Autómatas y Lenguajes Formales 2019-II
Facultad de Ciencias UNAM*
Nota de Clase 6: El autómata mínimo

Favio E. Miranda Perea   A. Liliana Reyes Cabello   Lourdes González Huesca

29 de marzo de 2019

En esta nota discutimos el problema de reducir el tamaño de autómatas deterministas en el sentido de usar el mínimo número de estados posibles, lo cual permite obtener máquinas más eficientes.

1. Eliminación de Estados Inaccesibles

Sea $M = \langle Q, \Sigma, \delta, q_0, F\rangle$ un AFD. Decimos que un estado $q \in Q$ es accesible si y sólo si existe $w \in \Sigma^\star$ tal que $\delta^\star(q_0, w) = q$. Es decir, q es accesible si y sólo si el procesamiento de alguna cadena termina en el estado q.

El conjunto de estados accesibles de un autómata M se denota $Acc(M)$. Si un estado no es accesible decimos que es inaccesible.

Es claro que el conjunto $Acc(M)$ puede construirse de manera algorítmica, por ejemplo como sigue:

```
A_N := {q_0}  % estados accesibles
A_V := ∅      % estados verificados
while  A_N ≠ A_V do
  A_V := A_N
  A_N := A_N ∪ {q ∈ Q | δ(p,a) = q, a ∈ Σ, p ∈ A_N}
return A_N
```

Otra manera de obtener $Acc(M)$ es directamente de la digráfica de estados, donde un estado es inaccesible si y sólo si el nodo correspondiente tiene ingrado cero (el estado inicial se considera con ingrado positivo)

Los estados inaccesibles en un autómata son inútiles y pueden ser eliminados sin afectar el lenguaje de aceptación como vemos a continuación:

Proposición 1. *Dado $M = \langle Q, \Sigma, \delta, q_0, F\rangle$ un AF, existe un AF $M' = \langle Q', \Sigma, \delta', q_0, F'\rangle$ equivalente a M que contiene únicamente a los estados accesibles de M, es decir, $Q' = Acc(M)$ y por lo tanto no contiene estados inaccesibles. Para lo anterior basta definir M' como sigue:*

*Material elaborado en el marco del proyecto PAPIME PE102117

- $Q' = Acc(M)$
- $\delta' = \delta\restriction_{Q'}$
- $F' = F \cap Q'$

Demostración. Ejercicio $\square$

Debido a este resultado de ahora en adelante podemos suponer que un autómata no tiene estados inaccesibles.

2. Equivalencia de estados y el autómata cociente

Puede ser el caso que unas partes de un autómata sean redundantes, es decir que las cadenas que son aceptadas por una parte del autómata también pueden ser procesadas y aceptadas por otra parte. Veamos cómo abstraer y generalizar partes de los autómatas para minimizar el número de estados sin afectar al lenguaje de aceptación.

Definición 1. *Decimos que dos estados $q, q' \in Q$ de un AFD son equivalentes $q \equiv q'$ si y sólo si:*

$$\forall w \in \Sigma^\star \ (\delta^\star(q, w) \in F \Leftrightarrow \delta^\star(q', w) \in F)$$

Es decir, si $\delta^\star(q, w), \delta^\star(q', w)$ son ambos finales o ambos no finales.

La relación $\equiv$ entre estados es una relación de equivalencia, es decir cumple lo siguiente:

- Reflexividad: $q \equiv q$.
- Simetria: si $q \equiv q'$ entonces $q' \equiv q$.
- Transitividad: si $q \equiv q'$ y $q' \equiv q''$ entonces $q \equiv q''$.

Adicionalmente la función de transición δ es compatible con $\equiv$, en el siguiente sentido:

$$\text{Si } q \equiv q' \text{ entonces } \forall a \in \Sigma \ \big(\delta(q, a) \equiv \delta(q', a)\big)$$

La relación de equivalencia $\equiv$ genera una **partición** del conjunto de estados dada por las clases de equivalencia de cada estado definidas como:

$$[q] := \{p \in Q \mid q \equiv p\}$$

Es decir, los conjuntos de estados $[q]$ cumplen lo siguiente:

- $\forall q \in Q \ ([q] \neq \varnothing\ )$.
- $\forall p, q \in Q \ ([q] = [p]$ ó $[q] \cap [p] = \varnothing)$.
- $\bigcup_{q \in Q} [q] = Q$.

Al agrupar por clases de equivalencia a los estados de un autómata, se puede calcular otro autómata llamado **autómata cociente** que, como veremos, tiene un número mínimo de estados.

Definición 2. *Dado un AFD* $M = \langle Q, \Sigma, \delta, q_0, F\rangle$ *existe el autómata cociente* $M/_{\equiv}$ *también conocido como* M^{min} *que es la minimización de* M *y se define como* $M^{min} = \langle Q_m, \Sigma, \delta_m, [q_0], F_m\rangle$ *donde:*

- $Q_m =_{def} Q/_{\equiv} =_{def} \{[q] \mid q \in Q\}$ *los estados son las clases de equivalencia*
- *la clase* $[q_0]$ *es el estado inicial.*
- $F_m := \{[q] \mid q \in F\}$
- $\delta_m : Q_m \times \Sigma \to Q_m$ *se define como* $\delta_m([q], a) = [\delta(q, a)]$

Es fácil ver que la función de transición δ_m está bien definida. Más aún, la equivalencia entre M y M^{min} se sigue de la siguiente propiedad:

Lema 1. *Sean* $M = \langle Q, \Sigma, \delta, q_0, F\rangle$ *un AFD y* M^{min} *su autómata cociente. Para cualesquiera* $q \in Q,\ w \in \Sigma^\star$ *se cumple*

$$\delta^\star_m([q], w) = [\delta^\star(q, w)]$$

Demostración. Inducción sobre w. □

Veamos ahora la minimalidad del autómata cociente.

Proposición 2. *Dado un AFD* $M = \langle Q, \Sigma, \delta, q_0, F\rangle$*, el autómata cociente*

$$M^{min} =_{def} \langle Q_m, \Sigma, \delta_m, [q_0], F_m\rangle$$

es el autómata mínimo equivalente a M*. Es decir, se tiene* $L(M) = L(M^{min})$ *y no existe un autómata equivalente a* M *con menos estados que* M^{min}*.*

La proposición se sigue del siguiente lema:

Lema 2. *Sean* $M = \langle Q, \Sigma, \delta, q_0, F\rangle$ *un AFD y* $M^{min} =_{def} \langle Q_m, \Sigma, \delta_m, [q_0], F_m\rangle$ *su autómata cociente. Si* $N = \langle R, \Sigma, \delta_N, r_0, F_N\rangle$ *es un AFD tal que* $|R| < |Q_m|$ *entonces* $L(N) \neq L(M)$*. Es decir, si* N *tiene menos estados que* M^{min} *entonces* M *y* N *no son equivalentes.*

Demostración. Sea $Q_m = \{p_0, \ldots, p_k\}$ con $p_0 = [q_0]$ el estado inicial y sea N un AFD con $|R| < |Q_m|$. Como no hay estados inaccesibles en M^{min} deben existir cadenas distintas $w_0, \ldots, w_k$ tales que $\delta^\star_m(p_0, w_i) = p_i,\ i = 0, \ldots, k$. Por otra parte, como N tiene menos estados que M^{min}, entonces para al menos dos de las cadenas anteriores, digamos w_l, w_j se tiene $\delta^\star_N(r_0, w_l) = \delta^\star_N(r_0, w_j)$. Ahora bien, como los estados p_l, p_j no son equivalentes en M^{min} entonces existe una cadena $x \in \Sigma^\star$ tal que $\delta^\star_m(p_l, x)$ no es final y $\delta^\star_m(p_j, x)$ sí o viceversa. En cualquier caso se tiene $\delta^\star_m(p_0, w_l x) = \delta^\star_m(p_l, x)$ y $\delta^\star_m(p_0, w_j x) = \delta^\star_m(p_j, x)$ y por lo tanto $w_l x$ es aceptada por M^{min} y $w_j x$ no o viceversa. Pero por otro lado se tiene que

$$\delta^\star_N(r_0, w_l x) = \delta^\star_N(\delta^\star_N(r_0, w_l), x) = \delta^\star_N(\delta^\star_N(r_0, w_j), x) = \delta^\star_N(r_0, w_j x)$$

Esto implica que N acepta o rechaza ambas cadenas y por lo tanto N y M^{min} no aceptan el mismo lenguaje, es decir, no son equivalentes. De donde tampoco M y N pueden ser equivalentes. □

3. Las relaciones de k-equivalencia

La construcción del autómata mínimo requiere de calcular las clases de equivalencia de una relación cuya definición involucra una propiedad de todas las cadenas en $\Sigma^\star$. Esto hace que en la práctica sea muy complicado encontrar las clases de equivalencia. Afortunadamente podemos resolver este problema sustituyendo la relación de equivalencia original por sus restricciones a cadenas finitas hasta cierta longitud.

Definición 3. *Sea M un AFD. Para cualquier $k \in \mathbb{N}$ definimos la relación de k-equivalencia de estados como sigue:*

$$q \equiv_k q' \Leftrightarrow_{def} \forall w \in \Sigma^\star \big(|w| \leq k \to (\delta^\star(q,w) \in F \leftrightarrow \delta^\star(q',w) \in F)\big)$$

Es decir, q es k-equivalente a q' si y sólo si para cualquier cadena w de longitud menor o igual que k, los estados $\delta^\star(q,w)$ y $\delta^\star(q',w)$ son ambos finales o ambos no finales.
Así $\equiv_k$ es una relación de equivalencia cuyas clases se denotan con $[q]_k$, es decir

$$[q]_k = \{p \in Q \mid q \equiv_k p\}$$

Veamos ahora algunas propiedades relevantes de esta familia de relaciones de equivalencia:

Proposición 3. *Las relaciones de k-equivalencia cumplen las siguientes propiedades:*

P1 $q \equiv q'$ *si y sólo si* $\forall k \in \mathbb{N}(q \equiv_k q')$.

P2 $q \equiv_0 q'$ *si y sólo si* $q, q' \in F$ *ó* $q, q' \in Q - F$.

P3 $[q]_0 = F$ *si y sólo si* $q \in F$.

P4 Si $q \equiv_k q'$ *entonces* $q \equiv_{k-1} q'$.

P5 $[q]_k \subseteq [q]_{k-1}$

P6 Si $q \equiv_k q'$ *entonces* $\forall \sigma \in \Sigma(\delta(q,\sigma) \equiv_{k-1} \delta(q',\sigma))$

P7 $q \equiv_k q'$ *si y sólo si* $q \equiv_{k-1} q'$ *y* $\forall \sigma \in \Sigma(\delta(q,\sigma) \equiv_{k-1} \delta(q',\sigma))$

Demostración.

P1 Es directo.

P2 Es directo pues $|w| \leq 0$ si y sólo si $w = \varepsilon$.

P3 Es consecuencia de P2.

P4 Es claro pues si $|w| \leq k-1$ entonces $|w| \leq k$

P5 Es consecuencia de P4.

P6 Ejercicio.

P7 ($\Rightarrow$) es consecuencia de P4 y P6. Para ($\Leftarrow$) obsérvese que si $|w| \leq k$ entonces $|w| \leq k-1$ o $|w| = k$. En el primer caso úsese que $q \equiv_{k-1} q'$. Para el segundo caso se tiene que $w = \sigma v$ con $\sigma \in \Sigma, v \in \Sigma^\star$ y $|v| = k-1$. La conclusión se sigue de ambas hipótesis, pues $\delta^\star(q,w) = \delta^\star(q,\sigma v) = \delta^\star(\delta(q,\sigma), v)$ y lo analogo sucede para q'.

□

Con las propiedades anteriores podemos sustituir el cálculo de las clases de equivalencia de $\equiv$ por el cálculo de las clases para $\equiv_k$, puesto que la propiedad P1 implica que para cualquier $q \in Q$:

$$[q] = \bigcap_{k \in \mathbb{N}} [q]_k$$

Aparentemente esto no soluciona el problema pues si bien las clases de k-equivalencia se pueden calcular más fácilmente pues sólo involucran a una cantidad finita de cadenas (aquellas de longitud a lo más k), es necesario calcular las clases para toda $k \in \mathbb{N}$. En un momento veremos que esto realmente no sucede así, pero primero veamos un procedimiento para construir las clases de equivalencia.

Dado un AFD $M = \langle Q, \Sigma, \delta, q_0, F\rangle$ el conjunto cociente $Q/\equiv$ puede construirse como sigue:

$Q :=\ Acc(M)$ `% estados accesibles`
$k := 0$
$P_0 := \{F, Q-F\}$ `% construir la particion inicial: estados finales y no-finales`
`repeat`
  $k := k+1$
  $P_k :=\ \{q \in\ P_{k-1}\ \mid\ \forall\ a \in \Sigma, [\delta(q,a)] = [\delta(q',a)]\ \}$
`until` $P_k = P_{k-1}$
`return` P_k

La partición P_k se construirá a partir de P_{k-1} manteniendo a dos estados q, q' en la misma clase si y sólo si para toda $a \in \Sigma$, los estados $\delta(q,a)$ y $\delta(q',a)$ estaban en la misma clase en P_{k-1}.
De esta manera P_k es en realidad la partición generada por $\equiv$, es decir $P_k = Q/_{\equiv}$.

El algoritmo anterior es parcialmente correcto, lo cual es consecuencia de las propiedades de las relaciones $\equiv_k$. Más aún la terminación del algoritmo es un corolario de la siguiente

Proposición 4. *Si $M = \langle Q, \Sigma, q_0, \delta, F\rangle$ es un AFD entonces la sucesión de particiones $P_0, P_1, \ldots, P_k$ generadas por las clases de k-equivalencia de estados se estaciona, es decir existe un $n \in \mathbb{N}$ tal que para toda $k \geq n$ se tiene que $P_k = P_n$. Más aún $n \leq |Q|$, es decir n es a lo más el número de estados de M.*

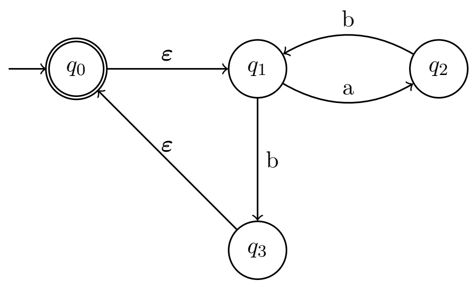
Demostración. Obsérvese que por la propiedad P5 de la proposición anterior, cada clase de k-equivalencia es subclase de una clase de $k-1$ equivalencia, es decir, la partición de P_k es un refinamiento de la partición P_{k-1}. Esto implica que el índice[1] de P_k es mayor o igual que el índice de P_{k-1}. Más aún, el índice de las particiones está acotado superiormente por el número de estados en

[1] Recuérdese que el índice de una partición es el número de clases de equivalencia que la conforman

Q, pues los refinamientos no pueden ir más allá del caso en que cada clase en P_k sea unitaria, esto sucede cuando P_k es la partición total (y el AFD original ya era mínimo). En el peor de los casos cada iteración del algoritmo genera únicamente una nueva clase por lo que no es posible iterar más allá de $n = |Q|$ veces para lograr que las particiones se estacionen. $\square$

4. Ejemplo

Sea M el siguiente autómata finito no-determinístico con transiciones ε:

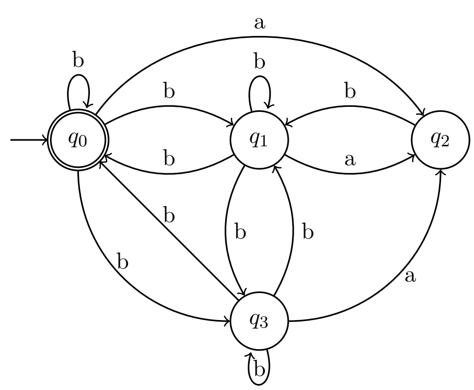


Usando los métodos descritos antes se obtendrá un autómata mínimo determinista de la siguiente forma:

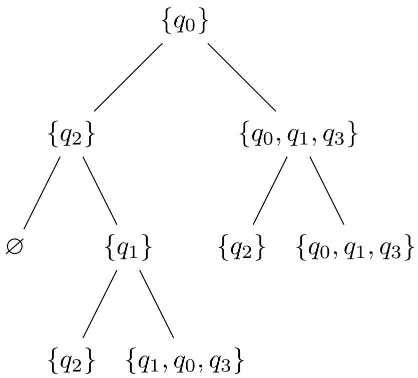
1. Se eliminan las ε-transiciones calculando los conjuntos Cl_ε de cada estado:

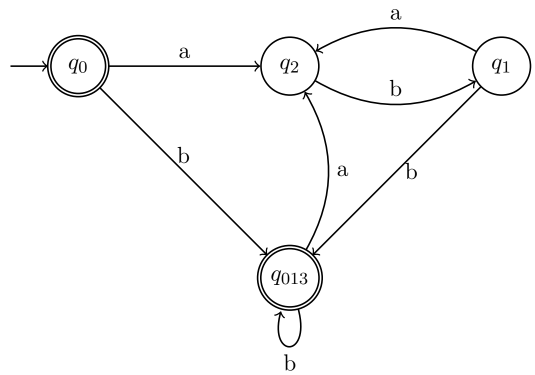
Q	Cl_ε
q_0	$\{q_0, q_1\}$
q_1	$\{q_1\}$
q_2	$\{q_2\}$
q_3	$\{q_3, q_0, q_1\}$

Q	a	b
q_0	$\{q_2\}$	$\{q_3, q_0, q_1\}$
q_1	$\{q_2\}$	$\{q_3, q_0, q_1\}$
q_2	$\varnothing$	$\{q_1\}$
q_3	$\{q_2\}$	$\{q_3, q_0, q_1\}$



2. Se transforma el autómata anterior en determinista:





3. Finalmente se minimiza:

	A		B	
	$\{q_0\}$	$\{q_0, q_1, q_3\}$	$\{q_2\}$	$\{q_1\}$
a	B	B	$--$	B
b	A	A	B	A

		A	B	C
	$\{q_0\}$	$\{q_0, q_1, q_3\}$	$\{q_2\}$	$\{q_1\}$
a	B	B	$--$	B
b	A	A	C	A