Ejercicio completo de minimización de autómatas y cálculo de clases de equivalencia de un lenguaje partiendo de un AFND-ϵ

Teoría de Lenguajes

2020

Introducción

Este documento tiene como objetivo mostrar un ejercicio completo de minimización de autómatas, partiendo de un AFND-ϵ pasando por la aplicación en cascada de los 3 algoritmos para llegar a un AFD Mínimo.
Habitualmente interesa conocer cuál es el lenguaje reconocido por el autómata y dar una definición de éste mediante una expresión regular. Entonces obteniendo el autómata mínimo, si logramos encontrar expresiones regulares asociadas a los estados finales, la unión de esas expresiones regulares nos dará una definicón del lenguaje. El procedimiento para el cálculo de las expresiones regulares asociadas a cada una de las clases de equivalencia de los estados del autómata que se presenta en otro documento *álculo de expresiones de las clases de* R_M en esta misma sección del *EVA*.

El no-determinismo con transiciones ϵ

Como ya se ha mencionado en las clases teóricas, en general siempre vamos a necesitar llegar a un **AFD** para demostrar algunas propiedades. Pero yendo a aspectos más prácticos, en los lenguajes de programación los analizadores lexicográficos suelen implementarse con simples AFDs, los cuales se unen utilizando transiciones ϵ. Luego, se implementan los algoritmos en cascada a efectos de obtener el **AFD** asociado a cada una de las expresiones regulares que especifican los *tokens* del lenguaje. Por eso es importante entender el uso y aplicación de estos autómatas y sus algoritmos.

El problema

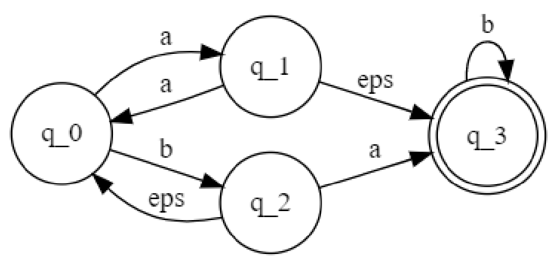
Sea L el lenguaje reconocido por el siguiente autómata finito M:

$$\text{M} = (\ \text{Q}, \Sigma, \delta, q_0, F\ ) \text{ donde:}$$

- $Q = \{q_0, q_1, q_2, q_3\}$ es el conjunto de estados
- $\Sigma = \{a, b\}$ es el alfabeto
- q_0 es el estado inicial
- $F = \{q_3\}$ es el conjunto de estados finales
- δ es la función de transición dada por:

δ	a	b	ϵ
q_0	$\{q_1\}$	$\{q_2\}$	$\{\}$
q_1	$\{q_0\}$	$\{\}$	$\{q_3\}$
q_2	$\{q_3\}$	$\{\}$	$\{q_0\}$
q_3	$\{\}$	$\{q_3\}$	$\{\}$

Gráficamente, el AFND-ϵ resulta:



Se puede ver que en algunos casos las transiciones de un estado con un símbolo da como resultado el conjunto vacío. Por ahora es algo que no nos molesta, pero que más adelante vamos a volver sobre ese tema cuando se hable de minimización y clases de equivalencia.

Pasaje de AFND- ϵ a AFND

Lo primero que vamos a hacer es calcular las ϵ - clausura de cada uno de los estados:

ϵ-clausura (q_0) = $\{q_0\}$
ϵ-clausura (q_1) = $\{q_1, q_3\}$
ϵ-clausura (q_2) = $\{q_0, q_2\}$
ϵ-clausura (q_3) = $\{q_3\}$

Recordemos que la ϵ - clausura de un estado es el conjunto de estados a los cuales se llega por un camino etiquetado **exclusivamente** por arcos ϵ.
A continuación se aplica el algoritmo que elimina las transiciones ϵ obteniéndose M_1 /

$$\mathrm{M}_1 = (Q_1, \Sigma, \delta_1, q_0, F_1\ ) \text{ donde:}$$

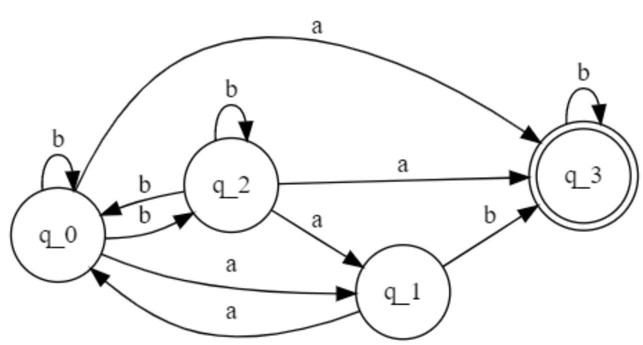
- $Q_1 = Q = \{q_0, q_1, q_2, q_3\}$ es el conjunto de estados (el mismo)
- $\Sigma = \{a, b\}$ es el alfabeto (el mismo)
- q_0 es el estado inicial (el mismo)
  y acá hay que ver que pasa con la ϵ-clausura (q_0), ya que si q_0 $\notin F, pero$ ϵ-clausura (q_0) $\cap F \neq \emptyset$ entonces hay que agregar q_0 al conjunto de estados finales del nuevo autómata, es decir, a F_1.
  En nuestro ejemplo eso **no** sucede, con lo cual:
- $F_1 = F = \{q_3\}$ es el conjunto de estados finales (el mismo)
- δ_1 es la función de transición dada por:

δ_1	a	b
q_0	$\{q_1, q_3\}$	$\{q_0, q_2\}$
q_1	$\{q_0\}$	$\{q_3\}$
q_2	$\{q_1, q_3\}$	$\{q_0, q_2\}$
q_3	$\{\}$	$\{q_3\}$

Al cual se llega aplicando la fórmula:

$$\delta_1(\text{q,a}) = \epsilon\text{-clausura}(\tilde{\delta}(\epsilon\text{-clausura (q),a}))$$

Quedando finalmente el siguiente diagrama de estados:

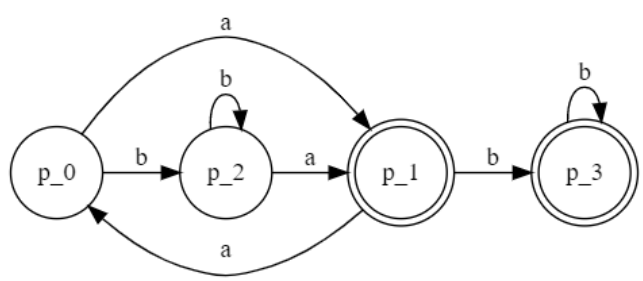


Pasaje de AFND a AFD

El siguiente paso consiste en obtener el AFD equivalente al AFND, aplicando el algoritmo visto en el curso, el cual consiste en ir ägrupando.estados con un comportamiento similar - los superestados que mencionábamos en la clase - en función de las transiciones y que es explicado en el documento *Modelado con autómatas AFND* (en esta misma sección del EVA).
Se obtiene entonces un nuevo autómata M_2 cuya función de transición está dada en la siguiente tabla:

	δ_2	a	b
$[p_0]$	$[q_0]$	$[q_1q_3]$	$[q_0q_2]$
$[p_1]$	$[q_1q_3]$	$[q_0]$	$[q_3]$
$[p_2]$	$[q_0q_2]$	$[q_1q_3]$	$[q_0q_2]$
$[p_3]$	$[q_3]$	$-$	$[q_3]$

Se agregó una columna más a la tabla que renombra los estados para mayor claridad.
Los nuevos estados finales serán los que contengan estados finales del AFND, que en este ejemplo son los que contengan a q_3, es decir p_1 y p_3.



Una vez que tenemos un AFD, lo que nos resta hacer ahora es obtener el Autómata Determinista Mínimo (AFDM), para lo cual se va a utilizar el algoritmo visto en clase que trata de ir realizando una partición de los estados del autómata tratando de que en cada una de las clases vayan quedando estados indistinguibles unos de otros.

Pasaje de AFD a AFDM

Se tiene como entrada el autómata M_2 obtenido en la sección anterior. Pero antes vamos realizar la siguiente consideración. Muchas veces puede llegar a interesar conocer las clases de equivalencia definidas en la relación R_L para un lenguaje L cualquiera y la tarea de encontrar justamente estas clases no suele ser algo sencillo de obtener al menos a simple vista.

Como se recordará, el Teorema de Myhill-Noerode hablaba que el autómata mínimo es aquél que se construye de manera tal de que cada estado es una clase de equivalencia dada a partir de la relación R_L.

Por otro lado, definimos una relación R_M, como una relación asociada a los autómatas finitos deterministas, que realiza una partición de las tiras de Σ^* en clases que son representadas por los estados del autómata M. Entonces, cuando ese autómata sea el mínimo, las clases de R_M coinciden con las clases de R_L.

Como última conisderación, como R_L es una relación que actúa sobre las tiras de Σ^*, es necesario que el autómata mínimo tenga una función de transición que sea **total**; es decir, considere todas las posibles entradas de elementos de Σ^*.
Por lo tanto, antes de minimizar el AFD con el algoritmo visto en el curso, en caso de que la función de transición no sea total, la extendemos definiendo las transiciones que falten y las mandamos hacia un *estado pozo*.

En nuestro ejemplo, puede verse que justamente pasa eso.
Procederemos a completar el AFD M_2 obtenido en la sección anterior, donde es necesario definir un estado pozo - que denominaremos p_p-, a donde se llegará desde el estado p_3 con el símbolo **a**.

El autómata a minimizar queda entonces:

$$M_3 = (\ \mathrm{Q}_3, \Sigma, \delta_3, p_0, F_3\ ) \text{ donde:}$$

- $Q_3 = \{p_0, p_1, p_2, p_3, p_p\}$ es el conjunto de estados
- $\Sigma = \{a, b\}$ es el alfabeto
- p_0 es el estado inicial
- $F_3 = \{p_1, p_3\}$ es el conjunto de estados finales
- δ_3 dado por:

δ_3	a	b
p_0	p_1	p_2
p_1	p_0	p_3
p_2	p_1	p_2
p_3	p_p	p_3
p_p	p_p	p_p

Partimos entonces de la partición inicial Π_0 que separa los estados finales de los no finales. En nuestro ejemplo:

$\Pi_0 = \{\{\mathrm{p}_0, p_2, p_p\}, \{p_1, p_3\}\}$

Para obtener la Π_1 se ejecuta la función de transición para cada par $\langle estado, símbolo \rangle$ de cada conjunto de la partición. Hacemos entonces:

$$\delta_3(\mathrm{p}_0, a) = \mathrm{p}_1 \qquad \delta_3(\mathrm{p}_2, a) = \mathrm{p}_1$$
$$\delta_3(\mathrm{p}_0, b) = \mathrm{p}_2 \qquad \delta_3(\mathrm{p}_2, b) = \mathrm{p}_2$$

Tanto para el caso del ímbolo **a**, como el símbolo **b** ambos estados (p_o y p_2) van al mismo estado (p_1 en el caso del **a** y p_2 en el caso del **b**), con lo cual ambos pertenecen a la misma clase de la nueva partición que estamos considerando (en este caso Π_0).
Hacemos lo mismo con p_p ya que pertenece a la misma clase que p_0 y p_2

$$\delta_3(\mathrm{p}_0, a) = \mathrm{p}_1 \qquad \delta_3(\mathrm{p}_p, a) = \mathrm{p}_p$$

y en este caso, como ya p_1 y p_p pertenecen a conjuntos distintos dentro de la partición Π_0, se van a separar en conjuntos disjuntos en la nueva a construir, que será Π_1.
Notar que **no** es necesario ver que sucede con δ_3 de ese par de estados con el símbolo **b** porque con el símbolo **a** ya se separan.
Se hace lo mismo con p_1 y p_3:

$$\delta_3(\mathrm{p}_1, a) = p_0 \qquad \delta_3(\mathrm{p}_3, a) = \mathrm{p}_p$$
$$\delta_3(\mathrm{p}_1, b) = p_3 \qquad \delta_2(\mathrm{p}_3, b) = \mathrm{p}_3$$

Se observa entonces que p_0 y p_3 son indistinguibles y por lo tanto no se pueden separar.
Con esto, la Π_1 queda con los siguientes subconjuntos:

$\Pi_1 = \{\{\mathrm{p}_0, p_2\}, \{p_p\}, \{p_1, p_3\}\}$

Es fácil de ver que como p_0 y p_2 siempre van al mismo estado con **a** en un caso y con **b** en otro, **nunca** se van a separar. Con lo cual volvemos a realizar lo mismo pero con p_1 y p_3.

Se observa ahora:

$$\delta_3(\mathrm{p}_1, a) = p_0 \qquad \delta_3(\mathrm{p}_3, a) = \mathrm{p}_p$$
$$\delta_3(\mathrm{p}_1, b) = p_3 \qquad \delta_2(\mathrm{p}_3, b) = \mathrm{p}_3$$

donde p_0 y p_p pertenecen a distinto subconjunto de la Π_1, con lo cual en esta nueva iteración del algoritmo donde se está armando la Π_2,estos estados se separan.

Una vez más, se vuelve a ejecutar el algoritmo armando la Π_3.

El algoritmo se detiene cuando se observa que existen dos iteraciones seguidas que tienen el mismo conjunto de clases.

El proceso de construcción de las distintas particiones se puede ver en el siguiente esquema:

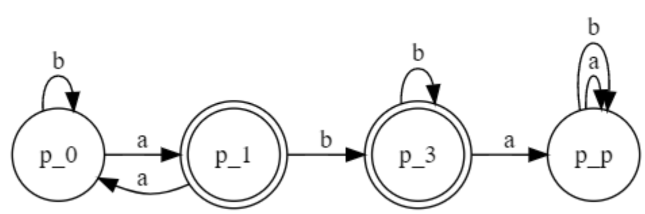
Π_0: $\{\mathrm{p}_0, p_2, p_p\}$	$\{p_1, p_3\}$		
Π_1: $\{\mathrm{p}_0, p_2\}$	$\{p_p\}$	$\{p_1, p_3\}$	
Π_2: $\{\mathrm{p}_0, p_2\}$	$\{p_p\}$	$\{p_1\}$	$\{p_3\}$
Π_3: $\{\mathrm{p}_0, p_2\}$	$\{p_p\}$	$\{p_1\}$	$\{p_3\}$

Es decir, que en nuestro ejemplo, el algoritmo de refinamiento de particiones se detiene al notar que Π_2 y Π_3 coinciden.
Finalmente se arma el AFDM M_{Min} /

$$M_{Min} = (Q_{Min}, \Sigma, \delta_{Min}, p_0, F_{Min}) \text{ donde:}$$

- $Q_{Min} = \{p_0, p_1, p_3, p_p\}$ es el conjunto de estados, tomando un representante de cada clase.
- $\Sigma = \{a, b\}$ es el alfabeto
- p_0 es el estado inicial (representante de la clase que contiene a p_0
- $F_{Min} = \{p_1, p_3\}$ es el conjunto de estados finales
- δ_{Min} en base a la δ_3 del AFD para aquellos estados que como decíamos elegimos como representantes de cada una de las clases

El diagrama entonces del Auómata Finito Determinista Mínimo es:



Obtención de las clases de equivalencia definidas por R_M

El siguiente paso es tratar de obtener las clases de equivalencia asociadas a los estados. No se explicará en detalle el procedimiento debido a que eso es presentado en el material (*Cálculo de expresiones de las clases de* R_M) en esta sección del EVA.

A continuación, el sistema de ecuaciones a resolver a partir del autómata obtenido en la sección anterior.

$X_0 = X_0b|X_1a|\ \epsilon$
$X_1 = X_0a$
$X_3 = X_1b|X_3b$
$X_p = X_3a|X_p(a|b)$

$X_0 = X_0b|X_0aa|\ \epsilon =X_0(b|aa)|\ \epsilon = (b|aa)^*$

$\boxed{X_0 = (b|aa)^*}$

por lo tanto

$\boxed{X_1 = (b|aa)^*a}$

$X_3 = X_3b|(b|aa)^*ab = (b|aa)^*abb^*$

$\boxed{X_3 = (b|aa)^*abb^*}$

quedando por definir la expresión regular asosciada al estado pozo

$X_p = (b|aa)^*abb^*a|X_p(a|b)$

$\boxed{X_p = (b|aa)^*abb^*a|(a|b)^*}$

La solución será la ER asociada a $X_1 \mid X_3$ por ser estados finales p_1 y p_3.

$$L(M) = X_1|X_3 = (b|aa)^*a \quad | \quad (b|aa)^*abb^* \;=\; \boxed{(b|aa)^*ab^*}$$