

Lenguajes Formales

Gabriel Del Salto, Sebastian Quinga, Stephani Rivera

Universidad de las Fuerzas Armadas ESPE

Modelos Discretos para Ingeniería

NRC: 9901

Sangolquí, 24 de mayo del 2023

Índice:

Introducción	3
Objetivo General	3
Desarrollo	3
Lenguajes Regulares	4
Ejemplos:	5
Lenguajes Libres De Contexto	5
Ejemplo:	5
Lenguajes Sensibles de Contexto	6
Lenguajes recursivamente enumerables	6
Conclusiones	7
Recomendaciones	8
Bibliografía	8

Introducción

La teoría de lenguajes formales estudia unas de las entidades matemáticas abstractas denominadas lenguajes que en ningún momento debemos confundir o equiparar con las lenguas naturales. (Balari, 2014)

En el presente documento hablaremos sobre los lenguajes formales, su definición, tipos de lenguajes formales, etc. Sabemos que un lenguaje formal es un lenguaje artificial que, a diferencia de los lenguajes naturales, no está centrado en la comunicación humana habitual, sino en la definición y aplicación de una serie de reglas específicas en un entorno específico.

Objetivo General

El objetivo de esta consulta es analizar y comprender los diferentes tipos de lenguajes formales utilizados en modelos discretos, con el fin de adquirir conocimientos sobre su estructura, características y aplicaciones. Se busca examinar los lenguajes regulares, los lenguajes libres de contexto, los lenguajes sensibles al contexto y los lenguajes recursivamente enumerables, comprendiendo sus propiedades y su relación con los modelos de computación.

Desarrollo

Un lenguaje formal es un lenguaje artificial que, a diferencia de los lenguajes naturales, no está centrado en la comunicación humana habitual, sino en la definición y aplicación de una serie de reglas específicas en un entorno específico. Consta de una serie de símbolos básicos, que se combinan en cadenas para formar expresiones complejas. (Editorial Etecé, 2022)

Los símbolos representan elementos de información como letras, números o símbolos especiales. Estos símbolos se combinan de acuerdo con las reglas de sintaxis establecidas, que determinan la estructura y el orden correcto de los símbolos en la estructura del lenguaje. La gramática determina qué secuencias de símbolos se consideran válidas y cuáles no.

- Símbolos terminales (Σ). Comprenden los elementos básicos del lenguaje (el *alfabeto*).
- Símbolos no terminales (N). Denominados a veces “variables sintácticas”, son de carácter auxiliar; expresan estados intermedios en el proceso de generación de las palabras. No forman parte del lenguaje.

- Símbolo inicial o axioma ($S \in N$). Es un símbolo no terminal de inicio, del que se parte para la formación de cualquiera de las palabras del lenguaje.
- Conjunto de reglas de producción (P). Son reglas que, partiendo del símbolo inicial, permiten realizar las transformaciones necesarias para obtener las palabras del lenguaje, mediante el reemplazo de los símbolos no terminales por símbolos terminales. (Editorial Etecé, 2022)

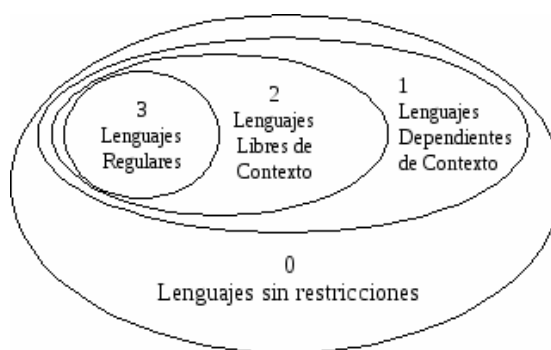


Figura1. https://www.ecured.cu/Lenguaje_regular.

Lenguajes Regulares

Los lenguajes regulares son una clase de lenguajes formales que pueden ser descritos por expresiones regulares o reconocidos por autómatas finitos. Los lenguajes regulares son la base de la teoría informática y tienen diversas aplicaciones en programación, compilación, análisis de texto y otros campos.

Un lenguaje regular consta de un conjunto de cadenas, o palabras, donde cada cadena consta de símbolos que pertenecen a un alfabeto dado. El alfabeto es solo un conjunto finito de símbolos.

Para describir los lenguajes regulares se utilizan expresiones regulares, que son descripciones formales de patrones de búsqueda o cadenas pertenecientes al lenguaje. Estas expresiones regulares se construyen usando operadores especiales y símbolos que representan ciertos comportamientos o propiedades en cadenas.

Los lenguajes regulares son utilizados en diversas aplicaciones de la teoría de la computación, como el análisis léxico en compiladores, la búsqueda y manipulación de cadenas en procesamiento de texto, la especificación de patrones en expresiones regulares utilizadas en la mayoría de los lenguajes de programación, entre otros. (Universidad Nacional Autónoma De México, 2011).

Ejemplos:

Un lenguaje es regular si resulta de una de estas operaciones:

1. *Unión*

Si L_1 y L_2 son regulares, entonces $L_1 \cup L_2$ es regular también.

2. *Concatenación*

Si L_1 y L_2 son regulares, entonces $L_1 L_2$ es regular también.

3. *Cerradura estrella*

Si L es regular, entonces L^* es regular también.

4. *Priorización por paréntesis*

Si L es regular, entonces (L) es regular también.

Lenguajes Libres De Contexto

El lenguaje libre de contexto es una clase de lenguajes formales que pueden ser descritos por una gramática libre de contexto. Estos lenguajes son más expresivos que los lenguajes regulares y se usan ampliamente en teoría informática, compilación de lenguajes de programación y otras áreas de la informática.

Una gramática libre de contexto consta de un conjunto de reglas de producción que especifican cómo se generan o derivan cadenas de símbolos. Cada regla de producción consiste en un no terminal que puede ser sustituido y una secuencia de terminales y no terminales que representan la expansión o sustitución del no terminal.

Los lenguajes libres de contexto son utilizados en muchas áreas de la informática, gramaticales en un lenguaje. (Universidad Autónoma de Puebla, 2012)

Ejemplo:

Una gramática $G = (V, T, S, P)$ se dice que es libre de contexto si todas las producciones en P tienen la forma $A \rightarrow x$, donde $A \in V$ y $x \in (V \cup T)^*$.

Se dice que un lenguaje L es libre de contexto si y sólo si existe una gramática libre de contexto G tal que $L = L(G)$.

La gramática $G = (\{S\}, \{a, b\}, S, P)$ con producciones.

$$S \rightarrow aSa,$$

$$S \rightarrow bSb,$$

$$S \rightarrow \lambda,$$

Lenguajes Sensibles de Contexto

Los lenguajes sensibles al contexto son un tipo de lenguaje formal en la teoría de la computación. Estos lenguajes son más expresivos que los lenguajes libres de contexto y pueden describir estructuras más complejas que requieren contextos específicos para su análisis.

En un lenguaje sensible al contexto, las reglas de producción de la gramática están sujetas a restricciones en cuanto a la forma de las producciones y especifican condiciones contextuales para su aplicación. Estas condiciones implican que la producción sólo puede aplicarse en un contexto determinado, teniendo en cuenta los símbolos vecinos en la cadena de entrada (Paredes, 2019).

Un ejemplo de lenguaje sensible al contexto es el lenguaje de los palíndromos de la forma $a^n b^n c^n d^n$, donde n es un número natural. Este lenguaje se puede describir utilizando una gramática sensible al contexto, donde las reglas de producción especifican las condiciones de igualdad de los símbolos de cada bloque (por ejemplo, a^n debe ser igual a b^n , y c^n debe ser igual a d^n).

- 1) $S \rightarrow abc \mid aAbc$
 - 2) $Ab \rightarrow bA$
 - 3) $Ac \rightarrow Bbcc$
 - 4) $bB \rightarrow Bb$
 - 5) $aB \rightarrow aaA \mid aa$

Figura 3.

<https://www.semanticscholar.org/paper/Reconocedor-de-lenguajes-con-base-en-gram%C3%A1tica-Salas-Alberto/84703f3883fd7ada2fa7d3ca806bef6dbb210506>

Lenguajes recursivamente enumerables

Los lenguajes recursivamente enumerables, también conocidos como lenguajes recursivamente enumerables o semidecidibles, son un tipo de lenguaje formal en la teoría de

la computación. Estos lenguajes representan una clase de lenguajes que pueden ser reconocidos por una máquina de Turing, pero no necesariamente decididos por ella.

Un lenguaje recursivamente enumerable puede ser generado por una máquina de Turing que eventualmente terminará en un estado de aceptación para las cadenas que pertenecen al lenguaje. Sin embargo, si una cadena no pertenece al lenguaje, la máquina de Turing podría entrar en un bucle infinito o no terminar en ningún estado de aceptación.

Una característica distintiva de los lenguajes recursivamente enumerables es que no existe un algoritmo general para decidir si una cadena dada pertenece o no al lenguaje. En otras palabras, no se puede construir una máquina de Turing que, para cualquier cadena de entrada, termine en un estado de aceptación o rechazo en un tiempo finito (Villar, 2012).

Un ejemplo de un lenguaje recursivamente enumerable es el lenguaje de las máquinas de Turing que acepta una cadena vacía. Este lenguaje consiste en todas las descripciones de máquinas de Turing que, al comenzar con una cinta en blanco, eventualmente se detienen y aceptan la cadena vacía. La máquina de Turing puede ser configurada de tal manera que siempre se detenga para las cadenas vacías, pero no hay un algoritmo general para determinar si una descripción de una máquina de Turing aceptará o no una cadena dada.



Figura 4.

<http://pitxaunlio.blogspot.com/2012/11/chomsky-i-la-maquina-de-turing.html>

Conclusiones

- La consulta nos ha permitido ampliar nuestros conocimientos sobre los lenguajes formales en modelos discretos, su estructura, características y aplicaciones. Esto nos proporciona una base sólida para comprender y abordar problemas relacionados con la teoría de la computación y la ciencia de la computación en general.

- El estudio de diferentes tipos de lenguajes formales utilizados en modelos discretos proporciona una comprensión profunda de la estructura y propiedades de estos lenguajes. Esto permite a los investigadores y especialistas en computación facilitar el desarrollo e implementación de sistemas y algoritmos.
- Comprender las limitaciones y capacidades de los diferentes tipos de lenguaje, los investigadores pueden elegir el modelo computacional correcto y desarrollar soluciones eficientes y correctas.

Recomendaciones

- Continúa explorando los lenguajes formales y su relación con los modelos de computación para fortalecer tus conocimientos en teoría de la computación.
- Explorar y estudiar ejemplos concretos de cada tipo de lenguaje formal. La práctica con la gramática puede ayudar a consolidar el conocimiento y a comprender mejor las reglas y las propiedades de los lenguajes formales.
- Aplicar los conocimientos adquiridos en proyectos reales. El uso de un lenguaje formal para implementar un sistema o desarrollar un algoritmo ayuda a profundizar la comprensión y ganar experiencia en el uso.

Bibliografía

Balari, S. (2014). *Teoría de lenguajes formales* (Universidad Autónoma de Barcelona ed.).

https://ddd.uab.cat/pub/lilibres/2014/116304/teolenfor_a2014p1iSPA.pdf

Editorial Etecé. (2022, 07 26). *Lenguajes formales*. Concepto.

<https://concepto.de/lenguajes-formales/>

Editorial Etecé. (2022, 07 26). *Lenguajes Formales*. Conceptos.

<https://concepto.de/lenguajes-formales/>

Paredes, M. (2019, Enero 12). *EXA. GRAMÁTICAS SENSIBLES AL CONTEXTO.*

Retrieved May 24, 2023, from

<https://users.exa.unicen.edu.ar/catedras/ccomp1/ApunteGramSensibles.pdf>

Universidad Autónoma de Puebla. (2012). *Lenguajes Libres de Contexto.* aleteya.

<http://aleteya.cs.buap.mx/~jlavalle/automata/5.1%20Lenguajes%20Libres%20de%20Contexto.pdf>

Universidad Nacional Autónoma De México. (2011, 08). *Tipos de Lenguajes Formales.*

[file:///C:/Users/Gaby/AppData/Local/Microsoft/Windows/INetCache/IE/OVK 483/lenguajes Nivardo\[1\].pdf](file:///C:/Users/Gaby/AppData/Local/Microsoft/Windows/INetCache/IE/OVK%20483/lenguajes+Nivardo%5B1%5D.pdf)

Villar, M. (2012, November 15). *Chomsky i la màquina de Turing.* La pitxa un lio. Retrieved

May 24, 2023, from

<http://pitxaunlio.blogspot.com/2012/11/chomsky-i-la-maquina-de-turing.html>