Teoría de la Computación

Clase 6: Lenguajes no regulares, Lema del bombeo

Mauro Artigiani

13 agosto 2020

Universidad del Rosario, Bogotá

Repaso

Lenguajes regulares

En las clases pasadas hemos definido los lenguajes regulares como los lenguajes reconocidos por autómatas de estados finitos (deterministas o no). También hemos visto que los lenguajes regulares se pueden describir por expresiones regulares.

Lenguajes regulares

En las clases pasadas hemos definido los lenguajes regulares como los lenguajes reconocidos por autómatas de estados finitos (deterministas o no). También hemos visto que los lenguajes regulares se pueden describir por expresiones regulares.

No todos los lenguajes son regulares.

Lenguajes regulares

En las clases pasadas hemos definido los lenguajes regulares como los lenguajes reconocidos por autómatas de estados finitos (deterministas o no). También hemos visto que los lenguajes regulares se pueden describir por expresiones regulares.

No todos los lenguajes son regulares.

En esta clase hablaremos de las limitaciones de los lenguajes regulares y como demostrar que un lenguaje no es regular. Más adelante en el curso introduciremos una clase más grande de lenguajes, que generaliza los lenguajes regulares.

Lenguajes no regulares

Limitaciones de los lenguajes regulares

Los lenguajes regulares son reconocidos por autómatas de estados *finitos*. Una manera intuitiva de tratar de construir un lenguaje no regular es construir una palabra que solo pueda ser aceptada por un autómata con un número *infinito* de estados.

Limitaciones de los lenguajes regulares

Los lenguajes regulares son reconocidos por autómatas de estados *finitos*. Una manera intuitiva de tratar de construir un lenguaje no regular es construir una palabra que solo pueda ser aceptada por un autómata con un número *infinito* de estados.

Por ejemplos, si $\Sigma = \{0, 1\}$, tomamos en consideración:

$$B = \{0^n 1^n, n \geq 0\}.$$

Un autómata que reconociera el lenguaje B necesitaría tener en un su memoria cuantos 0 están en la palabra, y este número es arbitrariamente grande. Así que parece que un autómata con un número finitos de estados no pudiera alcanzar a reconocer B.

Problemas de la intuición

Acabamos de convencernos, de manera intuitiva, que el lenguaje B no puede ser regular. Desafortunadamente, la intuición puede fallar. Consideramos, siendo $\Sigma = \{0, 1\}$,

$$C = \{w, w \text{ tiene un número igual de 0 y 1}\},$$

y

$$D = \{w, w \text{ tiene el mismo número de subpalabras 01 y 10}\}.$$

Problemas de la intuición

Acabamos de convencernos, de manera intuitiva, que el lenguaje B no puede ser regular. Desafortunadamente, la intuición puede fallar. Consideramos, siendo $\Sigma = \{0, 1\}$,

$$C = \{w, w \text{ tiene un número igual de 0 y 1}\},$$

y

$$D = \{w, w \text{ tiene el mismo número de subpalabras 01 y 10}\}.$$

A un primer vistazo, tanto C, como D parecen tener una naturaleza "infinita" y entonces no ser lenguajes regulares.

Problemas de la intuición

Acabamos de convencernos, de manera intuitiva, que el lenguaje B no puede ser regular. Desafortunadamente, la intuición puede fallar. Consideramos, siendo $\Sigma = \{0, 1\}$,

$$C = \{w, w \text{ tiene un número igual de 0 y 1}\},$$

y

$$D = \{w, w \text{ tiene el mismo número de subpalabras 01 y 10}\}.$$

A un primer vistazo, tanto C, como D parecen tener una naturaleza "infinita" y entonces no ser lenguajes regulares. De manera un poco sorprendente, se puede ver que D es regular.

Problemas de la intuición

Acabamos de convencernos, de manera intuitiva, que el lenguaje B no puede ser regular. Desafortunadamente, la intuición puede fallar. Consideramos, siendo $\Sigma = \{0, 1\}$,

$$C = \{w, w \text{ tiene un número igual de 0 y 1}\},$$

y

$$D = \{w, w \text{ tiene el mismo número de subpalabras 01 y 10}\}.$$

A un primer vistazo, tanto C, como D parecen tener una naturaleza "infinita" y entonces no ser lenguajes regulares. De manera un poco sorprendente, se puede ver que D es regular.

Esto nos muestra que nuestra intuición no es suficiente para demostrar que un lenguaje no es regular. Necesitamos una herramienta teórica.

El lema del bombeo

El lema de bombeo

El lema del bombeo (en inglés: “pumping lemma”), será nuestra herramienta para demostrar de manera clara que un lenguaje no es regular.

El lema de bombeo

El lema del bombeo (en inglés: "pumping lemma"), será nuestra herramienta para demostrar de manera clara que un lenguaje no es regular.

La idea atrás del lema del bombeo es que, si una palabra hace parte de un lenguaje regular, se puede *inflar* en el sentido que contiene una subpalabra que se puede repetir cuantas veces queremos generando nuevas palabras del lenguaje mismo.

El lema de bombeo

Lema (Lema del bombeo)

Sea A un lenguaje regular. Luego existe un número p, llamado la longitud de bombeo, tal que cualquier palabra s de A de longitud por lo menos p se puede descomponer en $s = xyz$, donde

1. *para todos $i \geq 0$, se tiene $xy^iz \in A$;*
2. $|y| > 0$;
3. $|xy| \leq p$.

El lema de bombeo

Lema (Lema del bombeo)

Sea A un lenguaje regular. Luego existe un número p, llamado la longitud de bombeo, tal que cualquier palabra s de A de longitud por lo menos p se puede descomponer en $s = xyz$, donde

1. *para todos $i \geq 0$, se tiene $xy^i z \in A$;*
2. $|y| > 0$;
3. $|xy| \leq p$.

Sin la segunda condición el Lema sería cierto de manera trivial: $s = x$, $y = z = \varepsilon$.

El lema de bombeo

Lema (Lema del bombeo)

Sea A un lenguaje regular. Luego existe un número p, llamado la longitud de bombeo, tal que cualquier palabra s de A de longitud por lo menos p se puede descomponer en $s = xyz$, donde

1. *para todos $i \geq 0$, se tiene $xy^i z \in A$;*
2. $|y| > 0$;
3. $|xy| \leq p$.

Sin la segunda condición el Lema sería cierto de manera trivial: $s = x$, $y = z = \varepsilon$. La tercera condición es una condición técnica que nos ayudará al momento de aplicar el Lema del bombeo a un lenguaje.

El lema de bombeo

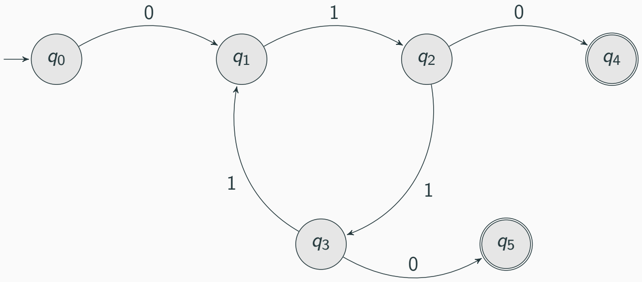
Lema (Lema del bombeo)

Sea A un lenguaje regular. Luego existe un número p, llamado la longitud de bombeo, tal que cualquier palabra s de A de longitud por lo menos p se puede descomponer en $s = xyz$, donde

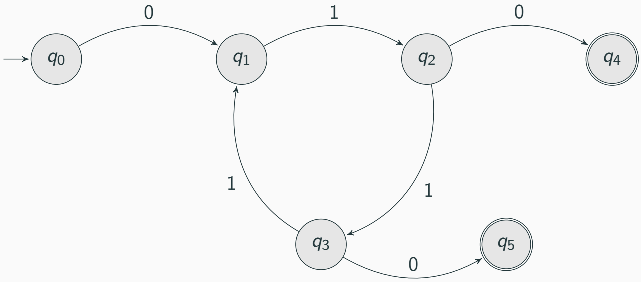
1. *para todos $i \geq 0$, se tiene $xy^iz \in A$;*
2. $|y| > 0$;
3. $|xy| \leq p$.

Sin la segunda condición el Lema sería cierto de manera trivial: $s = x$, $y = z = \varepsilon$. La tercera condición es una condición técnica que nos ayudará al momento de aplicar el Lema del bombeo a un lenguaje. La primera condición dice exactamente que podemos inflar (o desinflar) la palabra s repitiendo cuanto queremos la subpalabra y.

El lema del bombeo: ejemplo

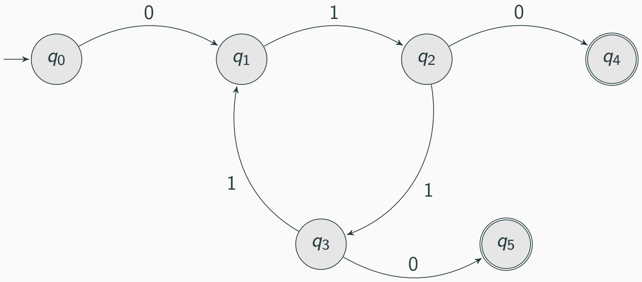


El lema del bombeo: ejemplo



$$q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_4$$

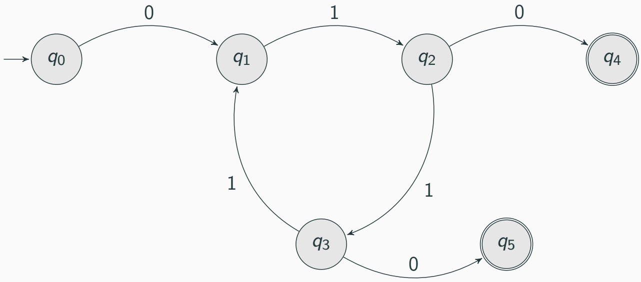
El lema del bombeo: ejemplo



$$q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_4$$

$$q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{1} q_3 \xrightarrow{1} q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_4$$

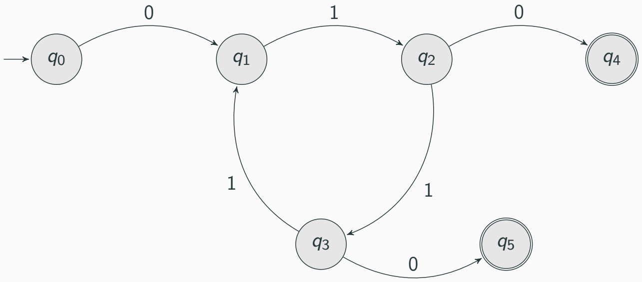
El lema del bombeo: ejemplo



$$q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_4$$

$$q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{1} q_3 \xrightarrow{1} q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_4$$

El lema del bombeo: ejemplo

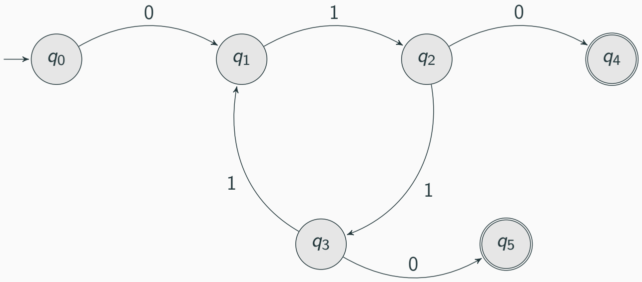


$$q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_4$$

$$q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{1} q_3 \xrightarrow{1} q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_4$$

$$q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{1} q_3 \xrightarrow{1} q_1 \xrightarrow{1} q_2 \xrightarrow{1} q_3 \xrightarrow{1} q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_4$$

El lema del bombeo: ejemplo



$$q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_4$$

$$q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{1} q_3 \xrightarrow{1} q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_4$$

$$q_0 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{1} q_3 \xrightarrow{1} q_1 \xrightarrow{1} q_2 \xrightarrow{1} q_3 \xrightarrow{1} q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_4$$

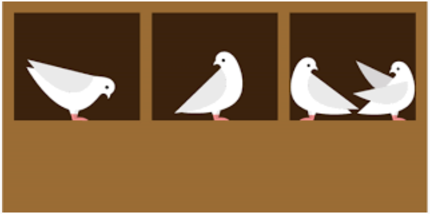
¿Por qué sabemos que una cadena de longitud 12 que sea aceptada por este autómata debe procesarse mediante un bucle?

El lema del bombeo

Antes de demostrar el Lema, acordémonos del principio del palomar

Principio del palomar

Si queremos poner más de p palomas en menos de p cajas, entonces hay por lo menos una caja con más de una paloma.

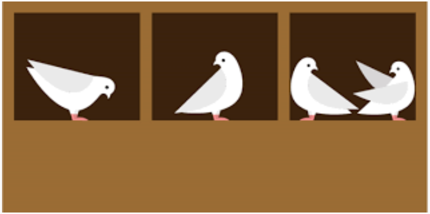


El lema del bombeo

Antes de demostrar el Lema, acordémonos del principio del palomar

Principio del palomar

Si queremos poner más de p palomas en menos de p cajas, entonces hay por lo menos una caja con más de una paloma.



Queremos utilizar el principio del palomar para demostrar el Lema del bombeo.

Demostración del Lema del bombeo

Siendo A un lenguaje regular, existe un `DFA` $M = (Q, \Sigma, \delta, q_1, F)$ que lo reconoce. Definamos la longitud de bombeo p como el número de estados de M: $p = |Q|$.

Demostración del Lema del bombeo

Siendo A un lenguaje regular, existe un $\texttt{DFA}$ $M = (Q, \Sigma, \delta, q_1, F)$ que lo reconoce. Definamos la longitud de bombeo p como el número de estados de M: $p = |Q|$.

Queremos mostrar que cualquier palabra s en A con longitud por lo menos p se puede descomponer en $s = xyz$, donde las subpalabras x, y, z satisfacen las conclusiones del Lema.

Demostración del Lema del bombeo

Siendo A un lenguaje regular, existe un `DFA` $M = (Q, \Sigma, \delta, q_1, F)$ que lo reconoce. Definamos la longitud de bombeo p como el número de estados de M: $p = |Q|$.

Queremos mostrar que cualquier palabra s en A con longitud por lo menos p se puede descomponer en $s = xyz$, donde las subpalabras x, y, z satisfacen las conclusiones del Lema.

El caso más fácil es cuando A no tiene ninguna palabra de longitud p o más. En este caso el Lema es cierto *por vacuidad*.

Demostración del Lema del bombeo

Ahora, sea $s \in A$ con $|s| = n \geq p = |Q|$. Siendo s aceptada por M, existe una secuencia de estados $q_1, q_2, \ldots, q_{n+1}$, con $q_{n+1} \in F$, que describe el movimiento de M mientra lee la palabra s.

Demostración del Lema del bombeo

Ahora, sea $s \in A$ con $|s| = n \geq p = |Q|$. Siendo s aceptada por M, existe una secuencia de estados $q_1, q_2, \ldots, q_{n+1}$, con $q_{n+1} \in F$, que describe el movimiento de M mientra lee la palabra s.

Siendo $n + 1 > n \geq p$ y siendo p el número de estados de M, por el principio del palomar existe un estado $q \in Q$ en donde pasamos más de una vez durante el cálculo de la palabra s.

Demostración del Lema del bombeo

Ahora, sea $s \in A$ con $|s| = n \geq p = |Q|$. Siendo s aceptada por M, existe una secuencia de estados $q_1, q_2, \ldots, q_{n+1}$, con $q_{n+1} \in F$, que describe el movimiento de M mientra lee la palabra s.

Siendo $n + 1 > n \geq p$ y siendo p el número de estados de M, por el principio del palomar existe un estado $q \in Q$ en donde pasamos más de una vez durante el cálculo de la palabra s.

$$\to q_1 \xrightarrow{s_1} q_2 \xrightarrow{s_2} \cdots \xrightarrow{s_i} q \xrightarrow{s_{i+1}} \cdots \xrightarrow{s_j} q \xrightarrow{s_{j+1}} \cdots \xrightarrow{s_n} q_{n+1}$$

Demostración del Lema del bombeo

Ahora, sea $s \in A$ con $|s| = n \geq p = |Q|$. Siendo s aceptada por M, existe una secuencia de estados $q_1, q_2, \ldots, q_{n+1}$, con $q_{n+1} \in F$, que describe el movimiento de M mientra lee la palabra s.

Siendo $n + 1 > n \geq p$ y siendo p el número de estados de M, por el principio del palomar existe un estado $q \in Q$ en donde pasamos más de una vez durante el cálculo de la palabra s.

$$\rightarrow q_1 \overbrace{\xrightarrow{s_1} q_2 \xrightarrow{s_2} \cdots \xrightarrow{s_i}}^{x} q \overbrace{\xrightarrow{s_{i+1}} \cdots \xrightarrow{s_j}}^{y} q \overbrace{\xrightarrow{s_{j+1}} \cdots \xrightarrow{s_n}}^{z} q_{n+1}$$

Descomponemos s en la siguiente manera: $x = s_1 s_2 \ldots s_{i-1}$, $y = s_i s_{i+1} \ldots s_j$, $z = s_{j+1} s_{j+2} \ldots s_n$. Nótese que y se puede quitar o repetir tantas veces como queremos, exactamente como pedía la propiedad 1 del Lema.

Demostración del Lema del bombeo

$$\rightarrow q_1 \overbrace{\xrightarrow{s_1} q_2 \xrightarrow{s_2} \cdots \xrightarrow{s_i}}^{x} q \overbrace{\xrightarrow{s_{i+1}} \cdots \xrightarrow{s_j}}^{y} q \overbrace{\xrightarrow{s_{j+1}} \cdots \xrightarrow{s_n}}^{z} q_{n+1}$$

La subpalabra y está entre dos ocurrencias de q, entonces tiene por lo menos un símbolo, como pedía la propiedad 2 del Lema.

Demostración del Lema del bombeo

$$\rightarrow q_1 \overbrace{\xrightarrow{s_1} q_2 \xrightarrow{s_2} \cdots \xrightarrow{s_i}}^{x} q \overbrace{\xrightarrow{s_{i+1}} \cdots \xrightarrow{s_j}}^{y} q \overbrace{\xrightarrow{s_{j+1}} \cdots \xrightarrow{s_n}}^{z} q_{n+1}$$

La subpalabra y está entre dos ocurrencias de q, entonces tiene por lo menos un símbolo, como pedía la propiedad 2 del Lema.

Ahora, siendo q el *primer* estado repetido, necesariamente su repetición aparece en los primeros $p+1 = |Q|+1$ estados, por el Principio del palomar, es decir $|xy| \leq p$, como pedía la propiedad 3 del Lema.

$$\rightarrow \underbrace{q_1 \overbrace{\xrightarrow{s_1} q_2 \xrightarrow{s_2} \cdots \xrightarrow{s_i}}^{x} q \overbrace{\xrightarrow{s_{i+1}} \cdots \xrightarrow{s_j}}^{y} q}_{\text{máximo } p+1 \text{ estados}} \overbrace{\xrightarrow{s_{j+1}} \cdots \xrightarrow{s_n}}^{z} q_{n+1} \quad \square$$

Ejemplos

El lenguaje B

Vamos a mostrar que

$$B = \{0^n 1^n, n \geq 0\},$$

no es regular.

El lenguaje B

Vamos a mostrar que

$$B = \{0^n 1^n, n \geq 0\},$$

no es regular.

La demostración es por contradicción. Supongamos, para buscar una contradicción que B sea regular. Entonces sea p la longitud de bombeo de B. Consideramos la palabra $s = 0^p 1^p$. Tenemos que $s \in B$ y $|s| > p$, entonces necesariamente se puede descomponer en $s = xyz$ como nos asegura el Lema del bombeo, y todas la palabras $xy^i z \in B$ con $i \geq 0$.

El lenguaje B

$s = 0^p 1^p = xyz$. Tenemos tres casos

1. La subpalabra y solo contiene 0. Entonces la palabra $xyyz$ tiene más ceros que unos y no es una palabra de B.

El lenguaje B

$s = 0^p 1^p = xyz$. Tenemos tres casos

1. La subpalabra y solo contiene 0. Entonces la palabra $xyyz$ tiene más ceros que unos y no es una palabra de B.
2. De manera similar, si la subpalabra y solo contiene 1. Entonces la palabra $xyyz$ tiene más unos que ceros y no es una palabra de B.

El lenguaje B

$s = 0^p 1^p = xyz$. Tenemos tres casos

1. La subpalabra y solo contiene 0. Entonces la palabra $xyyz$ tiene más ceros que unos y no es una palabra de B.
2. De manera similar, si la subpalabra y solo contiene 1. Entonces la palabra $xyyz$ tiene más unos que ceros y no es una palabra de B.
3. Finalmente, y puede contener ceros y unos. En este caso la palabra $xyyz$ tiene el mismo número de ceros y unos, pero en el *orden* incorrecto y, por lo tanto, no pertenece a B.

El lenguaje B

$s = 0^p 1^p = xyz$. Tenemos tres casos

1. La subpalabra y solo contiene 0. Entonces la palabra $xyyz$ tiene más ceros que unos y no es una palabra de B.
2. De manera similar, si la subpalabra y solo contiene 1. Entonces la palabra $xyyz$ tiene más unos que ceros y no es una palabra de B.
3. Finalmente, y puede contener ceros y unos. En este caso la palabra $xyyz$ tiene el mismo número de ceros y unos, pero en el *orden* incorrecto y, por lo tanto, no pertenece a B.

El lenguaje B

$s = 0^p 1^p = xyz$. Tenemos tres casos

1. La subpalabra y solo contiene 0. Entonces la palabra $xyyz$ tiene más ceros que unos y no es una palabra de B.
2. De manera similar, si la subpalabra y solo contiene 1. Entonces la palabra $xyyz$ tiene más unos que ceros y no es una palabra de B.
3. Finalmente, y puede contener ceros y unos. En este caso la palabra $xyyz$ tiene el mismo número de ceros y unos, pero en el *orden* incorrecto y, por lo tanto, no pertenece a B.

En todos casos hemos llegados a una contradicción y entonces B no puede ser un lenguaje regular.

El lenguaje B

$s = 0^p 1^p = xyz$. Tenemos tres casos

1. La subpalabra y solo contiene 0. Entonces la palabra $xyyz$ tiene más ceros que unos y no es una palabra de B.
2. De manera similar, si la subpalabra y solo contiene 1. Entonces la palabra $xyyz$ tiene más unos que ceros y no es una palabra de B.
3. Finalmente, y puede contener ceros y unos. En este caso la palabra $xyyz$ tiene el mismo número de ceros y unos, pero en el *orden* incorrecto y, por lo tanto, no pertenece a B.

En todos casos hemos llegados a una contradicción y entonces B no puede ser un lenguaje regular.

En este caso la palabra s era relativamente sencilla de adivinar. En general, encontrar una palabra que genere una contradicción es algo bastante creativo, un poco como diseñar autómatas.

Resumen

Resumen

Hoy aprendimos:

- Que los autómatas finitos tienen limitaciones;
- Una herramienta teórica para demostrar que un lenguaje no es regular;
- Como utilizar el Lema del bombeo para mostrar que un lenguaje no es regular.