

Lenguajes Formales y Autómatas

Características de la cursada

Teóricas: Miércoles de 18:30 a 20:30 hs (Fede) Aula: 249

Prácticas: Lunes de 19 a 21 hs (Ema) Aula: 255

Tenemos un Discord :) !

Más burocracia

- Asistencia y entregas semanales (son importanchis, tienen nota :D)
- Evaluaciones (Estas también son importantes)
- Medios de comunicación y consultas

Bueno, al final todo era importante...

Cómo ~~sobrevivir~~ aprobar esta cursada

Habr  dos trabajos con un recuperatorio

Promoci n: Entregar el **75%** de los ejercicios semanales y la obtenci n de un promedio m nimo de **7** (siete) puntos en las instancias parciales de evaluaci n y un m nimo de **6** (seis) puntos en cada una de ellas.

Entregar el **75%** (setenta y cinco por ciento) de los ejercicios semanales y la obtenci n de un m nimo de **4** (cuatro) puntos en cada instancia parcial de evaluaci n. Debe rendir examen final.

Las entregas constituyen un 10% de la nota final :)

Bibliografía de la materia

- Principal/Obligatoria
 - Michael Sipser. Introduction To The Theory Of Computation
- Auxiliar/Opcional
 - John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman. Introduction to automata theory, languages, and computation.
 - John Carroll and Darrell Long. Theory of Finite Automata with an Introduction to Formal Languages.

Calendario tentativo de la cursada (desactualizado)

Miercoles	15/08 Teórica	1 Alfabeto, lenguajes cadenasalfabeto
Lunes	21/08 Feriado	
Miercoles	23/08 Teórica	2 Qué es un autómata, ejemplos
Lunes	28/08 Práctica	Practica lenguajes, cadenas, alfabeto
Miercoles	30/08 Teórica	3 Formalismo AFD
Lunes	04/09 Práctica	Práctica AFD
Miercoles	06/09 Teórica	4 Minimización/traductores
Lunes	11/09 Práctica	Practica Minimizacion
Miercoles	13/09 Teórica	5 AFND
Lunes	18/09 Práctica	Practica AFND
Miercoles	20/09 Teórica	6 Propiedades Lenguajes Regulares
Lunes	25/09 Práctica	Practica propiedades
Miercoles	27/09 Teórica	ER
Lunes	02/10 Práctica	Practica ER
Miercoles	04/10 Teórica	Pumping Lema
Lunes	09/10 Práctica	Practica Pumping Lema
Miercoles	11/10 Teórica	Repaso Envío de Parcial Domiciliario

Lunes	16/10 Feriado	
Miercoles	18/10 Teórica	Gramaticas regulares
Lunes	23/10 Práctica	Practica GR
Miercoles	25/10 Teórica	Gramaticas GLC
Lunes	30/10 Práctica	Practica GLC
Miercoles	01/11 Teórica	AP
Lunes	06/11 Práctica	AP
Miercoles	08/11 Teórica	Equivalencia GLC AP
Lunes	13/11 Práctica	AP
Miercoles	15/11 Teórica	Turing
Lunes	20/11 Feriado	
Miercoles	22/11 Teórica	Practica Turing
Lunes	27/11 Práctica	Repaso Enviar 2do TP
Miercoles	29/11 Teórica	Consulta/TP
Lunes	04/12 Práctica	Rec 1 Ejercicio en pizarrón
Miercoles	06/12 Teórica	Rec 2 Ejercicio en pizarrón
Lunes	11/12 Práctica	Coloquio (menos de 4)
Miercoles	13/12 Teórica	

Bueno, ahora si :D

Lenguajes ? Formales ? Autómatas, What ?

Nuestro objetivo en la cursada va a ser estudiar ciertos **sistemas simbólicos**

- Conjuntos de secuencias de símbolos (strings/palabras/fórmulas)
- Maneras de definirlos, construirlos, procesarlos
- Reconocer si están bien formados

¿Por qué nos interesan estos sistemas simbólicos?

Nos interesa resolver problemas con computadoras

Programas y computadoras digitales: Manipuladores abstractos y concretos de símbolos (palabras de lenguajes de alto y bajo nivel)

¿Y por qué nos interesan estos símbolos?

Información digital: codificar y decodificar datos en símbolos (y al final en 0s y 1s)

Lenguajes: Conjuntos de cadenas/strings de símbolos

Entonces qué son los lenguajes?

Lenguajes: Se conforma de una **sintaxis** y una **semántica**

Lenguajes “naturales”: producto del desarrollo histórico y social

Lenguajes formales: Sintaxis y significado definidos de antemano, antes de usarlo.

Tipos de lenguajes formales

Lenguajes regulares → Autómatas finitos

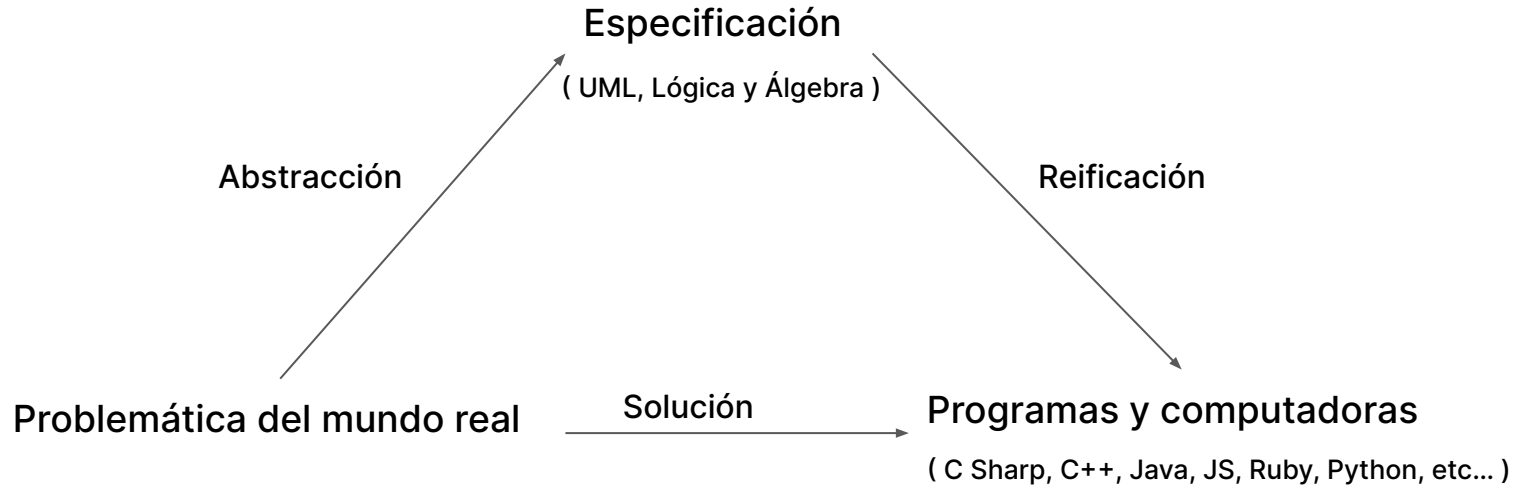
Lenguajes independientes del contexto → Autómatas de pila

Lenguajes sin restricciones → Máquinas de Turing

Por qué los queremos ?

- Queremos resolver problemas con computadoras (digitales)
- Donde tenemos entradas y salidas (cadenas de símbolos)
- Y un programa son... cadenas de símbolos que describen algoritmos para resolver problemas

Caso general



Algunas definiciones iniciales :)

- Alfabeto (Σ): un conjunto finito de símbolos
Ex: { a, b, c, ..., z }
- String/Cadena: secuencia de símbolos de algún alfabeto
Ex: "gato", "perro", "casa"
- Lenguaje: un conjunto de cadenas
Ex: { "gato", "perro", "casa" }

Veamos algunos ejemplos

- Σ (Alfabeto) = $\{ a, b \}$ $u = ab, v = bbbaaa, w = abba$
- $\Sigma = \{ 1, 2, 3, \dots, 9 \}$ $u = 123, v = 42, w = 21212121\dots$

Para un alfabeto puede haber un número arbitrario de lenguajes, por ejemplo para $\Sigma = \{ 1, 2, 3, \dots, 9 \}$

Primos = $\{ 2, 3, 5, 7, 11, 13, 17, \dots \}$ Pares = $\{ 2, 4, 6, 8, \dots \}$

Los lenguajes se usan para estructurar información y describir problemas

Operaciones con cadenas/strings

$$w = a_1 a_2 \cdots a_n$$

$$v = b_1 b_2 \cdots b_m$$

Concatenación

$$wv = a_1 a_2 \cdots a_n b_1 b_2 \cdots b_m$$

Reversa

$$w^R = a_n \cdots a_2 a_1$$

Longitud

$$|w| = n \qquad |uv| = |u| + |v|$$

Ocurrencias

$$|w|_b = \text{cantidad de } b \text{ en } w, \quad b \in \Sigma \text{ y } w \text{ string sobre } \Sigma$$

Observaciones

- Un string sin símbolos es denotado λ o ϵ (cadena vacía)
- Un substring de un string es una subsecuencia de caracteres consecutivos
- Sea la cadena $w = uv$, decimos que u es *prefijo* y v es *sufijo* de w

$$|\lambda| = 0 \quad |\lambda|_a = 0$$

$$\lambda w = w \lambda = w$$

$$\lambda abba = abba \lambda = ab \lambda ba = abba$$

<i>abbab</i>	<i>ab</i>
<i>abbab</i>	<i>abba</i>
<i>abbab</i>	<i>b</i>
<i>abbab</i>	<i>bbab</i>

<i>abbab</i>	λ	<i>abbab</i>
	<i>a</i>	<i>bbab</i>
	<i>ab</i>	<i>bab</i>
	<i>abb</i>	<i>ab</i>
	<i>abba</i>	<i>b</i>
	<i>abbab</i>	λ

Más operaciones sobre cadenas

$$w = a_1 a_2 \cdots a_n$$

$$v = b_1 b_2 \cdots b_m$$

Potencia

$$w^n = \underbrace{ww \cdots w}_n \quad w^0 = \lambda$$

Operación $*$ \rightarrow el conjunto de todos los posibles strings del alfabeto Σ

$$\Sigma = \{a, b\}$$

$$\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, aab, \dots\}$$

Operación $+$ \rightarrow el conjunto de todos los posibles strings del alfabeto Σ excepto λ

$$\Sigma^+ = \Sigma^* - \lambda$$

$$\Sigma^+ = \{a, b, aa, ab, ba, bb, aaa, aab, \dots\}$$

Hablemos de lenguajes

Un lenguaje sobre el alfabeto Σ es cualquier subconjunto de Σ^*

$$\Sigma = \{a, b\}$$

$$\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, \dots\}$$

Posibles lenguajes

$$\{\lambda\}$$

$$\{a, aa, aab\}$$

$$\{\lambda, abba, baba, aa, ab, aaaaaa\}$$

Lenguaje de números primos

Alfabeto

$$\Sigma = \{0,1,2,\dots,9\}$$

Lenguaje

$$PRIMOS = \{x : x \in \Sigma^* \text{ y } x \text{ es primo}\}$$

$$PRIMOS = \{2,3,5,7,11,13,17,\dots\}$$

Lenguaje de números pares e impares

Alfabeto

$$\Sigma = \{0,1,2,\dots,9\}$$

Lenguaje

$$PARES = \{x : x \in \Sigma^* \text{ y } x \text{ es par}\}$$

$$PARES = \{0,2,4,6,\dots,328,\dots\}$$

$$IMPARES = \{x : x \in \Sigma^* \text{ y } x \text{ es impar}\}$$

$$IMPARES = \{1,3,5,7,\dots,453,\dots\}$$

Lenguaje de suma unaria

Alfabeto

$$\Sigma = \{1, +, =\}$$

Lenguaje

$$ADICION = \{x + y = z : x = 1^n, y = 1^m, z = 1^k, \\ n + m = k\}$$

$$11 + 111 = 11111 \in ADICION$$

$$111 + 111 = 111 \notin ADICION$$

Observaciones

$$\emptyset = \{\} \neq \{\lambda\}$$

$$|\{\}| = |\emptyset| = 0$$

$$|\{\lambda\}| = 1$$

$$|\lambda| = 0$$

Operaciones sobre lenguajes

Operaciones de conjuntos

$$\{a, ab, aaaa\} \cup \{bb, ab\} = \{a, ab, bb, aaaa\}$$

$$\{a, ab, aaaa\} \cap \{bb, ab\} = \{ab\}$$

$$\{a, ab, aaaa\} - \{bb, ab\} = \{a, aaaa\}$$

$$\overline{L} = \Sigma^* - L \quad \Sigma = \{a, b\}$$

$$\overline{\{a, ba\}} = \{\lambda, b, aa, ab, bb, aaa, \dots\}$$

Operaciones sobre lenguajes

Reversa

$$L^R = \{w^R : w \in L\}$$

$$\{ab, aab, baba\}^R = \{ba, baa, abab\}$$

Concatenación

$$L_1 L_2 = \{xy : x \in L_1, y \in L_2\}$$

$$\{a, ab, ba\}\{b, aa\}$$

$$= \{ab, aaa, abb, abaa, bab, baaa\}$$

Potencia

$$L^n = \underbrace{LL \dots L}_n \quad L^0 = \{\lambda\}$$

$$\{a, bba, aaa\}^0 = \{\lambda\}$$

$$\{a, b\}^3 = \{a, b\}\{a, b\}\{a, b\} = \\ \{aaa, aab, aba, abb, baa, bab, bba, bbb\}$$

Clausura de Kleene ^{*} y ⁺

Todas las cadenas (strings) que pueden construirse a partir de L

$$L^* = L^0 \cup L^1 \cup L^2 \dots$$

$$\{a,bb\}^* = \left\{ \begin{array}{l} \lambda, \\ a,bb, \\ aa,abb,bba,bbbb, \\ aaa,aabb,abba,abbbb,\dots \end{array} \right\}$$

Todas las cadenas que pueden construirse a partir de L menos el string vacío

$$L^+ = L^* - \{\lambda\}$$
$$L^+ = L^1 \cup L^2 \cup \dots$$

$$\{a,bb\}^+ = \left\{ \begin{array}{l} a,bb, \\ aa,abb,bba,bbbb, \\ aaa,aabb,abba,abbbb,\dots \end{array} \right\}$$

Problema fundamental

$$\Sigma = \{a, b\}$$

$$L = \{a^n b^n : n \geq 0\}$$

$$\Sigma = \{0, 1, 2, \dots, 9\}$$

$$L = \textit{PARES} = \{x : x \in \Sigma^* \text{ y } x \text{ es par}\}$$

$$L = \textit{IMPARES} = \{x : x \in \Sigma^* \text{ y } x \text{ es impar}\}$$

Cómo podemos saber si algún $x \in L$?

Necesitamos métodos que solamente manipulando símbolos respondan la pregunta

Gramáticas

Las Gramáticas especifican lenguajes a través de reglas

$\langle \text{oración} \rangle \rightarrow \langle \text{sujeto} \rangle \langle \text{predicado} \rangle$

$\langle \text{sujeto} \rangle \rightarrow \langle \text{artículo} \rangle \langle \text{nombre} \rangle$

$\langle \text{predicado} \rangle \rightarrow \langle \text{verbo} \rangle$

$\langle \text{artículo} \rangle \rightarrow \text{un}$

$\langle \text{artículo} \rangle \rightarrow \text{el}$

$\langle \text{nombre} \rangle \rightarrow \text{gato}$

$\langle \text{nombre} \rangle \rightarrow \text{perro}$

$\langle \text{verbo} \rangle \rightarrow \text{corre}$

$\langle \text{verbo} \rangle \rightarrow \text{salta}$

una derivación de “el perro corre”

$\langle \text{oración} \rangle \Rightarrow \langle \text{sujeto} \rangle \langle \text{predicado} \rangle$

$\Rightarrow \langle \text{sujeto} \rangle \langle \text{verbo} \rangle$

$\Rightarrow \langle \text{artículo} \rangle \langle \text{nombre} \rangle \langle \text{verbo} \rangle$

$\Rightarrow \text{el} \langle \text{nombre} \rangle \langle \text{verbo} \rangle$

$\Rightarrow \text{el perro} \langle \text{verbo} \rangle$

$\Rightarrow \text{el perro corre}$

Cuál es el lenguaje que acepta la gramática ?

$L = \{$
 “un gato salta”, “un gato corre”, “el gato salta”,
 “el gato corre”, “un perro salta”, “un perro corre”,
 “el perro salta”, “el perro corre”
 $\}$

$\langle oración \rangle \rightarrow \langle sujeto \rangle \langle predicado \rangle$

$\langle sujeto \rangle \rightarrow \langle artículo \rangle \langle nombre \rangle$

$\langle predicado \rangle \rightarrow \langle verbo \rangle$

$\langle artículo \rangle \rightarrow un$

$\langle artículo \rangle \rightarrow el$

$\langle nombre \rangle \rightarrow gato$

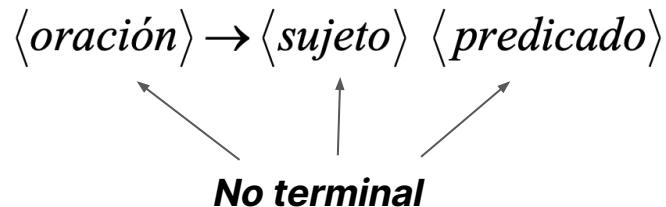
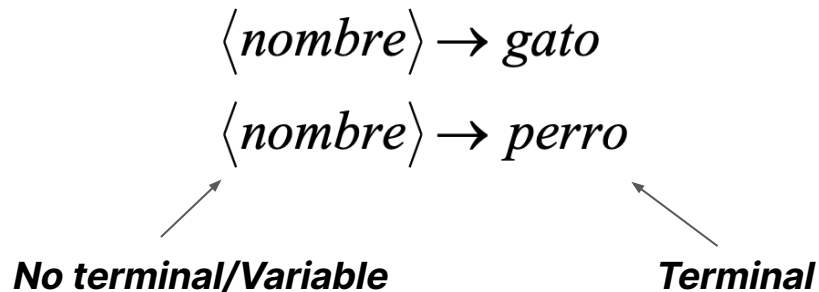
$\langle nombre \rangle \rightarrow perro$

$\langle verbo \rangle \rightarrow corre$

$\langle verbo \rangle \rightarrow salta$

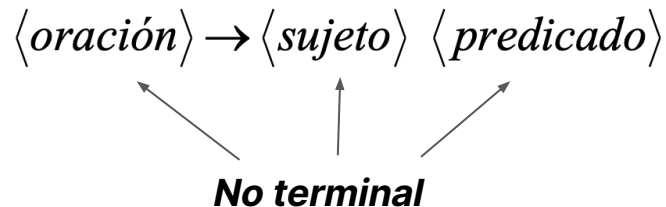
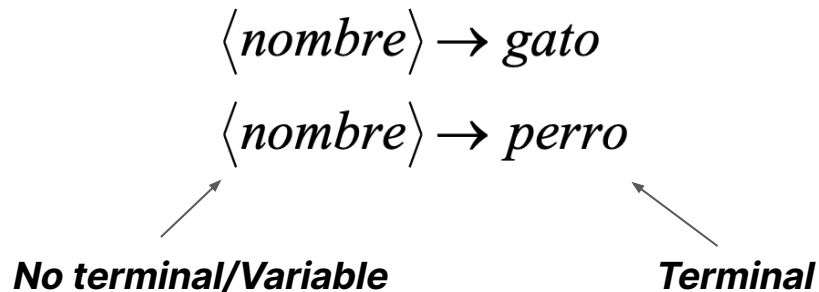
Notación

Este conjunto de definiciones se llaman *reglas de producción/reducción*



Notación

Este conjunto de definiciones se llaman *reglas de producción/reducción*



Ejemplo

Gramática

$$S \rightarrow aSb$$

$$S \rightarrow \lambda$$

Lenguaje

$$L = \{a^n b^n : n \geq 0\}$$

Derivación de la cadena "ab"

$$\begin{array}{ccc} & S \Rightarrow aSb \Rightarrow ab & (S \text{ reduce en dos pasos a } ab) \\ & \nearrow \quad \nwarrow & \\ S \rightarrow aSb & & S \rightarrow \lambda \end{array}$$

Derivación de la cadena "aabb"

$$\begin{array}{ccc} & S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb & \\ & \nearrow \quad \nwarrow \quad \nearrow & \\ S \rightarrow aSb & & S \rightarrow \lambda \end{array}$$

Cómo definimos formalmente una gramática ?

$$G = (V, \Sigma, S, P)$$

V: Conjunto de no terminales

Σ : Conjunto de terminales

S: No terminal/variable **inicial**

P: Conjunto de reglas de producción/reescritura

Qué gramática acepta el lenguaje L ?

$$L = \{a^n b^n : n \geq 0\}$$

Piensenlo 5 minutos :)

Qué gramática acepta el lenguaje L ?

$$L = \{a^n b^n : n \geq 0\}$$

Damos un $G = \langle V, \Sigma, S, P \rangle$

$$V: \{ S \} \quad \Sigma: \{ a, b \} \quad S: S \quad P: \{ S \rightarrow aSb, S \rightarrow \lambda \}$$

Detalles...

- Un string que contiene No terminales/variables y terminales lo llamamos **Forma sentencial**
- Un string que contiene sólo terminales lo llamamos **Sentencia**

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaabbb$$

- Escribimos $S \xRightarrow{*} aaabbb$ Para denotar $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aaaSbbb \Rightarrow aaabbb$

Pequeño ejercicio

Sea $G = \langle V, \Sigma, S, P \rangle$

Qué forma tienen V, Σ, S, P ?

$$\begin{array}{l} S \rightarrow Ab \\ A \rightarrow aAb \\ A \rightarrow \lambda \end{array}$$

Qué lenguaje acepta ?

Lenguaje generado

Dada una gramática G con símbolo inicial S , definimos $L(G)$ como:

$$L(G) = \{w \in \Sigma^* : S \xRightarrow{*} w\}$$

Nota: Usamos una convención notacional para $A \rightarrow aAb$, $A \rightarrow \lambda$ lo escribimos $A \rightarrow aAb \mid \lambda$

Ejercicio

Dar una reducción de G

Sea G :

$S \rightarrow aSBC \mid \lambda$ $CB \rightarrow BC$

$aB \rightarrow ab$ $bB \rightarrow bb$

$bC \rightarrow bc$ $cC \rightarrow cc$

Clasificación de gramáticas

TIPO	GRAMATICA
0	IRRESTRICTA $\alpha \rightarrow \beta$ $\alpha, \beta \in (V \cup \Sigma)^*$
1	SENSIBLE AL CONTEXTO $\alpha A \beta \rightarrow \alpha \gamma \beta$ $\alpha, \beta \in (V \cup \Sigma)^*$ $\gamma \in (V \cup \Sigma)^+, A \in V$ $\alpha \rightarrow \beta$ $ \alpha \leq \beta $
2	LIBRE DE CONTEXTO $A \rightarrow \alpha$ $\alpha \in (V \cup \Sigma)^*$ $A \in V$
3	LINEAL DERECHA O IZQUIERDA $A \rightarrow \alpha B$ $\alpha \in \Sigma^*$ $A \in V$ $A \rightarrow B \alpha$ $B \in (V \cup [\lambda])$

Jerarquía de Chomsky

TIPO	GRAMATICA	AUTOMATA	LENGUAJE
0	IRRESTRICTA $\alpha \rightarrow \beta \quad \alpha, \beta \in (V \cup \Sigma)^*$	MAQUINA DE TURING	TURING RECONOCIBLE
1	SENSIBLE AL CONTEXTO $\alpha A \beta \rightarrow \alpha \gamma \beta \quad \alpha, \beta \in (V \cup \Sigma)^*$ $\gamma \in (V \cup \Sigma)^+, A \in V$ $\alpha \rightarrow \beta$	AUTOMATA LINEALMENTE LIMITADO	SENSIBLE AL CONTEXTO
2	LIBRE DE CONTEXTO $A \rightarrow \alpha \quad \alpha \in (V \cup \Sigma)^*$ $A \in V$	AUTOMATA DE PILA	LIBRE DE CONTEXTO
3	LINEAL DERECHA O IZQUIERDA $A \rightarrow \alpha B \quad \alpha \in \Sigma^*$ $A \in V$ $A \rightarrow B \alpha \quad B \in (V \cup \{\lambda\})$	AUTOMATA FINITO	REGULAR