Ejemplos de Máquinas con Salida

Teoría de Lenguajes

2020

1 Introducción

Este documento tiene como objetivo mostrar algunos ejercicios de máquinas con salida.

Dichas máquinas con salida, se pueden ver como "traductores" de una entrada x en una salida y donde cada elemento de dichas tiras tiene un alfabeto que puede o no coincidir. Cabe destacar que las entradas y salidas deberían ser iguales en largo y en este mismo documento presentaremos la extensión con transiciones épsilon que permite tener largos de entrada y salida distintos.

Es importante destacar que, a diferencia de los otros modelos (de máquinas) vistos previamente, las máquinas con salidas carecen del concepto de reconocimiento, por lo tanto, **no tiene estados finales**.

En el teórico se puede apreciar que hay **dos modelos equivalentes**, Mealy y Moore (**ambos deterministas**), donde se imprime siempre utilizando un único símbolo de salida.

Por cuestiones de practicidad, utilizaremos las máquinas con salida de Mealy.

Es recomendable mirar la clase 12 del curso en OpenFing para entender con facilidad los ejemplos (https://open.fing.edu.uy/courses/tl/12)

2 Ejemplo 1 - Traducción de entrada a salida

Construir un autómata con salida M: (Q,$\Sigma, \Lambda, \delta, \lambda, q_0$):

- Imprima P por cada símbolo de entrada mientras la cantidad de a's es par.
- Imprima I por cada símbolo de entrada mientras la cantidad de a's sea impar.

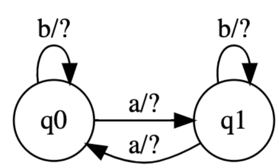
Para ambos casos, se contabiliza el símbolo que se está procesando para determinar la salida y estos son algunos ejemplos de entrada y salida:

Entrada	Salida
abab	IIPP
bbbb	PPPP
baab	PIPP

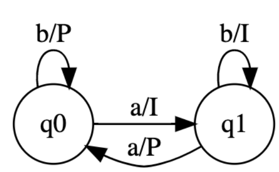
2.0.1 Diseñando una posible solución

Este primer ejemplo nos pide imprimir la salida en función del estado parcial que se tiene mientras se está procesando la entrada, es decir, nos pide procesar la paridad de la cantidad de a's en la tira de entrada para, en función de eso, imprimir la salida.

Es fácil crear una máquina que guarde el estado de la cantidad de a's. Olvidándonos de la salida durante unos instantes, la solución parcial tiene la siguiente forma:



Ahora solo resta agregar las salidas, para ello, vamos a tomar en cuenta los ejemplos dados y resultando en la siguiente solución:



Por último, restaría evaluar **los casos bordes** y **los ejemplos** para asegurarnos que nuestra máquina funciona bien.

3 Extensión de Máquinas con Salida

Existe una extensión para las máquinas con salida en las cuales se permite utilizar la transición ϵ para moverse entre estados y también para emitir ϵ.

En el caso de utlizarla, las máquinas deberían continuar siendo deterministas, es decir, no debería suceder que se tenga para una misma entrada, distintas salidas en función de si dicha transición épsilon fue utilizada o no. O dicho en otras palabras, si existe una transición épsilon en un estado Qx entonces no debería existir otra transición en dicho estado para que siga siendo determinista la máquina.

3.1 Ejemplo 2 - Salidas potencialmente más pequeñas que la entrada

Construir un autómata con salida $(Q, \Sigma, \Lambda, \delta, \lambda, q_0)$:

- Por cada secuencia *aba* imprima I
- Por cada secuencia *bb* imprima R

Considere: $\Lambda = \{I, R\}; \Sigma = \{a, b\}; \delta : Q \times (\Sigma \cup \{\epsilon\}) \rightarrow (\Lambda \cup \{\epsilon\})$ y los siguientes ejemplos de entrada y salida:

Entrada	Salida
aababaabbba	IIRR
bbbbaaab	RRR
baaabbab	R
baaaab	ϵ
babababab	III

3.1.1 Diseñando una posible solución

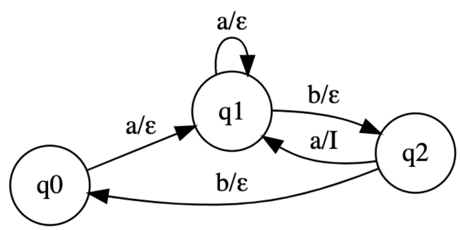
La letra deja en claro que hay "traducir" dos casos, que al no estar muy relacionados entre si, permiten construir la máquina en forma incremental.

Adicionalmente, si miramos los ejemplos de entradas y salidas, vemos que en todos los casos la salida tiene largo menor a la entrada;
Si observamos también que:
$\delta : Q \times (\Sigma \cup \{\epsilon\}) \rightarrow (\Lambda \cup \{\epsilon\})$
Vemos que un símbolo de salida válido puede ser ϵ.

Enfoquémonos en el primer caso, como reconocemos las tiras que *por cada secuencia aba* $\rightarrow I$

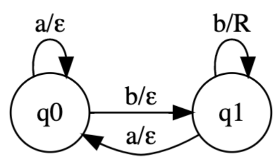
Si miramos el ejemplo de salida: *babababab* $\rightarrow$ *III*, vemos que el final de una secuencia *aba* puede tomarse como principio de otra secuencia *aba*, o lo que es lo mismo: ababa $\rightarrow$ II

Por lo que, una posible solución para satisfacer el primer caso, es el siguiente autómata:

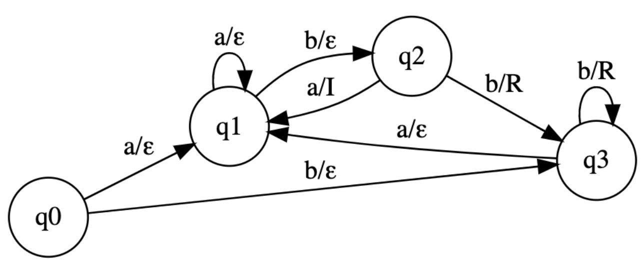


Ahora miremos el segundo caso, *por cada secuencia bb* $\rightarrow$ *R*

Dicho caso es más simple ya que implica simplemente imprimir por cada dos b's consecutivas una R. Notar que se cumple lo observado anteriormente donde $bbbbaaab \rightarrow RRR$, es decir, $bbb \rightarrow RR$.



Para pensar la solución completa, hace falta consolidar los dos autómatas en uno de forma que se satisfagan ambos casos. Una forma de hacer esto es fusionando ambos autómatas en uno solo de forma de reutilizar estados. Por ejemplo el q0 en la parte del reconocimiento de la secuencia bb luego es el q1 en el fusionado (idem con q1 y q3)



Por último, restaría evaluar **los casos bordes** y **los ejemplos** para asegurarnos que nuestra máquina funciona bien.

3.2 Ejemplo 3 - Salidas potencialmente más grandes que la entrada

Construya una máquina con salida que dada una entrada x sobre el alfabeto {a, b} emita una salida y sobre el alfabeto {c, d} que verifique:

- $cant_c(y) = \lfloor cant_a(x)/2 \rfloor$
- $cant_d(y) = 2 \times cant_b(x)$

Ejemplos de entrada y salida:

Entrada	Salida
aab	cdd
abaa	ddc
bbb	dddddd

Observación: para una misma entrada existen eventualmente muchas salidas. Su máquina secuencial debe dar sólo una de ellas.

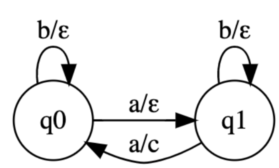
3.2.1 Diseñando una posible solución

Comencemos por analizar por separado las condiciones que nos piden que cumpla nuestra máquina:

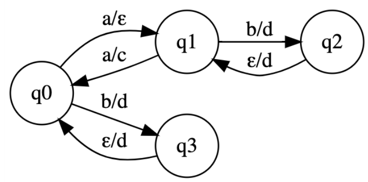
- $cant_c(y) = \lfloor cant_a(x)/2 \rfloor$: esto significa que, por cada 2 a's que se consuman de la entrada, se imprimirá una c. Al especificar utlizando la parte entera, nos obliga a imprimir dicha 'c' cuando se consuman las a's que están en lugares pares ya que, por ejemplo, si se consume únicamente una a entonces no tendremos c's.

- $cant_d(y) = 2 \times cant_b(x)$: esto significa simplente que por cada b que se consuma en la entrada se imprimirán 2 d's.

Comencemos solucionando la primera condición, es fácil ver que la máquina va a tener la siguiente forma:



Ahora bien, solo falta hacer que se cumpla la segunda condición, para ello simplemente deberíamos modificar que sucede cuando se consume una b en la entrada:



Por último, restaría evaluar **los casos bordes** y **los ejemplos** para asegurarnos que nuestra máquina funciona bien.