todos los posibles caminos de ejecución.

El problema

Ahora sí, abordemos un ejemplo. En este caso veremos cómo modelar el lenguaje sobre $\Sigma = \{a, b, c\}$ definido como

$$L_3 = L((a|b|c)^*b(a|b|c)(a|b|c))$$

Así que L_3 está compuesto por todas las tiras de Σ^* cuyo tercer símbolo desde la derecha es una b.

Diseñando una posible solución

Antes de empezar, tómense un momento para pensar cómo sería un posible AFD que reconozca a L_3...

...sí, complejo.

Al pensar un AFD para este lenguaje nos vemos en la dificultad de que cada vez que leemos una b no tenemos la certeza de que luego solamente vengan 2 símbolos y se acabe la tira. En estos casos es que el AFND nos permite visualizar una solución que contemple ambos casos, haciendo uso de su **no-determinismo**.

Diseñemos entonces algún AFND que reconozca el lenguaje L_3. Vamos a necesitar estados para:

1. Leer cualquier a, b o c
2. Luego, leer la b obligatoria, que es el tercer símbolo contando desde la derecha
3. Leer 2 símbolos más de Σ. Este paso es el que hace que la b leída se convierta en el 3er símbolo desde la derecha.

Es sencillo notar que el estado final será el correspondiente al ítem 3.

Implementando la solución

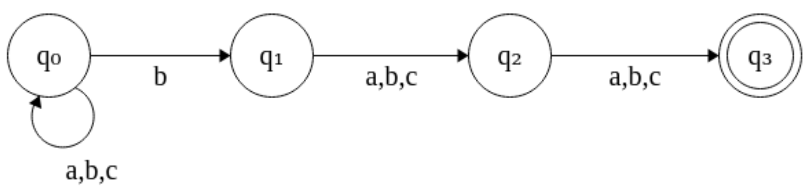
Entonces, la función de transición nos queda:

δ	a	b	c
q_0	$\{q_0\}$	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\{q_2\}$	$\{q_2\}$	$\{q_2\}$
q_2	$\{q_3\}$	$\{q_3\}$	$\{q_3\}$
q_3	$\{\}$	$\{\}$	$\{\}$

Y ya tenemos los elementos de nuestro AFND:

- $Q = \{q_0, q_1, q_2, q_3\}$ es el conjunto de estados.
- $\Sigma = \{a, b, c\}$ es el alfabeto.
- δ es la función de transición que acabamos de definir.
- q_0 es el estado inicial.
- $F = \{q_3\}$ es el conjunto de estados finales.

Gráficamente, el AFND resulta:



Como habrán notado, las transiciones para q_3 no están definidas. Esto se debe a que no incluimos el estado pozo. Cuando la función de transición δ tiene casos sin definir, decimos que δ **no es total**.

¿Cómo sería un posible AFD para este problema?

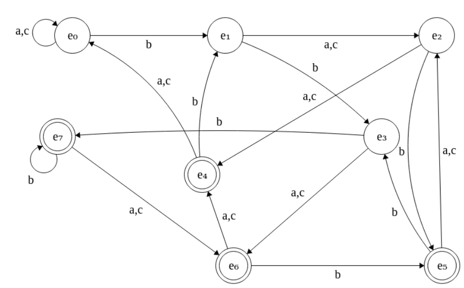
Pensar el AFND fue bastante directo ¿no? Sin embargo, como comentamos al inicio, pensar un AFD para este problema resulta particularmente rebuscado. Por curiosidad y vía de comparación, vamos a usar el **algoritmo de pasaje de AFND a AFD** para obtener un AFD equivalente.

Tal como se explica en la clase teórica, partiendo desde q_0 veremos a qué conjunto de estados se transicionará con cada símbolo. Los nuevos estados finales serán los que contengan estados finales del AFND (en nuestro caso, los que contengan a q_3).

δ	a	b	c
$[q_0]$	$[q_0]$	$[q_0q_1]$	$[q_0]$
$[q_0q_1]$	$[q_0q_2]$	$[q_0q_1q_2]$	$[q_0q_2]$
$[q_0q_2]$	$[q_0q_3]$	$[q_0q_1q_3]$	$[q_0q_3]$
$[q_0q_1q_2]$	$[q_0q_2q_3]$	$[q_0q_1q_2q_3]$	$[q_0q_2q_3]$
$[q_0q_3]$	$[q_0]$	$[q_0q_1]$	$[q_0]$
$[q_0q_1q_3]$	$[q_0q_2]$	$[q_0q_1q_2]$	$[q_0q_2]$
$[q_0q_2q_3]$	$[q_0q_3]$	$[q_0q_1q_3]$	$[q_0q_3]$
$[q_0q_1q_2q_3]$	$[q_0q_2q_3]$	$[q_0q_1q_2q_3]$	$[q_0q_2q_3]$

Entonces, renombrando a los estados, obtenemos un AFD equivalente:

δ	a	b	c
e_0	e_0	e_1	e_0
e_1	e_2	e_3	e_2
e_2	e_4	e_5	e_4
e_3	e_6	e_7	e_6
e_4	e_0	e_1	e_0
e_5	e_2	e_3	e_2
e_6	e_4	e_5	e_4
e_7	e_6	e_7	e_6



¿Se les hubiera ocurrido ese AFD?...

...bastante rebuscado ¿verdad?

Como pueden observar, en algunos casos resulta más simple e intuitivo pensar un AFND que un AFD. Así que dependerá de cada lenguaje cómo es que conviene razonar el modelo, si de manera determinista o no-determinista.