Herramientas computacionales para el procesamiento automático de textos

Profesora Titular
Julia Milanese
Colaboradores
Federico Alvarez
Catalina Rubio
Victoria Colombo
Colaboradores invitados
Martín Kondratzky, Fernando Carranza, Macarena Fernández Urquiza, Fernando Schiaffino

Contenidos

Unidad I: Nociones básicas

- Jerarquía de Chomsky.
- Costo computacional: tiempo lineal, polinómico y exponencial; nociones básicas de tratabilidad y decidibilidad.
- Gramáticas, autómatas y lenguajes formales.
- Introducción a las expresiones regulares.

Contenidos

Unidad I: Nociones básicas

- Jerarquía de Chomsky.
- Costo computacional: tiempo lineal, polinómico y exponencial; nociones básicas de tratabilidad y decidibilidad.
- Gramáticas, autómatas y lenguajes formales.
- Introducción a las expresiones regulares.

Bibliografía:

- Partee, B., T. Meulen, y R. Wall (1993). Capítulo 16: "Basic Concepts". *Mathematical Methods in Linguistics*. Dordrecht: Kluwer Academic Publishers. pp 433-454.
- Moreno Sandoval, A. (2001). Apéndice 2: "Nociones de Lingüística Matemática". *Gramáticas de Unificación y rasgos*. Madrid: Antonio Machado. pp. 227-256

Alfabeto

- Un alfabeto es un conjunto no vacío de símbolos.
  Por ejemplo: ALFABETO-LATINO $= \{a, b, c, d...\}$,
  NÚMEROS-NATURALES$=\{1, 2, 3, 4...\}$

Alfabeto

- Un alfabeto es un conjunto no vacío de símbolos.
  Por ejemplo: ALFABETO-LATINO $= \{a, b, c, d...\}$,
  NÚMEROS-NATURALES$=\{1, 2, 3, 4...\}$
- Convencionalmente, el alfabeto se nombra con el símbolo Σ

Cadenas

- Todas las cadenas de determinada longitud k que se pueden construir con un alfabeto Σ se representan convencionalmente Σ^k

  Por ejemplo, dado el alfabeto $\Sigma = \{a, b\}$, se dan las siguientes extensiones:

  $\Sigma^0 = \{\emptyset\}$
  $\Sigma^1 = \{a, b\}$
  $\Sigma^2 = \{aa, ab, ba, bb\}$
  $\Sigma^3 = \{aaa, aab, abb, aba, bbb, bba, baa, bab\}...$

Cadenas

- Todas las cadenas de determinada longitud k que se pueden construir con un alfabeto Σ se representan convencionalmente Σ^k

  Por ejemplo, dado el alfabeto $\Sigma = \{a, b\}$, se dan las siguientes extensiones:
  $\Sigma^0 = \{\emptyset\}$
  $\Sigma^1 = \{a, b\}$
  $\Sigma^2 = \{aa, ab, ba, bb\}$
  $\Sigma^3 = \{aaa, aab, abb, aba, bbb, bba, baa, bab\}...$

- Para representar el conjunto de todas las cadenas posibles que se pueden obtener a partir de un alfabeto Σ se usa la notación Σ^*.

  En términos de teoría de conjuntos,
  $\Sigma^* = \{\Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3 \cup \Sigma^4 \cup ...\}$

Los lenguajes

"A set of strings of which are chosen from Σ, where Σ is a particular alphabet, is called *a language*"

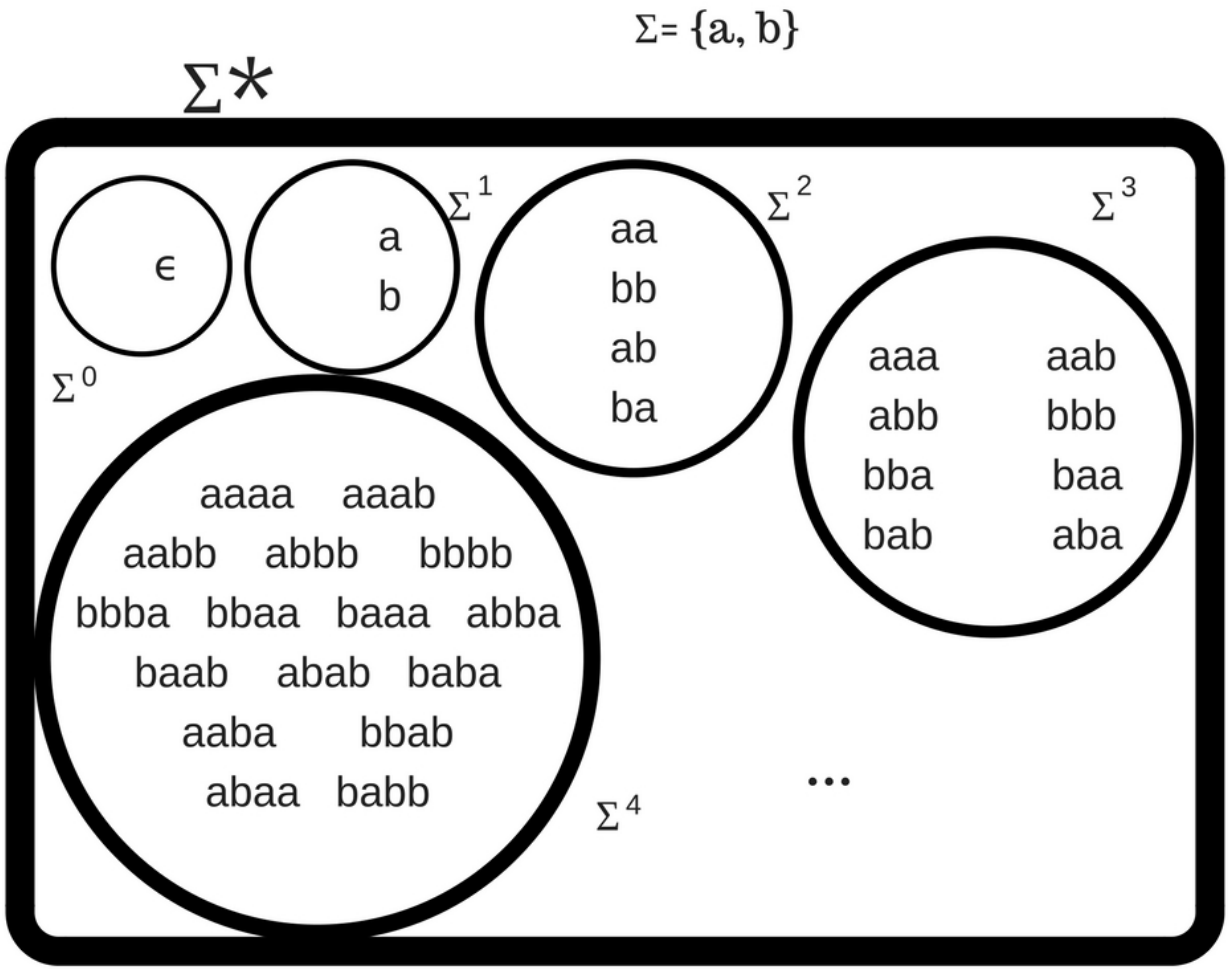


Figura: Imagen de ejemplo de las cadenas posibles generadas por un alfabeto Σ

Los lenguajes

¿Qué es el lenguaje?

Los lenguajes

¿Qué es el lenguaje?
At one level of description, **a natural language is simply a set of strings** –finite sequences of words, morpheme, phonemes, or whatever. Not every possible sequence is in the language: we distinguish the *grammatical* strings from those that are *ungrammatical*. (**?**: 433)

Los lenguajes

En adelante entenderé que una *lengua* es un conjunto (finito o infinito) de oraciones, cada una de ellas de una longitud finita y construida a partir de un conjunto de elementos finito.
Chomsky 1957, 27.

Los lenguajes

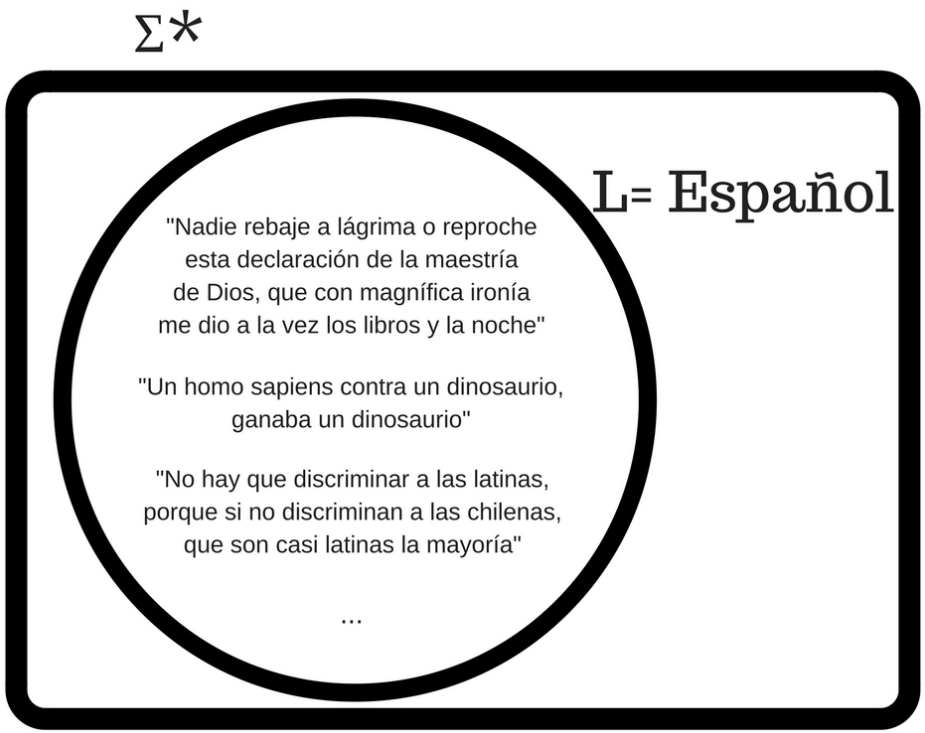


Figura: Imagen de ejemplo de la lengua natural española entendida como lenguaje formal a partir de un alfabeto Σ que contiene todos los grafemas del sistema de escritura del español

Los lenguajes

Dado un vocabulario $\{0, 1\}$, consideren los siguientes lenguajes:

Los lenguajes

Dado un vocabulario $\{0, 1\}$, consideren los siguientes lenguajes:

1. L1: cualquier número de unos y ceros en cualquier orden
2. L2: un número cualquiera de unos seguidos de un número cualquiera de ceros
3. L2: un número cualquiera de unos seguidos del mismo número de ceros.
4. L3: un número de unos equivalente al cuadrado del número de ceros que haya
5. L4: un par formado por una cadena de ceros y unos que conformen un programa que haga operaciones con cadenas de ceros y unos y una cadena de ceros y unos que sea un input válido para ese programa

Los lenguajes se caracterizan por su grado de complejidad en distintos tipos.

- Lenguajes irrestrictos
- Lenguajes sensibles al contexto
- Lenguajes independientes de contexto
- Lenguajes regulares

Formas de describir un lenguajes

Los lenguajes son entonces conjuntos de cadenas. No obstante, a menudo es difícil o imposible nombrarlos por extensión. Dos de las formas más comunes de caracterizar un lenguaje es a través de autómatas y de gramáticas.

Las gramáticas

A grammar (...) is some explicit device form (...) selecting a subset of strings that are grammatical, from the set of all possible strings formed from an initially given language.
(Partee : 433)

Gramáticas Tipo 0	Gramáticas irrestrictas
Gramáticas tipo 1	Gramáticas sensibles al contexto
Gramáticas tipo 2	Gramáticas independientes de contexto
Gramáticas tipo 3	Gramáticas regulares

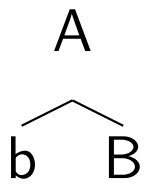
Las gramáticas se expresan usualmente en forma de reglas de reescritura de la forma $X \rightarrow Z$ (X se reescribe como Z) o de grafos dirigidos, coloquialmente denominados “árboles”.

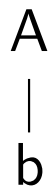
Gramáticas Regulares

Las gramáticas regulares son aquellas que solo pueden producir lenguajes regulares. Estas gramáticas restringen sus reglas de reescritura a solamente dos tipos:

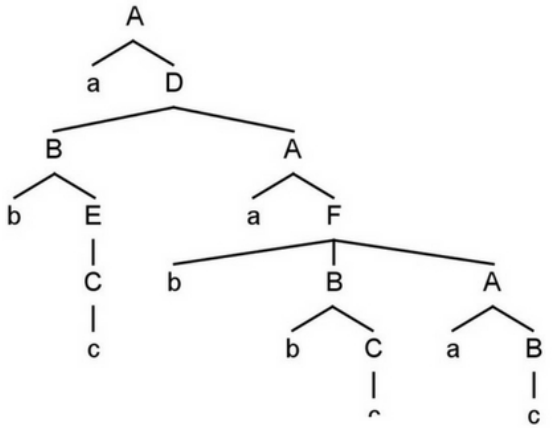
- $A \rightarrow b\ B$
- $A \rightarrow b$

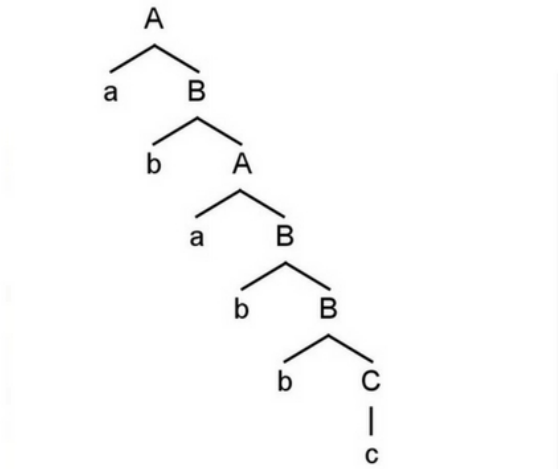
Es decir, a dos clases de árboles:





Gramáticas Regulares

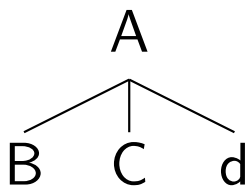


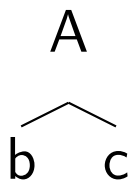


Gramáticas Independientes de Contexto

Las gramáticas independientes de contexto son capaces de generar lenguajes regulares e independientes de contexto y restringen sus reglas de reescritura a reglas que del lado izquierdo tengan un solo elemento. No tienen ninguna restricción respecto de la clase de cosas que puede haber del lado derecho.

$A \rightarrow$ lo que quieras





Gramáticas Sensibles al Contexto

Las gramáticas sensibles al contexto generan lenguajes regulares, independientes de contexto y sensibles al contexto. Sus reglas de reescritura permiten reescribir un elemento en función de los elementos que lo rodean. $ABC \rightarrow ADC$

Gramáticas Irrestrictas

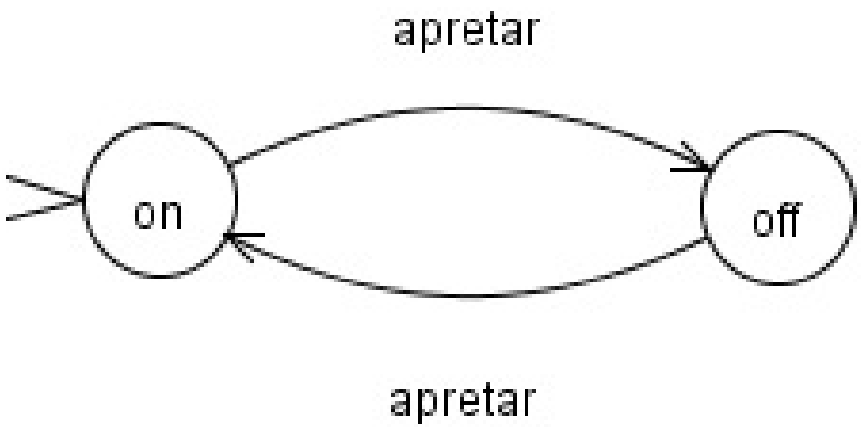
Las gramáticas irrestrictas pueden generar todos los tipos de lenguajes de la jerarquía de Chomsky. Sus reglas de reescritura pueden tomar un conjunto elementos y reescribirlo en otro conjunto, es decir, no restringen lo que puede haber ni del lado derecho ni del lado izquierdo de la regla de reescritura.

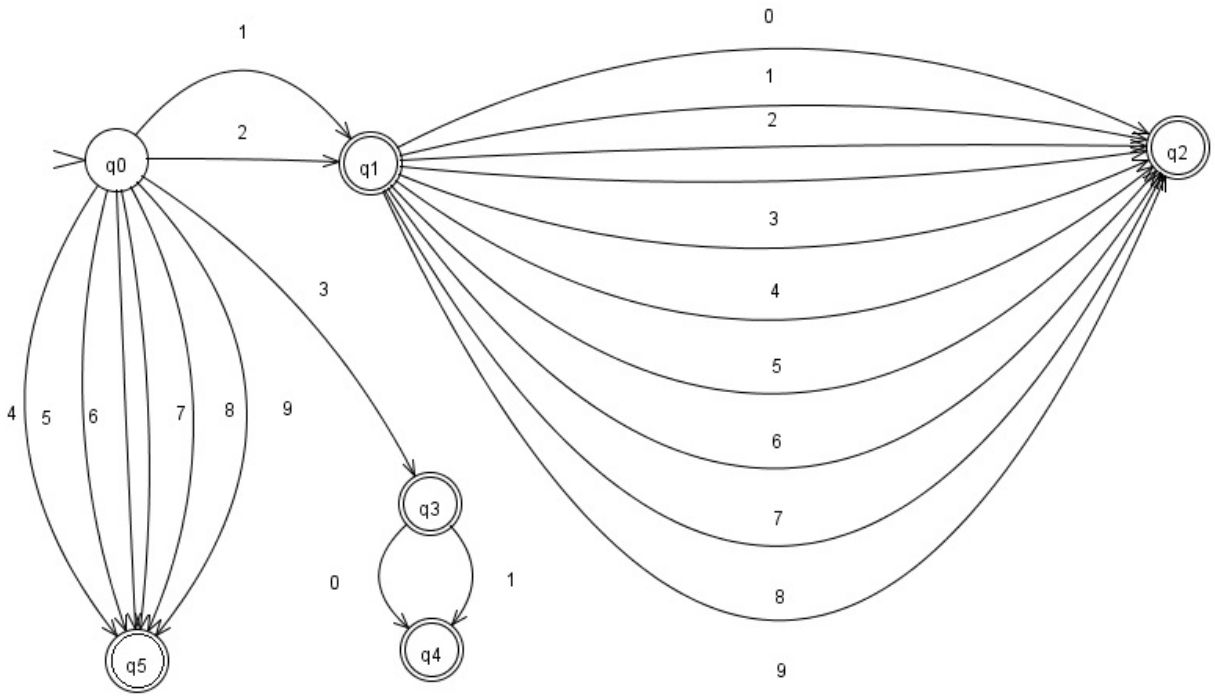
Autómatas

- Los autómatas son computadoras abstractas, esto es, mecanismos formales capaces de computar.
- Al ser abstractos, tienen la característica de tener todas las restricciones formales de las computadoras reales pero ninguna de las restricciones físicas o tecnológicas. Esto quiere decir que el estudio de los límites de lo que un autómata puede o no puede hacer permite saber cuáles son los límites de lo que una computadora podrá hacer jamás.
- Existen distintos tipos de autómata según su poder expresivo. Cuanto más complejo es un lenguaje, se necesita un autómata de mayor poder.

Autómatas de estados finitos

Producen lenguajes regulares, esto es, tienen un poder equivalente a las gramáticas regulares.




q0
q1
q2
q3
q4
q5
1
2
3
4
5
6
7
8
9
0
1
2
3
4
5
6
7
8
9
0
1

Autómatas de pila

La pila es una memoria. Cuando tienen una sola pila, estos autómatas producen lenguajes independientes de contexto, esto es, tienen un poder equivalente a una gramática independiente de contexto.

Autómatas linealmente acotados

Generan lenguajes sensibles al contexto, por lo que tienen un poder equivalente a una gramática sensible al contexto. Constan de un conjunto de reglas que pueden leer una cadena y cambiar los símbolos de esa cadena pero no pueden extenderse más allá de los límites de esa cadena.

Máquinas de Turing

Las máquinas de Turing son los autómatas más poderosos de todos. Constan de un conjunto de reglas que pueden leer una cadena y cambiar los símbolos de esa cadena, así como extender la cadena hacia atrás o hacia adelante.

Equivalencia entre Lenguajes y problemas

- La pregunta por la pertenencia de una determinada cadena a un lenguaje a partir de un alfabeto Σ califica de *problema*.
- Del mismo modo, cualquier problema puede traducirse en términos de la pregunta de la pertenencia de una cadena a un lenguaje.

La jerarquía de Chomsky

Si se ordenan los lenguajes, autómatas y gramáticas equivalentes entre sí en una escala de mayor a menor complejidad, se obtiene lo que se conoce como Jerarquía de Chomsky

Lenguajes irrestrictos	Gramáticas Tipo 0 o irrestrictas	Máquinas de Turing
Lenguajes sensibles al contexto	Gramáticas tipo 1 o sensibles al contexto	Autómatas linealmente acotados
Lenguajes independientes de contexto	Gramáticas tipo 2 o independientes de contexto	Autómatas de pila
Lenguajes regulares	Gramáticas tipo 3 o Gramáticas regulares	Autómatas de estados finitos

Complejidad

El principio de la **complejidad** mide la dificultad de resolver un problema computacional, medido en términos de recursos consumidos durante la computación. Normalmente se toma como referencia el espacio o el tiempo. (...) Cuanto más complejo sea el autómata permitido, tanto más complejas serán las lenguas reconocidas por él. (Sandoval 2001: 233)

Complejidad medida en términos de tiempo

Tipos de tiempos	**Notación O mayúscula**
Tiempo constante	$O(1)$
Tiempo lineal	$O(n)$
Tiempo polinómico	$O(n^c)$
Tiempo exponencial	$O(c^n)$
tiempo factorial	$O(n!)$

Complejidad medida en términos de tiempo

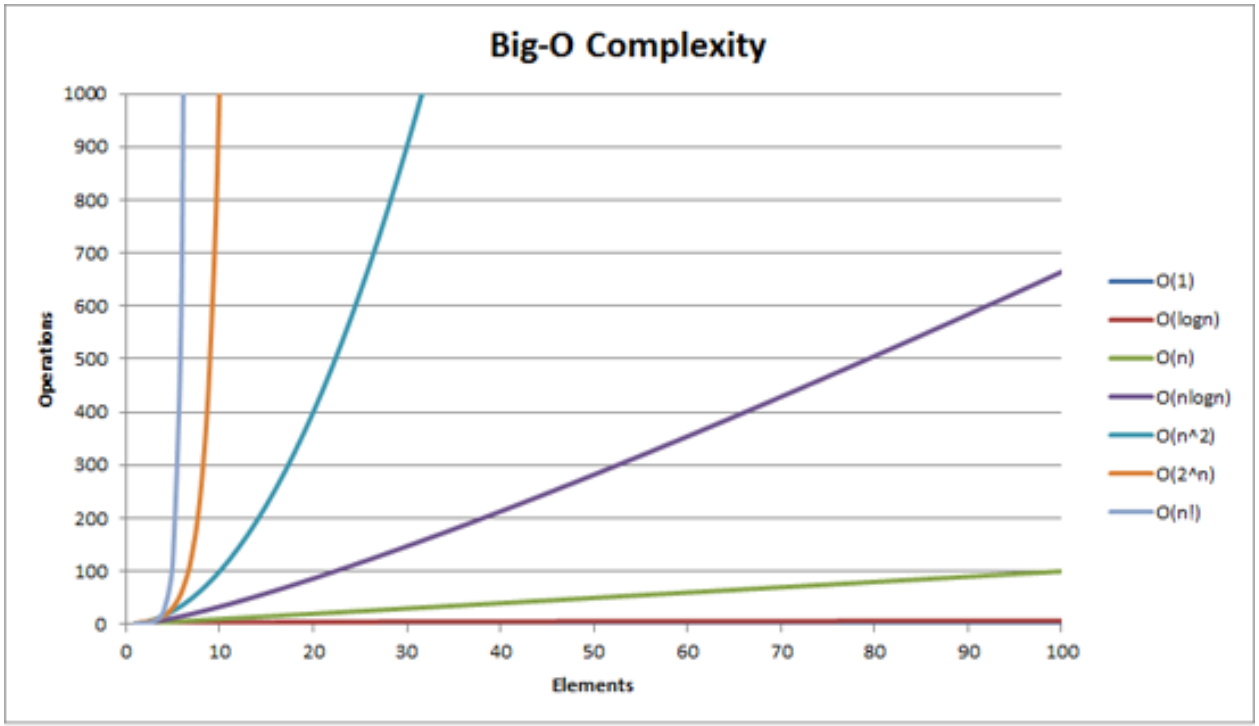


Figura: Comparación de curvas de los distintos costos de procesamiento (tomado de https://i.stack.imgur.com/Aq09a.png)

Complejidad medida en términos de tiempo

Se sabe que el procedimiento para decidir si un elemento pertenece a un lenguaje tiene un costo de procesamiento según la siguiente tabla:

Tipos de tiempos	**Notación O mayúscula**
Lenguajes regulares	tiempo lineal
Lenguajes independientes de contexto	tiempo polinómico
Lenguajes sensibles al contexto	tiempo exponencia (intratable)
Lenguajes irrestrictos	indecidible

Capacidad generativa fuerte y débil

La *capacidad generativa fuerte* de una teoría lingüística T es el conjunto de los sistemas D_i de descripciones lingüísticas para la gramática G_i que la teoría T proporciona. La *capacidad generativa débil* es el conjunto de conjuntos de oraciones L_i determinados por cada gramática G_i.
Quesada, J. Daniel. (1974). *La lingüística generativo transformacional: supuestos e implicaciones*. Madrid: Alianza. p. 98)

Capacidad generativa fuerte y débil

La **capacidad generativa** de una gramática es el término utilizado en lingüística para referirse a la capacidad de predicción de una gramática. Es decir, la forma en que describen explícitamente todas las oraciones de una lengua. Esta capacidad predictiva puede ser débil o fuerte. Se dice que tienen *capacidad generativa débil* cuando las reglas predicen cómo son las oraciones generadas por una gramática (gramaticales o agramaticales). *Capacidad generativa fuerte* es cuando las reglas además pueden predecir qué estructura subyace a las oraciones que describen. Es decir, proporcionan un diagrama o representación de la estructura interna de la oración, lo que también se conoce por 'asignación de estructura'.
Moreno Sandoval, Antonio (2001). *Gramáticas de Unificación y rasgos*. Madrid: Antonio Machado, p. 230.

Bibliografía usada

Sandoval, A. M. (2001). *Gramáticas de unificación y rasgos*. A. Machado Libros.