Si las longitudes de las cadenas de un lenguaje $L$ están acotadas superiormente, por ejemplo: $x \in L \Rightarrow |x| \leq 20$, entonces $L$ tiene que ser finito.

Qué condición tiene que cumplir el grafo de un autómata finito para que el lenguaje que éste acepta sea infinito?

Debe existir un camino desde el estado inicial hasta algún estado final que pase por algún ciclo.

Si las longitudes de las cadenas de un lenguaje $L$ están acotadas superiormente, por ejemplo: $x \in L \Rightarrow |x| \leq 20$, entonces $L$ tiene que ser finito.

Qué condición tiene que cumplir el grafo de un autómata finito para que el lenguaje que éste acepta sea infinito?

Debe existir un camino desde el estado inicial hasta algún estado final que pase por algún ciclo.

Si las longitudes de las cadenas de un lenguaje $L$ están acotadas superiormente, por ejemplo: $x \in L \Rightarrow |x| \leq 20$, entonces $L$ tiene que ser finito.

Qué condición tiene que cumplir el grafo de un autómata finito para que el lenguaje que éste acepta sea infinito?

Debe existir un camino desde el estado inicial hasta algún estado final que pase por algún ciclo.

Si las longitudes de las cadenas de un lenguaje $L$ están acotadas superiormente, por ejemplo: $x \in L \Rightarrow |x| \leq 20$, entonces $L$ tiene que ser finito.

Qué condición tiene que cumplir el grafo de un autómata finito para que el lenguaje que éste acepta sea infinito?

Debe existir un camino desde el estado inicial hasta algún estado final que pase por algún ciclo.

Si las longitudes de las cadenas de un lenguaje $L$ están acotadas superiormente, por ejemplo: $x \in L \Rightarrow |x| \leq 20$, entonces $L$ tiene que ser finito.

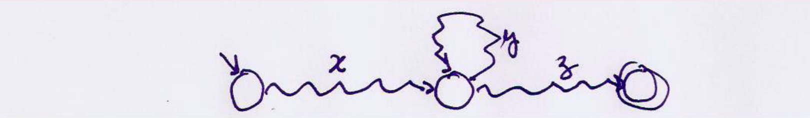
Qué condición tiene que cumplir el grafo de un autómata finito para que el lenguaje que éste acepta sea infinito?

Debe existir un camino desde el estado inicial hasta algún estado final que pase por algún ciclo.

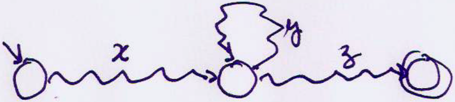
Si las longitudes de las cadenas de un lenguaje $L$ están acotadas superiormente, por ejemplo: $x \in L \Rightarrow |x| \leq 20$, entonces $L$ tiene que ser finito.

Qué condición tiene que cumplir el grafo de un autómata finito para que el lenguaje que éste acepta sea infinito?

Debe existir un camino desde el estado inicial hasta algún estado final que pase por algún ciclo.



## Lema de Pumping para lenguajes regulares




$$L \text{ regular} \Rightarrow \exists n, \forall w, \ (w \in L \wedge |w| \geq n) \Rightarrow \exists x, y, z: \begin{array}{ll} w = xyz & \wedge \\ |xy| \leq n & \wedge \\ |y| \geq 1 & \wedge \\ \forall i \geq 0, \ xy^iz \in L & \end{array}$$

# Lema de Pumping para lenguajes regulares

$$\underbrace{\forall n, \exists w, \left( (w \in L \wedge |w| \geq n) \wedge \left( \forall x, y, z : \begin{array}{l} w \neq xyz \quad \vee \\ |xy| > n \quad \vee \\ |y| < 1 \quad \vee \\ \exists i \geq 0,\ xy^i z \notin L \end{array} \right) \right)}$$

$$\Downarrow$$

$L$ no regular

Ejemplo: lenguaje regular, cumple pumping

$L = \{ 0 1^k : k \geq 1 \}$

$n = 2$

p.ej. $w = \underbrace{0}_{x}\,\underbrace{1}_{y}\,\underbrace{1\ 1\ 1}_{z}$

$w = xyz$

$|xy| = |01| = 2 \leq n = 2$

$|y| = |1| \geq 1$

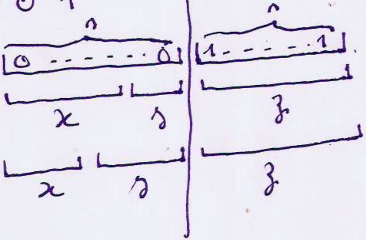
$\forall i \geq 0,\ x y^i z = 0 1^i 111$

Ejemplo: lenguaje que no cumple Pumping y que por lo tanto no es regular.

$L = \{0^k 1^k : k \geq 1\}$

tomemos un n "genérico", ya que tenemos que probar "para todo n" ($\forall n$)

$w = 0^n 1^n$




- $w = xyz$
- $|xy| \leq n$
- $|y| \geq 1$
- pero vemos que $\exists i \geq 0$, $xy^iz \notin L$ por ejemplo: cualquier $i \neq 1$.

**Ejemplo:** el lenguaje $L = \{1^p : p \text{ es primo}\}$ **no es regular**.

Tomemos un $n$ "genérico" y la cadena $1^p$ con $p \geq n+2$.

Si $L$ fuera regular, debería cumplir Pumping, o sea $1^p$ debería poseer una ~~cadena~~ descomposición "admisible":

Supongamos que: $|y| = m \geq 1$, $|xz| = p - m$, $|xy| \leq n$

$$\overbrace{x \quad \underbrace{y}_{m} \quad z}^{p}$$

Si $L$ cumpliera con Pumping, debería ser cierto que

$\forall i \geq 0,\ xy^iz \in L$ o sea

$\forall i \geq 0,\ (p-m)+im$ deberían ser **primos** $\forall i$

en particular para $i = p - m$. Entonces debería ser

$$(p-m) + (p-m)m \quad \text{primo}$$

pero $(p-m) + (p-m)m = (p-m)(1+m)$.

Pero $\boxed{p - m \geq 2}$ porque: $-\ p \geq n+2$ y

$-\ m \leq |y| \leq |xy| \leq n$

Por otro lado $\boxed{m + 1 \geq 2}$ porque: $y \neq \lambda \Rightarrow m \geq 1$

De esto surge que

$(p-m)(1+m)$ no es primo.

Por lo tanto, $\exists i$ $(i = p-m)$ que hace que

$\forall i \geq 0,\ xy^iz \in L$ sea FALSO.