

Autómatas de Pila y Lenguajes Incontextuales

Elvira Mayordomo

Universidad de Zaragoza

5 de noviembre de 2012

Contenido de este tema

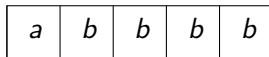
1. **Introducción a los autómatas de pila**
2. Definiciones
3. Equivalencia con gramáticas
4. Propiedades de clausura de los incontextuales
5. Lema de bombeo: lenguajes no incontextuales

¿Que es un autómata de pila?

- ▶ Un autómata de pila (AdP) es un **autómata finito** no determinista (AFnD) al que se le ha añadido una **memoria de pila**
- ▶ En cada paso **sólo se puede acceder** al último símbolo de la pila (**la cima**)
- ▶ En cada paso **se puede desapilar** el símbolo que hay **y/o apilar** otros

Aspecto de un autómata de pila

Entrada

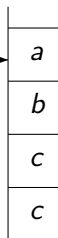


Estado



Pila

cima



Cómo funciona un autómeta de pila

- ▶ **Configuración inicial:** el autómeta está en el estado inicial q_0 leyendo al principio de la entrada y la pila sólo contiene el símbolo \$
- ▶ **La entrada se lee** de izquierda a derecha una vez
- ▶ **Cada movimiento depende de** en qué estado estoy, qué símbolo leo en la entrada y qué símbolo leo en la pila
- ▶ **En cada movimiento** cambio de estado y apilo o desapilo
- ▶ Es una máquina **no determinista**, puede haber varias opciones y puedo mover sin leer símbolo de la entrada

Ejemplo de AdP

- ▶ Estados q_0, q_1, q_2
- ▶ Estado inicial q_0
- ▶ Estado final q_2
- ▶ Alfabeto de entrada $\Sigma = \{0, 1\}$
- ▶ Alfabeto de pila $\Gamma = \{a, \$\}$
- ▶ Transiciones:

entrada	0		1		ϵ	
cima	a	$\$$	a	$\$$	a	$\$$
q_0	(q_0, aa)	$(q_0, a\$)$	(q_1, ϵ)			$(q_2, \$)$
q_1			(q_1, ϵ)			$(q_2, \$)$

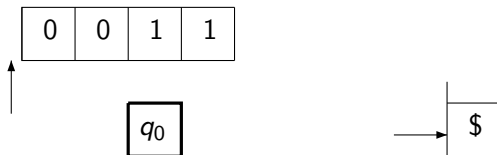
Ejemplo de AdP

entrada	0		1		ϵ	
cima	a	$\$$	a	$\$$	a	$\$$
q_0	(q_0, aa)	$(q_0, a\$)$	(q_1, ϵ)			$(q_2, \$)$
q_1			(q_1, ϵ)			$(q_2, \$)$

- ▶ Si estoy en el estado q_0 con 0 en la entrada y $\$$ en la cima:
 $(q_0, a\$)$ quiere decir **apila** a y continúa en estado q_0
- ▶ Si estoy en el estado q_0 con 1 en la entrada y a en la cima:
 (q_1, ϵ) quiere decir **desapila** a y cambia al estado q_1

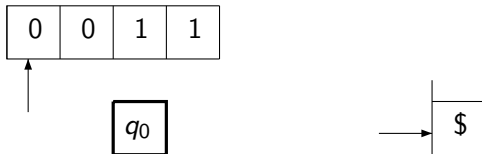
Ejemplo de AdP

- ▶ Con entrada 0011: configuración inicial



Ejemplo de AdP

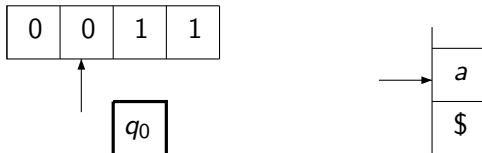
- ▶ Con entrada 0011: lectura del primer símbolo



entrada	0		1		ϵ	
cima	a	$\$$	a	$\$$	a	$\$$
q_0	(q_0, aa)	$(q_0, a\$)$	(q_1, ϵ)			$(q_2, \$)$
q_1			(q_1, ϵ)			$(q_2, \$)$

Ejemplo de AdP

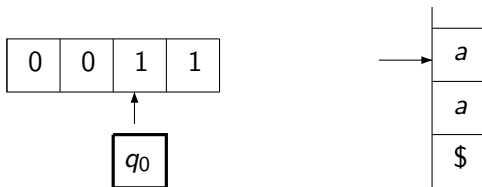
- ▶ Con entrada 0011: lectura del segundo símbolo



entrada	0		1		ϵ	
cima	a	$\$$	a	$\$$	a	$\$$
q_0	(q_0, aa)	$(q_0, a\$)$	(q_1, ϵ)			$(q_2, \$)$
q_1			(q_1, ϵ)			$(q_2, \$)$

Ejemplo de AdP

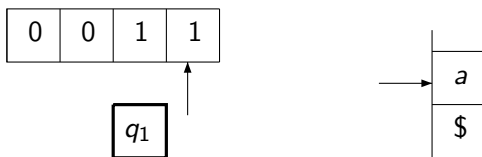
- ▶ Con entrada 0011: lectura del tercer símbolo



entrada	0		1		ϵ	
cima	a	$\$$	a	$\$$	a	$\$$
q_0	(q_0, aa)	$(q_0, a\$)$	(q_1, ϵ)			$(q_2, \$)$
q_1			(q_1, ϵ)			$(q_2, \$)$

Ejemplo de AdP

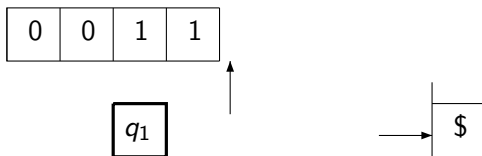
- ▶ Con entrada 0011: lectura del cuarto símbolo



entrada	0		1		ϵ	
cima	a	$\$$	a	$\$$	a	$\$$
q_0	(q_0, aa)	$(q_0, a\$)$	(q_1, ϵ)			$(q_2, \$)$
q_1			(q_1, ϵ)			$(q_2, \$)$

Ejemplo de AdP

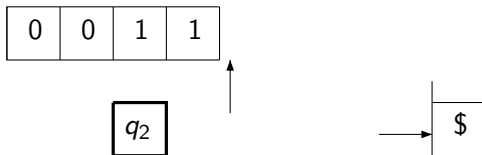
- Con entrada 0011: ϵ -transición



entrada	0		1		ϵ	
cima	a	\$	a	\$	a	\$
q_0	(q_0, aa)	$(q_0, a\$)$	(q_1, ϵ)			$(q_2, \$)$
q_1			(q_1, ϵ)			$(q_2, \$)$

Ejemplo de AdP

- ▶ Con entrada 0011: ha terminado



- ▶ Como q_2 es un estado final, **acepta la entrada**

Otra entrada

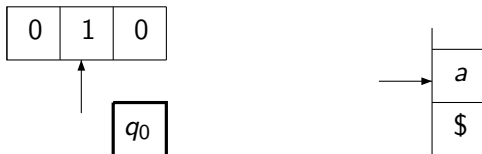
- Con entrada 010: lectura del primer símbolo



entrada	0		1		ϵ	
cima	a	\$	a	\$	a	\$
q_0	(q_0, aa)	$(q_0, a\$)$	(q_1, ϵ)			$(q_2, \$)$
q_1			(q_1, ϵ)			$(q_2, \$)$

Otra entrada

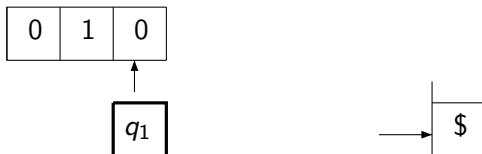
- Con entrada 010: lectura del segundo símbolo



entrada	0		1		ϵ	
cima	a	\$	a	\$	a	\$
q_0	(q_0, aa)	$(q_0, a\$)$	(q_1, ϵ)			$(q_2, \$)$
q_1			(q_1, ϵ)			$(q_2, \$)$

Otra entrada

- Con entrada 010: lectura del tercer símbolo



Rechaza la entrada porque no hay transición definida, ni en esta ni en otras computaciones

entrada	0		1		ϵ	
cima	a	\$	a	\$	a	\$
q_0	(q_0, aa)	$(q_0, a\$)$	(q_1, ϵ)			$(q_2, \$)$
q_1			(q_1, ϵ)			$(q_2, \$)$

¿Qué hace este AdP?

entrada	0		1		ϵ	
cima	a	$\$$	a	$\$$	a	$\$$
q_0	(q_0, aa)	$(q_0, a\$)$	(q_1, ϵ)			$(q_2, \$)$
q_1			(q_1, ϵ)			$(q_2, \$)$

- ▶ En el estado q_0 apila una a por cada 0
- ▶ En el estado q_1 desapila una a por cada 1

Acepta el lenguaje $\{0^n 1^n \mid n \in \mathbb{N}\}$

Contenido de este tema

1. Introducción a los autómatas de pila
2. **Definiciones**
3. Equivalencia con gramáticas
4. Propiedades de clausura de los incontextuales
5. Lema de bombeo: lenguajes no incontextuales

Definición formal de autómatata de pila

Definición

Un **autómata de pila (AdP)** es $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ tal que

- ▶ Q es el conjunto finito de estados
- ▶ Σ es el alfabeto de entrada
- ▶ Γ es el alfabeto de pila que incluye el símbolo $\$$
- ▶ $q_0 \in Q$ es el estado inicial
- ▶ $F \subseteq Q$ es el conjunto de los estados finales o de aceptación.
- ▶ $\delta : Q \times (\Sigma \cup \epsilon) \times \Gamma \rightarrow \mathcal{P}(Q \times \Gamma^*)$ es la función de transición
 $\delta(q, a, b) = R$ quiere decir que si estoy en el estado q , leo el símbolo a en la entrada y b en la cima puedo ir a cualquiera de los estados q' con $(q', c) \in R$ cambiando la cima a c

Notación: $\mathcal{P}(Q \times \Gamma^*)$ es el conjunto de subconjuntos de $Q \times \Gamma^*$

Lenguaje aceptado por un AdP

- ▶ Dado un AdP M , una computación de M con entrada w es

$$(q_0, \$)(q_1, y_1) \dots (q_m, y_m)$$

de forma que q_i es el estado e y_i es el contenido completo de la pila en el instante i , es decir

- ▶ $w = w_1 \dots w_m$ con $w_i \in \Sigma \cup \epsilon$
- ▶ $q_{i+1} \in \delta(q_i, w_{i+1}, \text{cima}(y_i))$
- ▶ Una **computación aceptadora** de M con entrada w es una computación $(q_0, \$)(q_1, y_1) \dots (q_m, y_m)$ de M con entrada w que cumple $q_m \in F$
- ▶ El **lenguaje aceptado por M** es $L(M)$ definido como

$$L(M) = \{w \mid \text{existe una computación aceptadora de } M \text{ con entrada } w\}$$

¿Cómo definimos un AdP que acepte ...

... el lenguaje $\{w \in \{a, b\}^* \mid |w|_a = |w|_b\}$

- ▶ La pila lleva la **cuenta** de número de a 's menos número de b s
- ▶ Si en la cima hay una b y leo una a **desapilo**
- ▶ Si en la cima hay una a y leo una b **desapilo**
- ▶ Si en la cima hay una a y leo una a **apilo**
- ▶ Si en la cima hay una b y leo una b **apilo**
- ▶ Si termino la entrada con la pila vacía **acepto**

Solución

- ▶ Estados q_0, q_1
- ▶ Estado inicial q_0
- ▶ Estado final q_1
- ▶ Alfabeto de entrada $\Sigma = \{a, b\}$
- ▶ Alfabeto de pila $\{a, b, \$\}$
- ▶ Transiciones:

e.	a			b			ϵ
c.	a	b	$\$$	a	b	$\$$	$\$$
q_0	(q_0, aa)	(q_0, ϵ)	$(q_0, a\$)$	(q_0, ϵ)	(q_0, bb)	$(q_0, b\$)$	$(q_1, \$)$

¿Cómo definimos un AdP que acepte ...

... el lenguaje $\{ ww^R \mid w \in \{a, b\}^* \}$

- ▶ La pila apila la **primera parte** de la palabra
- ▶ Usamos **no determinismo** para empezar la segunda parte
- ▶ En la segunda parte si en la cima hay una a y leo una a **desapilo**
- ▶ En la segunda parte si en la cima hay una b y leo una b **desapilo**
- ▶ Si termino la entrada con la pila vacía **acepto**

Solución

- ▶ Estados q_0, q_1, q_2
- ▶ Estado inicial q_0
- ▶ Estado final q_2
- ▶ Alfabeto de entrada $\Sigma = \{a, b\}$
- ▶ Alfabeto de pila $\{a, b, \$\}$
- ▶ Transiciones:

entrada	a			b		
cima	a	b	$\$$	a	b	$\$$
q_0	(q_0, aa)	(q_0, ab)	$(q_0, a\$)$	(q_0, ba)	(q_0, bb)	$(q_0, b\$)$
q_1	(q_1, ϵ)				(q_1, ϵ)	

entrada	ϵ		
cima	a	b	$\$$
q_0	(q_1, a)	(q_1, b)	$(q_1, \$)$
q_1			$(q_2, \$)$

Autómatas de pila deterministas

- ▶ Los autómatas de pila DETERMINISTAS no son equivalentes a los AdP
- ▶ Por ejemplo el lenguaje $A = \{a^i b^j c^k \mid i \neq j \text{ ó } j \neq k\}$ es reconocido por un AdP pero no por un autómata de pila determinista
- ▶ Es muy diferente de los autómatas finitos ...

Contenido de este tema

1. Introducción a los autómatas de pila
2. Definiciones
3. **Equivalencia con gramáticas**
4. Propiedades de clausura de los incontextuales
5. Lema de bombeo: lenguajes no incontextuales

Equivalencia de AdP y gramáticas

- ▶ Para cada gramática incontextual G existe un AdP M con $L(G) = L(M)$ Para cada gramática existe un AdP que reconoce el lenguaje generado por la gramática
- ▶ Para cada AdP M existe una gramática incontextual G con $L(M) = L(G)$ Para cada AdP existe una gramática que genera el lenguaje aceptado por el AdP

Convertir una gramática en AdP

- ▶ Definimos un AdP que tenga como alfabeto de pila la unión de terminales y no terminales, en la pila al principio colocamos S
- ▶ Si tenemos una regla en la gramática que diga $A \rightarrow \alpha$ en el autómatas cuando la cima sea A se desapila A y se apila α
- ▶ Si la cima de la pila coincide con el símbolo de la entrada se desapila
- ▶ Acepta si la pila queda vacía

Convertir una gramática en AdP: ejemplo

$$S \rightarrow aSb \mid ab$$

entrada	a	b	ϵ	
cima	a	b	S	$\$$
q_0				$(q_1, S\$)$
q_1	(q_1, ϵ)	(q_1, ϵ)	$(q_1, aSb)(q_1, ab)$	$(q_2, \$)$

Estado final q_2

Equivalencia de AdP y gramáticas: idea de la demostración

Convertir un AdP en gramática

- ▶ Definimos una gramática que tenga un noterminal $A_{p,q}$ para cada par de estados del autómata $p, q \in Q$
- ▶ $A_{p,q}$ genera todas las entradas que llevan del estado p con pila \$ al estado q con pila \$
- ▶ Se añaden las reglas:

- ▶ Si los finales son $F = \{q_1, \dots, q_r\}$,

$$S \rightarrow A_{q_0, q_1} \mid \dots \mid A_{q_0, q_r}$$

- ▶ Se añade la regla

$$A_{p,q} \rightarrow aA_{r,s}b$$

cuando la entrada a lleva de p a r apilando u y la entrada b lleva de s a q desapilando u

- ▶ Para todos los estados p, q, r , se añade la regla

$$A_{p,q} \rightarrow A_{p,r}A_{r,q}$$

- ▶ para cualquier estado p se añade la regla

$$A_{p,p} \rightarrow \epsilon$$

Convertir un AdP en gramática: Ejemplo

► Estado final q_1

e.	a			b			ϵ
c.	a	b	$\$$	a	b	$\$$	$\$$
q_0	(q_0, aa)	(q_0, ϵ)	$(q_0, a\$)$	(q_0, ϵ)	(q_0, bb)	$(q_0, b\$)$	$(q_1, \$)$

$$S \rightarrow A_{q_0, q_1}$$

$$A_{q_0, q_0} \rightarrow aA_{q_0, q_0}b$$

$$A_{q_0, q_0} \rightarrow bA_{q_0, q_0}a$$

$$A_{q_0, q_0} \rightarrow A_{q_0, q_0}A_{q_0, q_0} \mid A_{q_0, q_1}A_{q_1, q_0}$$

$$A_{q_0, q_1} \rightarrow A_{q_0, q_0}A_{q_0, q_1} \mid A_{q_0, q_1}A_{q_1, q_1}$$

$$A_{q_1, q_0} \rightarrow A_{q_1, q_0}A_{q_0, q_0} \mid A_{q_1, q_1}A_{q_1, q_0}$$

$$A_{q_1, q_1} \rightarrow A_{q_1, q_0}A_{q_0, q_1} \mid A_{q_1, q_1}A_{q_1, q_1}$$

$$A_{q_0, q_0} \rightarrow \epsilon$$

$$A_{q_1, q_1} \rightarrow \epsilon$$

Equivalencia de AdP y gramáticas: uso práctico

- ▶ Hay lenguajes para los que es más sencillo diseñar un AdP:
 $\{w \in \{a, b\}^* \mid |w|_a = |w|_b\}$
- ▶ Hay lenguajes para los que es más sencillo diseñar una gramática: $\{ww^R \mid w \in \{a, b\}^*\}$
- ▶ La semana que viene veremos para qué sirve conocer la gramática que genera un lenguaje y cómo genera cada palabra
- ▶ Por ejemplo esto último es muy útil para **compiladores** de lenguajes de programación

Contenido de este tema

1. Introducción a los autómatas de pila
2. Definiciones
3. Equivalencia con gramáticas
4. **Propiedades de clausura de los incontextuales**
5. Lema de bombeo: lenguajes no incontextuales

Propiedades de clausura de los incontextuales

- ▶ Los incontextuales son cerrados por unión, concatenación y estrella de Kleene
- ▶ Los incontextuales NO son cerrados por intersección y complemento
- ▶ La intersección de regular e incontextual es incontextual

Los incontextuales son cerrados por unión

- ▶ Si tenemos una gramática G_1 con símbolo inicial S_1
- ▶ Si tenemos una gramática G_2 con símbolo inicial S_2
- ▶ La gramática G con un nuevo símbolo inicial S , las reglas:

$$S \rightarrow S_1 \mid S_2$$

y todas las reglas de G_1 y G_2 cumple

$$L(G) = L(G_1) \cup L(G_2)$$

Los incontextuales son cerrados por unión: ejemplo

- ▶ $\{0^n 1^m \mid m = n \text{ ó } m = 2n\} =$
 $\{0^n 1^m \mid m = n\} \cup \{0^n 1^m \mid m = 2n\}$
- ▶ G_1 genera $\{0^n 1^m \mid m = n\}$:

$$S_1 \rightarrow 0S_11 \mid \epsilon$$

- ▶ G_2 genera $\{0^n 1^m \mid m = 2n\}$:

$$S_2 \rightarrow 0S_211 \mid \epsilon$$

- ▶ La gramática G con un nuevo símbolo inicial S , las reglas:

$$S \rightarrow S_1 \mid S_2$$

y todas las reglas de G_1 y G_2 cumple

$$L(G) = L(G_1) \cup L(G_2)$$

Los incontextuales son cerrados por concatenación

- ▶ Si tenemos una gramática G_1 con símbolo inicial S_1
- ▶ Si tenemos una gramática G_2 con símbolo inicial S_2
- ▶ La gramática G con un nuevo símbolo inicial S , las reglas:

$$S \rightarrow S_1 S_2$$

y todas las reglas de G_1 y G_2 cumple

$$L(G) = L(G_1) \cdot L(G_2)$$

Los incontextuales son cerrados por concatenación: ejemplo

- ▶ $\{0^n 1^n 2^m 1^m \mid n, m \in \mathbb{N}\} = \{0^n 1^n \mid n \in \mathbb{N}\} \cdot \{2^m 1^m \mid m \in \mathbb{N}\}$
- ▶ G_1 genera $\{0^n 1^n \mid n \in \mathbb{N}\}$:

$$S_1 \rightarrow 0S_11 \mid \epsilon$$

- ▶ G_2 genera $\{2^m 1^m \mid m \in \mathbb{N}\}$:

$$S_2 \rightarrow 2S_21 \mid \epsilon$$

- ▶ La gramática G con un nuevo símbolo inicial S , las reglas:

$$S \rightarrow S_1 S_2$$

y todas las reglas de G_1 y G_2 cumple

$$L(G) = L(G_1) \cdot L(G_2)$$

Los incontextuales son cerrados por estrella de Kleene

- ▶ Si tenemos una gramática G_1 con símbolo inicial S_1
- ▶ La gramática G con un nuevo símbolo inicial S , las reglas:

$$S \rightarrow SS_1 \mid \epsilon$$

y todas las reglas de G_1 cumple

$$L(G) = L(G_1)^*$$

Los incontextuales son cerrados por estrella de Kleene: ejemplo

- ▶ $\{0^n 1^n \mid n\}^*$
- ▶ G_1 genera $\{0^n 1^n \mid n \in \mathbb{N}\}$:

$$S_1 \rightarrow 0S_11 \mid \epsilon$$

- ▶ La gramática G con un nuevo símbolo inicial S , las reglas:

$$S \rightarrow SS_1 \mid \epsilon$$

y todas las reglas de G_1 cumple

$$L(G) = L(G_1)^*$$

Además ...

- ▶ Los incontextuales NO son cerrados por intersección y complemento (lo veremos)
- ▶ La intersección de regular e incontextual es incontextual (no veremos la demostración, es parecida a la intersección de regulares)
- ▶ La segunda afirmación es útil para ver que un lenguaje no es incontextual

Contenido de este tema

1. Introducción a los autómatas de pila
2. Definiciones
3. Equivalencia con gramáticas
4. Propiedades de clausura de los incontextuales
5. **Lema de bombeo: lenguajes no incontextuales**

Lema de bombeo para incontextuales

- ▶ Se trata de un lema que nos permite demostrar que algunos lenguajes no son incontextuales ($\{a^n b^n c^n \mid n \in \mathbb{N}\}$)
- ▶ Está basado en los bucles de las gramáticas, por ejemplo $S \rightarrow aSb$ o bien $S \rightarrow Sb$
- ▶ Si tienes una derivación de la palabra

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaaabbbb$$

puedes repetir el bucle $S \rightarrow aSb$ y generar palabras más largas usando $S \Rightarrow^* a^n Sb^n$

- ▶ Lo mismo si tienes reglas $A \rightarrow aBa$ y $B \rightarrow bA$

$$S \Rightarrow AB \Rightarrow aBaB \Rightarrow abAaB$$

puedes repetir el bucle $A \Rightarrow^* abAa$ y generar palabras más largas usando $A \Rightarrow^* (ab)^n Aa^n$

Lema de bombeo para incontextuales

Lema de bombeo

- ▶ Dado un lenguaje infinito A
- ▶ si $\forall N \exists w$ con $w \in A$, $|w| \geq N$
 - ▶ tal que $\forall u, v, x, y, z$ con $w = uvxyz$, $|vy| \geq 1$ y $|vxy| \leq N$
 - ▶ $\exists i$ con $uv^i xy^i z \notin A$
- ▶ entonces A no es incontextual

Intuitivamente

- ▶ Dado un lenguaje infinito A
- ▶ si existe una palabra $w \in A$ todo lo larga que quiera
- ▶ tal que para cualquier partición w no se puede bombear
- ▶ entonces A no es incontextual

Ejemplo, $A = \{a^n b^n c^n \mid n \in \mathbb{N}\}$

- ▶ Para todo N existe una palabra en A ,
 - ▶ $w = a^N b^N c^N \mid |w| = 3N \geq N$
- ▶ tal que para cualquier partición de w u, v, x, y, z con $w = uvxyz$, $|vy| \geq 1$ y $|vxy| \leq N$,
 - ▶ la partición puede ser

$$vxy = a^r b^s \quad v = a^t, y = a^l b^s, t + l + s \geq 1$$

$$vxy = a^r b^s \quad v = a^t b^l, y = b^j, t + l + j \geq 1$$

$$vxy = b^r c^s \quad v = b^t, y = b^l c^s, t + l + s \geq 1$$

$$vxy = b^r c^s \quad v = b^t c^l, y = c^j, t + l + j \geq 1$$

- ▶ $\exists i$ con $uv^i xy^i z \notin A$
 - ▶ caso $vxy = a^r b^s, v = a^t, y = a^l b^s, t + l + s \geq 1$, para $i = 2$,

$$uv^2 xy^2 z = a^{N-r} a^{2t} a^{r-t-l} a^l b^s a^l b^s b^{N-s} c^N$$

si $s \geq 1$ y si $s = 0$ se puede ver que $uv^2 xy^2 z \notin A$

- ▶ Hay que mirar los otros tres casos
- ▶ luego A no es incontextual

Lema de bombeo para incontextuales

- ▶ Se parece al lema de bombeo para regulares
- ▶ La principal **diferencia** es que se bombean dos trozos a la vez
- ▶ Hay que considerar las **particiones en 5 trozos** con $|vy| \geq 1$ y $|vxy| \leq N$
- ▶ En general al ser 5 trozos suelen salir más casos a tener en cuenta

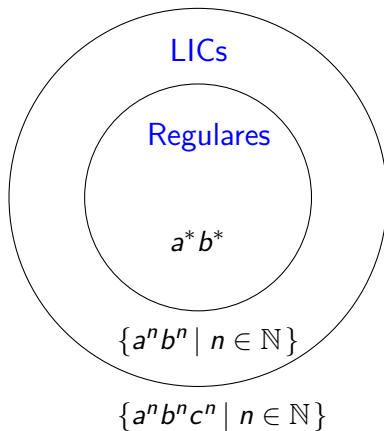
Otras formas de ver que un lenguaje no es incontextual

Usando las propiedades de clausura:

- ▶ Si A y B son incontextuales entonces $A \cup B$ es incontextual.
Si $A \cup B$ no es incontextual y B es incontextual entonces A no es incontextual.
- ▶ Si A es incontextual y B es regular entonces $A \cap B$ es incontextual.
Si $A \cap B$ no es incontextual y B es regular entonces A no es incontextual.

Lenguaje incontextual = Leng. independiente del contexto = LIC

Regulares e incontextuales



Bibliografía

- ▶ Sipser (2a edición), secciones 2.2. y 2.3.
- ▶ Kelley, secciones 3.6 a 3.8.