Teoría de la Computación

Clase 5: Teorema de Kleene

Mauro Artigiani

09 agosto 2021

Universidad del Rosario, Bogotá

Repaso

Definición

Decimos que R es una expresión regular si R es

1. a para algún elemento del alfabeto Σ;
2. ε;
3. $\emptyset$;
4. $(R_1 \cup R_2)$, donde R_1 y R_2 son expresiones regulares;
5. $(R_1 \circ R_2)$, donde R_1 y R_2 son expresiones regulares;
6. R_1^*, donde R_1 es una expresión regular;

Teorema de Kleene

Hemos enunciado el siguiente

Teorema de Kleene

Un lenguaje es regular si y solo si hay una expresión regular que lo describe.

Teorema de Kleene

Hemos enunciado el siguiente

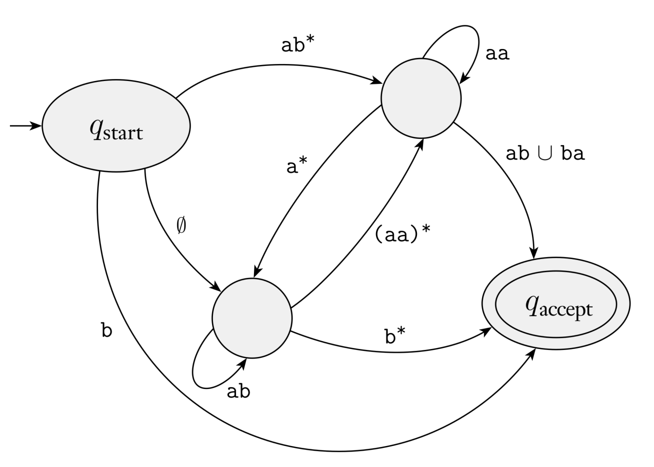
Teorema de Kleene

Un lenguaje es regular si y solo si hay una expresión regular que lo describe.

La otra vez hemos visto una implicación. Hoy veremos la otra.

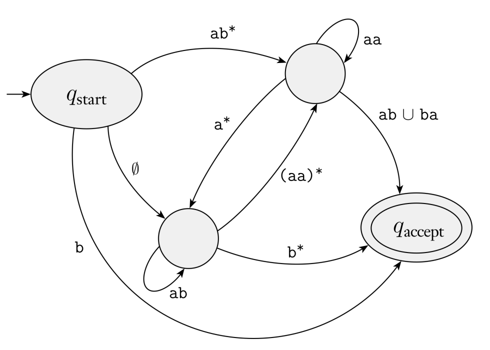
GNFAs

GNFA



Ejemplo

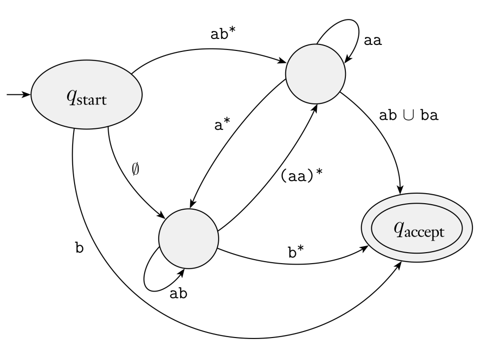
Procesamiento de $w = abbaaba$:



Procesamiento de los GNFAs

Ejemplo

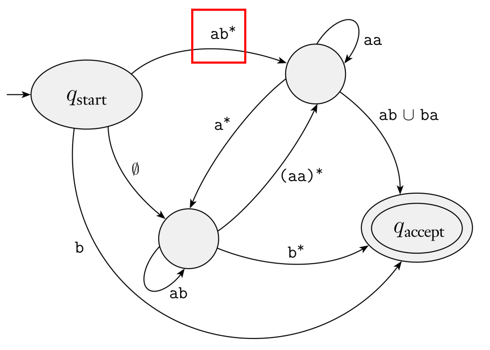
Procesamiento de w = (abb)(a)(ab)(a):



Procesamiento de los GNFAs

Ejemplo

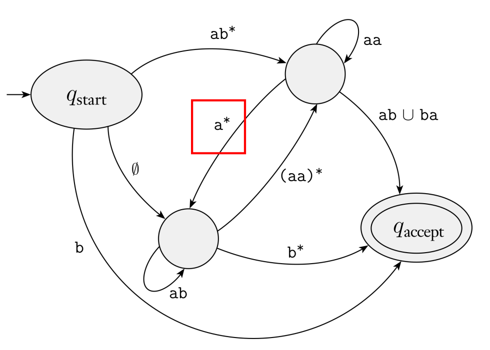
Procesamiento de w = (abb)(a)(ab)(a):



Procesamiento de los GNFAs

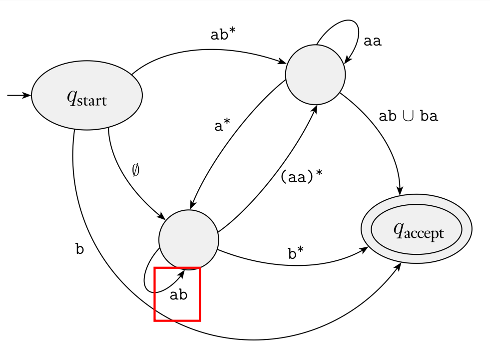
Ejemplo

Procesamiento de w = (abb)(a)(ab)(a):



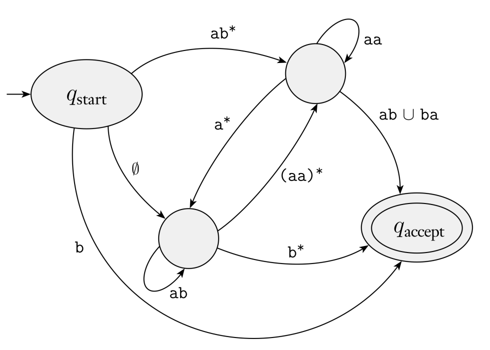
Ejemplo

Procesamiento de w = (abb)(a)(ab)(a):



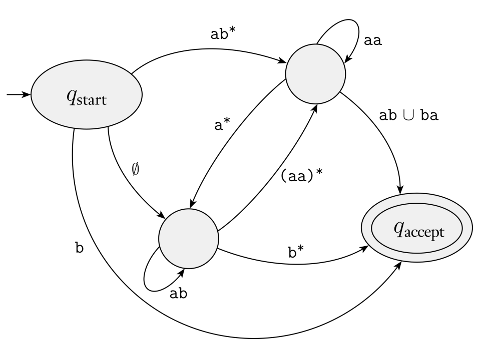
Ejemplo

Procesamiento de w = (abb)(a)(ab)(a): Cómputo interrumpido



Ejemplo

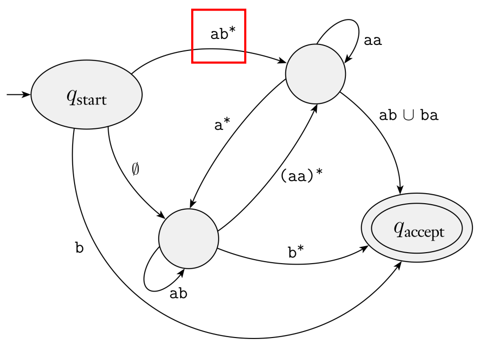
Procesamiento de w = (abb)(aa)(ba):



Procesamiento de los GNFAs

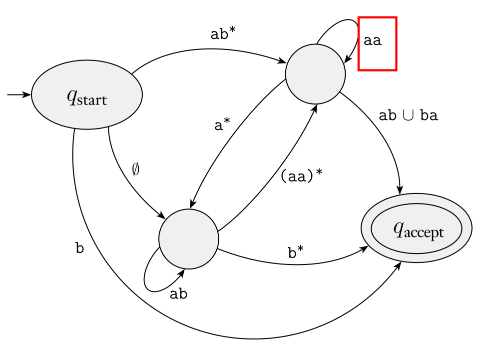
Ejemplo

Procesamiento de w = (abb)(aa)(ba):



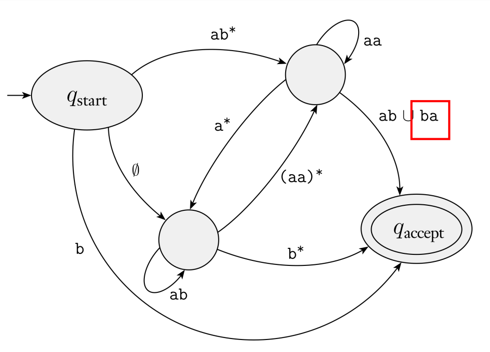
Ejemplo

Procesamiento de w = (abb)(aa)(ba):



Ejemplo

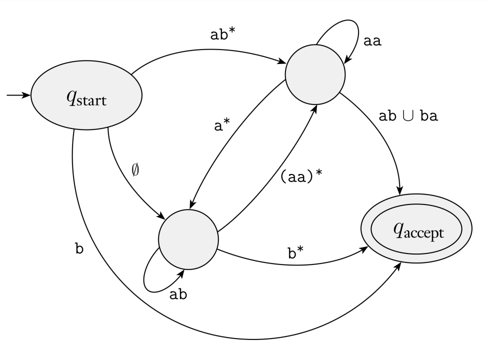
Procesamiento de w = (abb)(aa)(ba):



Procesamiento de los GNFAs

Ejemplo

Procesamiento de w = (abb)(aa)(ba): Cadena aceptada



GNFA

Para simplificar nuestro trabajo, vamos a pedir unas condiciones adicionales a nuestros GNFAs.

- El estado inicial tiene flechas que salen de él para *cualquier* otro estado y *no* tiene ninguna flecha entrante.
- Hay un único estado de aceptación, distinto del estado inicial. Este estado *no* tiene ninguna flecha que sale de él y *todos* los demás estados tienen una flecha que sale de ellos para llegar al estado final.
- Cualquier estado que no sea ni final ni inicial tiene una *única* flecha que lo conecta a *cada uno* de los otros estados (*incluyéndolo* a él mismo).

GNFA

Siempre se puede transformar un DFA en un GNFA equivalente que satisface las condiciones de antes:

- Añadimos un estado inicial nuevo con una transición ε al viejo estado inicial;

GNFA

Siempre se puede transformar un DFA en un GNFA equivalente que satisface las condiciones de antes:

- Añadimos un estado inicial nuevo con una transición ε al viejo estado inicial;
- Añadimos un nuevo estado final con una transición ε hacia él desde todos los viejos estados finales (que ya no lo son);

GNFA

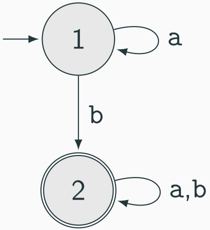
Siempre se puede transformar un DFA en un GNFA equivalente que satisface las condiciones de antes:

- Añadimos un estado inicial nuevo con una transición ε al viejo estado inicial;
- Añadimos un nuevo estado final con una transición ε hacia él desde todos los viejos estados finales (que ya no lo son);
- Si hay múltiples flechas conectados dos estados, ponemos una sola flecha cuya etiqueta sea la unión de las etiquetas de antes;

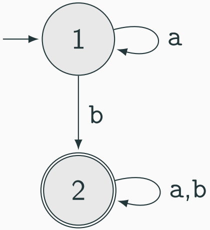
Siempre se puede transformar un DFA en un GNFA equivalente que satisface las condiciones de antes:

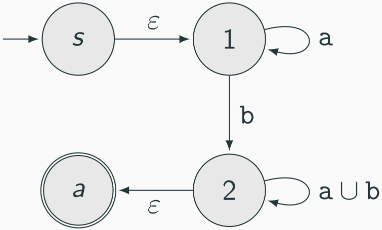
- Añadimos un estado inicial nuevo con una transición ε al viejo estado inicial;
- Añadimos un nuevo estado final con una transición ε hacia él desde todos los viejos estados finales (que ya no lo son);
- Si hay múltiples flechas conectados dos estados, ponemos una sola flecha cuya etiqueta sea la unión de las etiquetas de antes;
- Finalmente, añadimos flechas $\emptyset$ entre estados que antes no estaban conectados.

Ejemplo



Ejemplo





Teorema de Kleene: segunda parte

Teorema de Kleene: segunda parte

Lema

Si A es un lenguaje regular, entonces existe una expresión regular R tal que $L(R) = A$.

Teorema de Kleene: segunda parte

Lema

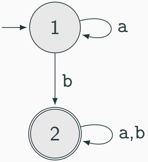
Si A es un lenguaje regular, entonces existe una expresión regular R tal que $L(R) = A$.

Demostración

Sea A un lenguaje regular y sea M un `DFA` que lo reconoce. Empezamos convirtiendo M en un `GNFA` G que satisface nuestro requerimientos. Mostraremos que podemos reducir G a G' un `GNFA` con solo dos estados. La etiqueta entre los dos estados será la expresión regular que corresponde a A.

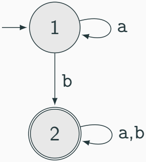
Un ejemplo

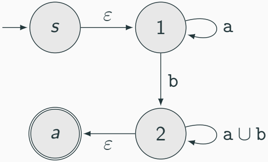
Sea A el lenguaje regular reconocido por el siguiente autómata:



Un ejemplo

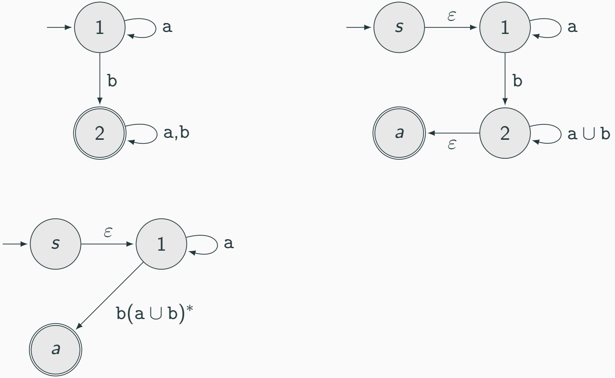
Sea A el lenguaje regular reconocido por el siguiente autómata:





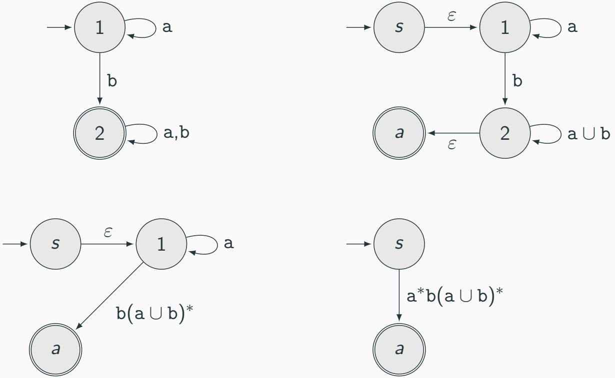
Un ejemplo

Sea A el lenguaje regular reconocido por el siguiente autómata:



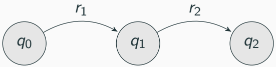
Un ejemplo

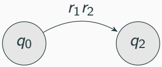
Sea A el lenguaje regular reconocido por el siguiente autómata:

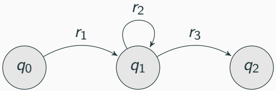


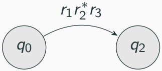
Casos de simplificación (1/2)

Casos de simplificación (1/2)

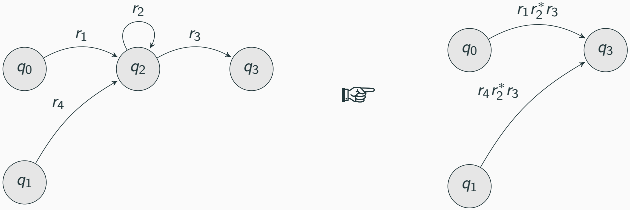




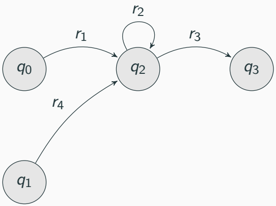




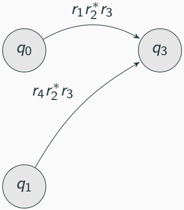
Casos de simplificación (2/2)

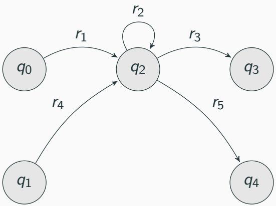


Casos de simplificación (2/2)

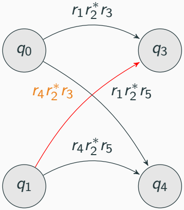


☞



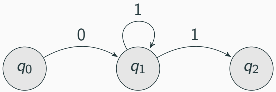


☞



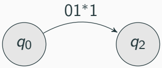
Ejemplos (1/2)

Ejemplos (1/2)

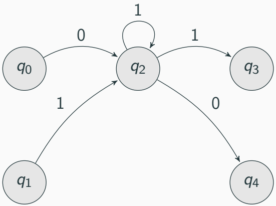


1
0
1
q_0
q_1
q_2

☞

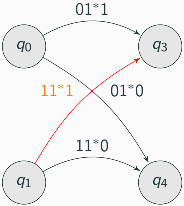


01*1
q_0
q_2



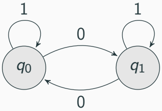
1
0
1
q_0
q_2
q_3
1
0
q_1
q_4

☞



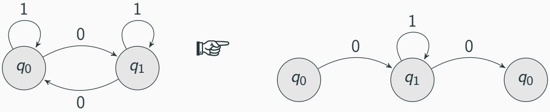
01*1
q_0
q_3
11*1
01*0
11*0
q_1
q_4

Ejemplos (2/2)



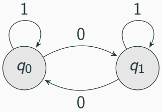
☞

Ejemplos (2/2)

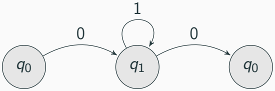


Organizamos las transiciones que llegan a q_1

Ejemplos (2/2)



☞



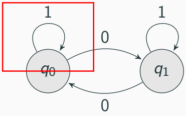
Organizamos las transiciones que llegan a q_1

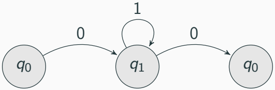
.

☞

Simplificamos

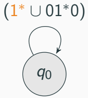
Ejemplos (2/2)





Organizamos las transiciones que llegan a q_1

Simplificamos



Unimos con el bucle de q_0

$\texttt{CONVERT}(G)$

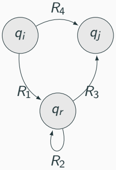
1. Sea k el número de estados de G.

CONVERT(G)

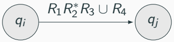
1. Sea k el número de estados de G.
2. Si $k = 2$, entonces G tiene solamente un estado inicial y uno final. La única flecha que conecta el estado inicial con el estado final tiene como etiqueta una expresión regular R. Retornamos R.

$\texttt{CONVERT}(G)$

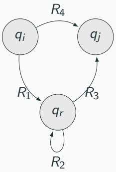
1. Sea k el número de estados de G.
2. Si $k = 2$, entonces G tiene solamente un estado inicial y uno final. La única flecha que conecta el estado inicial con el estado final tiene como etiqueta una expresión regular R. Retornamos R.
3. Si $k > 2$, escogemos un estado $q_r \in Q$ diferente de los estados inicial y final.



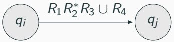
Simplificamos



1. Sea k el número de estados de G.
2. Si $k = 2$, entonces G tiene solamente un estado inicial y uno final. La única flecha que conecta el estado inicial con el estado final tiene como etiqueta una expresión regular R. Retornamos R.
3. Si $k > 2$, escogemos un estado $q_r \in Q$ diferente de los estados inicial y final.



Simplificamos



4. Retornamos CONVERT(G')

$\texttt{CONVERT}(G)$

Mostramos que el procedimiento es correcto:

Afirmación

Para cualquier $\texttt{GNFA}$ G, $\texttt{CONVERT}(G)$ es equivalente a G.

CONVERT(G)

Mostramos que el procedimiento es correcto:

Afirmación

Para cualquier GNFA G, CONVERT(G) es equivalente a G.

Demostración

La demostración es por *inducción* sobre k.

Caso base: Si $k = 2$, entonces G tiene solamente un estado inicial y uno final. La única flecha que conecta el estado inicial con el estado final tiene como etiqueta una expresión regular R, la cual describe el lenguaje que G reconoce. Entonces R es equivalente a G.

Continuación de la demostración (1/2)

Paso inductivo: Supongamos que la afirmación es cierta para $k-1$. Demostraremos para k que $L(G) = L(G')$.

Continuación de la demostración (1/2)

Paso inductivo: Supongamos que la afirmación es cierta para $k-1$. Demostraremos para k que $L(G) = L(G')$.

$\Rightarrow$) Supongamos que G acepta una palabra w. Esto significa que hay una forma de procesar w por los estados q_{start}, q_1, $\ldots$, q_{accept}. Tenemos dos casos:

Continuación de la demostración (1/2)

Paso inductivo: Supongamos que la afirmación es cierta para $k-1$. Demostraremos para k que $L(G) = L(G')$.

$\Rightarrow$) Supongamos que G acepta una palabra w. Esto significa que hay una forma de procesar w por los estados q_{start}, q_1, $\ldots$, q_{accept}. Tenemos dos casos:

A. Ninguno de los estados es q_r. Entonces G' también acepta w.

Continuación de la demostración (1/2)

Paso inductivo: Supongamos que la afirmación es cierta para $k-1$. Demostraremos para k que $L(G) = L(G')$.

$\Rightarrow$) Supongamos que G acepta una palabra w. Esto significa que hay una forma de procesar w por los estados q_{start}, q_1, $\dots$, q_{accept}. Tenemos dos casos:

A. Ninguno de los estados es q_r. Entonces G' también acepta w.
B. Alguno de los estados es q_r, es decir, el procesamiento pasa por q_{start}, q_1, $\dots$, q_i, q_r, q_j, $\dots$, q_{accept}. Por el proceso de simplificación (ver paso 3 del algoritmo), en G' existe una flecha entre q_i y q_j con una expresión regular que describe cómo moverse de q_i a q_j pasando por q_r. Entonces G' también acepta w.

Continuación de la demostración (1/2)

Paso inductivo: Supongamos que la afirmación es cierta para $k-1$. Demostraremos para k que $L(G) = L(G')$.

$\Rightarrow$) Supongamos que G acepta una palabra w. Esto significa que hay una forma de procesar w por los estados q_{start}, q_1, $\ldots$, q_{accept}. Tenemos dos casos:

A. Ninguno de los estados es q_r. Entonces G' también acepta w.
B. Alguno de los estados es q_r, es decir, el procesamiento pasa por q_{start}, q_1, $\ldots$, q_i, q_r, q_j, $\ldots$, q_{accept}. Por el proceso de simplificación (ver paso 3 del algoritmo), en G' existe una flecha entre q_i y q_j con una expresión regular que describe cómo moverse de q_i a q_j pasando por q_r. Entonces G' también acepta w.

En cualquier caso, G' también acepta w.

Continuación de la demostración (2/2)

$\Leftarrow$) Supongamos que G' acepta una palabra w. Cada flecha entre dos estados de G' me dice cómo moverme entre los mismos estados en G. Entonces, G también acepta w.

Por lo tanto, G y G' son equivalentes.

Continuación de la demostración (2/2)

$\Leftarrow$) Supongamos que G' acepta una palabra w. Cada flecha entre dos estados de G' me dice cómo moverme entre los mismos estados en G. Entonces, G también acepta w.

Por lo tanto, G y G' son equivalentes. Como G' tiene $k-1$ estados, por hipótesis de inducción se sigue que $\texttt{CONVERT}(G')$ es equivalente a G' y en consecuencia G es equivalente a $\texttt{CONVERT}(G)$.

Continuación de la demostración (2/2)

$\Leftarrow$) Supongamos que G' acepta una palabra w. Cada flecha entre dos estados de G' me dice cómo moverme entre los mismos estados en G. Entonces, G también acepta w.

Por lo tanto, G y G' son equivalentes. Como G' tiene $k-1$ estados, por hipótesis de inducción se sigue que $\texttt{CONVERT}(G')$ es equivalente a G' y en consecuencia G es equivalente a $\texttt{CONVERT}(G)$.

Hemos demostrado para todo G que G es equivalente a $\texttt{CONVERT}(G)$. $\square$

Resumen

Resumen

Hoy aprendimos:

- La definición de los autómatas finitos no deterministas generalizados (`GNFA`);
- Encontrar el lenguaje regular reconocido por un `DFA`;
- A construir un `GNFA` que reconozca el mismo lenguaje de un `DFA`.