Una Jerarquía de Lenguajes Formales y Autómatas III

José de Jesús Lavalle Martínez

Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación
Lenguajes Formales y Autómatas CCOS 014

Contenido

Motivación

- Entre las gramáticas restringidas y las gramáticas sin restricciones, se pueden definir una gran variedad de gramáticas “restringidas en algún sentido”.

Motivación

- No todos los casos arrojan resultados interesantes; entre las que sí lo hacen, las gramáticas sensibles al contexto han recibido una atención considerable.

Motivación

- Estas gramáticas generan lenguajes asociados con una clase restringida de máquinas de Turing, los autómatas acotados linealmente, que presentamos en la sesión anterior.

Definición 1

Se dice que una gramática $G = (V, T, S, P)$ es sensible al contexto si todas las producciones son de la forma

$$x \to y,$$

donde $x, y \in (V \cup T)^+$ y

$$|x| \leq |y|. \tag{1}$$

Gramáticas sensibles al contexto y lenguajes II

- Esta definición muestra claramente un aspecto de este tipo de gramáticas; no se contraen, en el sentido de que la longitud de las sucesivas formas sentenciales nunca puede disminuir.

Gramáticas sensibles al contexto y lenguajes II

- Es menos obvio por qué tales gramáticas deben llamarse sensibles al contexto, pero se puede demostrar (ver, por ejemplo, Salomaa 1973) que todas esas gramáticas pueden reescribirse en una forma normal en la que todas las producciones son de la forma

$$xAy \rightarrow xvy.$$

Gramáticas sensibles al contexto y lenguajes II

- Es menos obvio por qué tales gramáticas deben llamarse sensibles al contexto, pero se puede demostrar (ver, por ejemplo, Salomaa 1973) que todas esas gramáticas pueden reescribirse en una forma normal en la que todas las producciones son de la forma

$$xAy \rightarrow xvy.$$

- Esto equivale a decir que la producción

$$A \rightarrow v$$

se puede aplicar solo en la situación en la que A ocurre en un contexto donde la cadena x está a la izquierda y la cadena y a la derecha.

Gramáticas sensibles al contexto y lenguajes II

- Si bien usamos la terminología que surge de esta interpretación en particular, la forma en sí es de poco interés para nosotros aquí, y nos basaremos completamente en la Definición 1.

Lenguajes sensibles al contexto y autómatas acotados linealmente I

- Como sugiere la terminología, las gramáticas sensibles al contexto están asociadas con una familia de lenguajes con el mismo nombre.

Lenguajes sensibles al contexto y autómatas acotados linealmente I

Definición 2

Se dice que un lenguaje L es sensible al contexto si existe una gramática sensible al contexto G, tal que $L = L(G)$ o $L = L(G) \cup \{\lambda\}$.

Lenguajes sensibles al contexto y autómatas acotados linealmente II

- En esta definición, reintroducimos la cadena vacía.

Lenguajes sensibles al contexto y autómatas acotados linealmente II

- La Definición 1 implica que $x \rightarrow \lambda$ no está permitida, por lo que una gramática sensible al contexto nunca puede generar un lenguaje que contenga la cadena vacía.

Lenguajes sensibles al contexto y autómatas acotados linealmente II

- Sin embargo, cada lenguaje libre de contexto sin λ puede ser generado por un caso especial de una gramática sensible al contexto, digamos por una en la forma normal de Chomsky o Greibach, las cuales satisfacen las condiciones de la Definición 1.

Lenguajes sensibles al contexto y autómatas acotados linealmente II

- Al incluir la cadena vacía en la definición de un lenguaje sensible al contexto (pero no en la gramática), podemos afirmar que la familia de lenguajes libres de contexto es un subconjunto de la familia de lenguajes sensibles al contexto.

Ejemplo 1 I

Ejemplo 1

El lenguaje $L = \{a^n b^n c^n : n \geq 1\}$ es un lenguaje sensible al contexto.

- Demostramos esto exhibiendo una gramática sensible al contexto para el lenguaje.

Ejemplo 1 I

Ejemplo 1

El lenguaje $L = \{a^n b^n c^n : n \geq 1\}$ es un lenguaje sensible al contexto.

- Demostramos esto exhibiendo una gramática sensible al contexto para el lenguaje.
- Tal gramática es

$$\begin{aligned} S &\to abc|aAbc, \\ Ab &\to bA, \\ Ac &\to Bbcc, \\ bB &\to Bb, \\ aB &\to aa|aaA. \end{aligned}$$

Ejemplo 1 I

Ejemplo 1

El lenguaje $L = \{a^n b^n c^n : n \geq 1\}$ es un lenguaje sensible al contexto.

- Demostramos esto exhibiendo una gramática sensible al contexto para el lenguaje.
- Tal gramática es

$$\begin{aligned} S &\to abc|aAbc, \\ Ab &\to bA, \\ Ac &\to Bbcc, \\ bB &\to Bb, \\ aB &\to aa|aaA. \end{aligned}$$

- Podemos ver cómo funciona esto al observar una derivación de $a^3b^3c^3$.

$$\begin{aligned} S &\Rightarrow aAbc \Rightarrow abAc \Rightarrow abBbcc \\ &\Rightarrow aBbbcc \Rightarrow aaAbbcc \Rightarrow aabAbcc \\ &\Rightarrow aabbAcc \Rightarrow aabbBbccc \\ &\Rightarrow aabBbbccc \Rightarrow aaBbbbccc \\ &\Rightarrow aaabbbccc. \end{aligned}$$

Ejemplo 1 II

$$
\begin{aligned}
S &\rightarrow abc|aAbc,\\
Ab &\rightarrow bA,\\
Ac &\rightarrow Bbcc,\\
bB &\rightarrow Bb,\\
aB &\rightarrow aa|aaA.
\end{aligned}
$$

- La solución utiliza eficazmente las variables A y B como mensajeros.

Ejemplo 1 II

$$
\begin{aligned}
S &\to abc|aAbc,\\
Ab &\to bA,\\
Ac &\to Bbcc,\\
bB &\to Bb,\\
aB &\to aa|aaA.
\end{aligned}
$$

- La solución utiliza eficazmente las variables A y B como mensajeros.
- Se crea una A a la izquierda, viaja a la derecha hasta la primera c, donde crea otra b y c.

Ejemplo 1 II

$$
\begin{aligned}
S &\to abc|aAbc,\\
Ab &\to bA,\\
Ac &\to Bbcc,\\
bB &\to Bb,\\
aB &\to aa|aaA.
\end{aligned}
$$

- La solución utiliza eficazmente las variables A y B como mensajeros.
- Se crea una A a la izquierda, viaja a la derecha hasta la primera c, donde crea otra b y c.
- Luego envía al mensajero B de regreso a la izquierda para crear la correspondiente a.

Ejemplo 1 II

$$
\begin{aligned}
S &\rightarrow abc|aAbc,\\
Ab &\rightarrow bA,\\
Ac &\rightarrow Bbcc,\\
bB &\rightarrow Bb,\\
aB &\rightarrow aa|aaA.
\end{aligned}
$$

- La solución utiliza eficazmente las variables A y B como mensajeros.
- Se crea una A a la izquierda, viaja a la derecha hasta la primera c, donde crea otra b y c.
- Luego envía al mensajero B de regreso a la izquierda para crear la correspondiente a.
- El proceso es muy similar a la forma en que se puede programar una máquina de Turing para que acepte el lenguaje L.

Lenguajes sensibles al contexto y autómatas acotados linealmente III

- Dado que el lenguaje del ejemplo anterior no es libre de contexto, vemos que la familia de lenguajes libres de contexto es un subconjunto propio de la familia de lenguajes sensibles al contexto.

Lenguajes sensibles al contexto y autómatas acotados linealmente III

- El Ejemplo 1 también muestra que no es fácil encontrar una gramática sensible al contexto, incluso para ejemplos relativamente simples.

Lenguajes sensibles al contexto y autómatas acotados linealmente III

- A menudo, la solución se obtiene más fácilmente comenzando con un programa de máquina de Turing y luego encontrando una gramática equivalente para él.

Lenguajes sensibles al contexto y autómatas acotados linealmente III

- Algunos ejemplos mostrarán que, siempre que el lenguaje sea sensible al contexto, la máquina de Turing correspondiente tiene requisitos de espacio predecibles; en particular, puede verse como un autómata acotado linealmente.

Teorema 1 I

Teorema 1

Para cada lenguaje sensible al contexto L sin incluir λ, existe algún autómata acotado linealmente M tal que $L = L(M)$.

Teorema 1 I

Teorema 1

Para cada lenguaje sensible al contexto L sin incluir λ, existe algún autómata acotado linealmente M tal que $L = L(M)$.

Demostración:

- Si L es sensible al contexto, entonces existe una gramática sensible al contexto para $L - \{\lambda\}$.

Teorema 1 I

Teorema 1

Para cada lenguaje sensible al contexto L sin incluir λ, existe algún autómata acotado linealmente M tal que $L = L(M)$.

Demostración:

- Mostramos que las derivaciones en esta gramática pueden ser simuladas por un autómata acotado linealmente.

Teorema 1 I

Teorema 1

Para cada lenguaje sensible al contexto L sin incluir λ, existe algún autómata acotado linealmente M tal que $L = L(M)$.

Demostración:

- El autómata acotado linealmente tendrá dos pistas, una que contiene la cadena de entrada w, la otra que contiene las formas sentenciales derivadas usando G.

Teorema 1 I

Teorema 1

Para cada lenguaje sensible al contexto L sin incluir λ, existe algún autómata acotado linealmente M tal que $L = L(M)$.

Demostración:

- Un punto clave de este argumento es que ninguna forma sentencial puede tener una longitud mayor que $|w|$.

Teorema 1 II

- Otro punto a tener en cuenta es que un autómata acotado linealmente es, por definición, no determinista.

Teorema 1 II

- Esto es necesario en el argumento, ya que podemos afirmar que siempre se puede adivinar la producción correcta y que no es necesario buscar alternativas improductivas.

Teorema 1 II

- Por lo tanto, el cálculo descrito en el Teorema 1 de la sesión anterior (Cualquier lenguaje generado por una gramática sin restricciones es enumerable de forma recursiva.) se puede realizar sin usar espacio excepto el que originalmente ocupaba w; es decir, se puede realizar mediante un autómata acotado linealmente.

■

Teorema 2 I

Teorema 2

Si un lenguaje L es aceptado por algún autómata acotado linealmente M, entonces existe una gramática sensible al contexto que genera a L.

Teorema 2 I

Teorema 2

Si un lenguaje L es aceptado por algún autómata acotado linealmente M, entonces existe una gramática sensible al contexto que genera a L.

Demostración:

- La construcción aquí es similar a la del Teorema 2 de la sesión anterior (Para cada lenguaje L recursivamente enumerable, existe una gramática G sin restricciones, tal que $L = L(G)$).

Teorema 2 I

Teorema 2

Si un lenguaje L es aceptado por algún autómata acotado linealmente M, entonces existe una gramática sensible al contexto que genera a L.

Demostración:

- Todas las producciones generadas en el Teorema 2 de la sesión anterior no se contraen excepto,

$$\square \rightarrow \lambda$$

Teorema 2 I

Teorema 2

Si un lenguaje L es aceptado por algún autómata acotado linealmente M, entonces existe una gramática sensible al contexto que genera a L.

Demostración:

- Pero esta producción puede omitirse.

Teorema 2 I

Teorema 2

Si un lenguaje L es aceptado por algún autómata acotado linealmente M, entonces existe una gramática sensible al contexto que genera a L.

Demostración:

- Sólo es necesaria cuando la máquina de Turing se mueve fuera de los límites de la entrada original, lo cual no es el caso aquí.

Teorema 2 I

Teorema 2

Si un lenguaje L es aceptado por algún autómata acotado linealmente M, entonces existe una gramática sensible al contexto que genera a L.

Demostración:

- La gramática obtenida por la construcción sin esta producción innecesaria no se contrae, completando el argumento.

Relación entre lenguajes recursivos y sensibles al contexto I

- El teorema 2 nos dice que cada lenguaje sensible al contexto es aceptado por alguna máquina de Turing y, por lo tanto, es recursivamente enumerable.
- El teorema 3 se sigue fácilmente de esto.

Teorema 3 I

Teorema 3

Todo lenguaje L sensible al contexto es recursivo.

Teorema 3 I

Teorema 3

Todo lenguaje L sensible al contexto es recursivo.

Demostración:

- Considere el lenguaje sensible al contexto L con una gramática sensible al contexto asociada G, y observe una derivación de w

$$S \Rightarrow x_1 \Rightarrow x_2 \Rightarrow \cdots \Rightarrow x_n \Rightarrow w.$$

Teorema 3 I

Teorema 3

Todo lenguaje L sensible al contexto es recursivo.

Demostración:

- Considere el lenguaje sensible al contexto L con una gramática sensible al contexto asociada G, y observe una derivación de w

$$S \Rightarrow x_1 \Rightarrow x_2 \Rightarrow \cdots \Rightarrow x_n \Rightarrow w.$$

- Podemos suponer sin ninguna pérdida de generalidad que todas las formas sentenciales en una sola derivación son diferentes; es decir, $x_i \neq x_j$ para todo $i \neq j$.

Teorema 3

Todo lenguaje L sensible al contexto es recursivo.

Demostración:

- Considere el lenguaje sensible al contexto L con una gramática sensible al contexto asociada G, y observe una derivación de w

$$S \Rightarrow x_1 \Rightarrow x_2 \Rightarrow \cdots \Rightarrow x_n \Rightarrow w.$$

- Podemos suponer sin ninguna pérdida de generalidad que todas las formas sentenciales en una sola derivación son diferentes; es decir, $x_i \neq x_j$ para todo $i \neq j$.
- El quid de nuestro argumento es que el número de pasos en cualquier derivación es una función acotada de $|w|$.

Teorema 3

Todo lenguaje L sensible al contexto es recursivo.

Demostración:

- Considere el lenguaje sensible al contexto L con una gramática sensible al contexto asociada G, y observe una derivación de w

$$S \Rightarrow x_1 \Rightarrow x_2 \Rightarrow \cdots \Rightarrow x_n \Rightarrow w.$$

- Podemos suponer sin ninguna pérdida de generalidad que todas las formas sentenciales en una sola derivación son diferentes; es decir, $x_i \neq x_j$ para todo $i \neq j$.
- El quid de nuestro argumento es que el número de pasos en cualquier derivación es una función acotada de $|w|$.
- Sabemos que

$$|x_j| \leq |x_{j+1}|,$$

porque G no se contrae.

Teorema 3 II

- Lo único que necesitamos agregar es que existe algún m, dependiendo sólo de G y w, de manera que

$$|x_j| < |x_{j+m}|,$$

para todo j, con $m = m(|w|)$ una función acotada de $|V \cup T|$ y $|w|$.

Teorema 3 II

- Lo único que necesitamos agregar es que existe algún m, dependiendo sólo de G y w, de manera que

$$|x_j| < |x_{j+m}|,$$

  para todo j, con $m = m(|w|)$ una función acotada de $|V \cup T|$ y $|w|$.
- Esto se debe a que la finitud de $|V \cup T|$ implica que sólo hay un número finito de cadenas de una longitud determinada.

Teorema 3 II

- Lo único que necesitamos agregar es que existe algún m, dependiendo sólo de G y w, de manera que

$$|x_j| < |x_{j+m}|,$$

  para todo j, con $m = m(|w|)$ una función acotada de $|V \cup T|$ y $|w|$.
- Esto se debe a que la finitud de $|V \cup T|$ implica que sólo hay un número finito de cadenas de una longitud determinada.
- Por lo tanto, la longitud de una derivación de $w \in L$ es como máximo $|w|m(|w|)$.

Teorema 3 II

- Lo único que necesitamos agregar es que existe algún m, dependiendo sólo de G y w, de manera que

$$|x_j| < |x_{j+m}|,$$

  para todo j, con $m = m(|w|)$ una función acotada de $|V \cup T|$ y $|w|$.
- Esto se debe a que la finitud de $|V \cup T|$ implica que sólo hay un número finito de cadenas de una longitud determinada.
- Por lo tanto, la longitud de una derivación de $w \in L$ es como máximo $|w|m(|w|)$.
- Esta observación nos da inmediatamente un algoritmo de pertenencia para L.

Teorema 3 II

- Lo único que necesitamos agregar es que existe algún m, dependiendo sólo de G y w, de manera que

$$|x_j| < |x_{j+m}|,$$

  para todo j, con $m = m(|w|)$ una función acotada de $|V \cup T|$ y $|w|$.
- Esto se debe a que la finitud de $|V \cup T|$ implica que sólo hay un número finito de cadenas de una longitud determinada.
- Por lo tanto, la longitud de una derivación de $w \in L$ es como máximo $|w|m(|w|)$.
- Esta observación nos da inmediatamente un algoritmo de pertenencia para L.
- Comprobamos todas las derivaciones de longitud hasta $|w|m(|w|)$.

Teorema 3 II

- Lo único que necesitamos agregar es que existe algún m, dependiendo sólo de G y w, de manera que

$$|x_j| < |x_{j+m}|,$$

  para todo j, con $m = m(|w|)$ una función acotada de $|V \cup T|$ y $|w|$.
- Esto se debe a que la finitud de $|V \cup T|$ implica que sólo hay un número finito de cadenas de una longitud determinada.
- Por lo tanto, la longitud de una derivación de $w \in L$ es como máximo $|w|m(|w|)$.
- Esta observación nos da inmediatamente un algoritmo de pertenencia para L.
- Comprobamos todas las derivaciones de longitud hasta $|w|m(|w|)$.
- Dado que el conjunto de producciones de G es finito, solo hay un número finito de estos.

Teorema 3 II

- Lo único que necesitamos agregar es que existe algún m, dependiendo sólo de G y w, de manera que

$$|x_j| < |x_{j+m}|,$$

  para todo j, con $m = m(|w|)$ una función acotada de $|V \cup T|$ y $|w|$.
- Esto se debe a que la finitud de $|V \cup T|$ implica que sólo hay un número finito de cadenas de una longitud determinada.
- Por lo tanto, la longitud de una derivación de $w \in L$ es como máximo $|w|m(|w|)$.
- Esta observación nos da inmediatamente un algoritmo de pertenencia para L.
- Comprobamos todas las derivaciones de longitud hasta $|w|m(|w|)$.
- Dado que el conjunto de producciones de G es finito, solo hay un número finito de estos.
- Si alguno de ellos da w, entonces $w \in L$, de lo contrario no está en L.

Teorema 4 I

Teorema 4

Existe un lenguaje recursivo que no es sensible al contexto.

Teorema 4 I

Teorema 4

Existe un lenguaje recursivo que no es sensible al contexto.

Demostración:

- Considere el conjunto de todas las gramáticas sensibles al contexto sobre $T = \{a, b\}$.

Teorema 4 I

Teorema 4

Existe un lenguaje recursivo que no es sensible al contexto.

Demostración:

- Considere el conjunto de todas las gramáticas sensibles al contexto sobre $T = \{a, b\}$.
- Podemos usar una convención en la que cada gramática tiene un conjunto de variables de la forma

$$V = \{V_0, V_1, V_2, \ldots\}.$$

Teorema 4 I

Teorema 4

Existe un lenguaje recursivo que no es sensible al contexto.

Demostración:

- Considere el conjunto de todas las gramáticas sensibles al contexto sobre $T = \{a, b\}$.
- Podemos usar una convención en la que cada gramática tiene un conjunto de variables de la forma

$$V = \{V_0, V_1, V_2, \ldots\}.$$

- Cada gramática sensible al contexto está completamente especificada por sus producciones; podemos pensar en ellas como si estuvieran escritas en una sola cadena

$$x_1 \to y_1; x_2 \to y_2; \ldots; x_m \to y_m.$$

Teorema 4 II

- A esta cadena aplicamos ahora el homomorfismo

$$
\begin{aligned}
h(a) &= 010, \\
h(b) &= 01^2 0, \\
h(\rightarrow) &= 01^3 0, \\
h(;) &= 01^4 0, \\
h(V_i) &= 01^{i+5} 0.
\end{aligned}
$$

Teorema 4 II

- A esta cadena aplicamos ahora el homomorfismo

$$\begin{aligned} h(a) &= 010, \\ h(b) &= 01^2 0, \\ h(\rightarrow) &= 01^3 0, \\ h(;) &= 01^4 0, \\ h(V_i) &= 01^{i+5} 0. \end{aligned}$$

- Por lo tanto, cualquier gramática sensible al contexto se puede representar de forma única mediante una cadena de $L((011^*0)^*)$.

Teorema 4 II

- A esta cadena aplicamos ahora el homomorfismo

$$\begin{aligned} h(a) &= 010, \\ h(b) &= 01^2 0, \\ h(\rightarrow) &= 01^3 0, \\ h(;) &= 01^4 0, \\ h(V_i) &= 01^{i+5} 0. \end{aligned}$$

- Por lo tanto, cualquier gramática sensible al contexto se puede representar de forma única mediante una cadena de $L((011^*0)^*)$.
- Además, la representación es invertible en el sentido de que, dada dicha cadena, hay como máximo una gramática sensible al contexto correspondiente.

Teorema 4 III

- Introduzcamos un orden propio en $\{0,1\}^+$, de modo que podamos escribir cadenas en el orden w_1, w_2, etc.

Teorema 4 III

- Introduzcamos un orden propio en $\{0,1\}^+$, de modo que podamos escribir cadenas en el orden w_1, w_2, etc.
- Una cadena dada w_j puede no definir una gramática sensible al contexto; pero si lo hace, llame a la gramática G_j.

Teorema 4 III

- Introduzcamos un orden propio en $\{0,1\}^+$, de modo que podamos escribir cadenas en el orden w_1, w_2, etc.
- Una cadena dada w_j puede no definir una gramática sensible al contexto; pero si lo hace, llame a la gramática G_j.
- A continuación, definimos un lenguaje L mediante

$$L = \{w_i : w_i \text{ define una gramática sensible al contexto } G_i \text{ y } w_i \notin L(G_i)\}.$$

Teorema 4 III

- Introduzcamos un orden propio en $\{0,1\}^+$, de modo que podamos escribir cadenas en el orden w_1, w_2, etc.
- Una cadena dada w_j puede no definir una gramática sensible al contexto; pero si lo hace, llame a la gramática G_j.
- A continuación, definimos un lenguaje L mediante

$$L = \{w_i : w_i \text{ define una gramática sensible al contexto } G_i \text{ y } w_i \notin L(G_i)\}.$$

- L está bien definido y de hecho es recursivo.

Teorema 4 III

- Introduzcamos un orden propio en $\{0,1\}^+$, de modo que podamos escribir cadenas en el orden w_1, w_2, etc.
- Una cadena dada w_j puede no definir una gramática sensible al contexto; pero si lo hace, llame a la gramática G_j.
- A continuación, definimos un lenguaje L mediante

$$L = \{w_i : w_i \text{ define una gramática sensible al contexto } G_i \text{ y } w_i \notin L(G_i)\}.$$

- L está bien definido y de hecho es recursivo.
- Para ver esto, construimos un algoritmo de pertenencia.

Teorema 4 III

- Introduzcamos un orden propio en $\{0,1\}^+$, de modo que podamos escribir cadenas en el orden w_1, w_2, etc.
- Una cadena dada w_j puede no definir una gramática sensible al contexto; pero si lo hace, llame a la gramática G_j.
- A continuación, definimos un lenguaje L mediante

$$L = \{w_i : w_i \text{ define una gramática sensible al contexto } G_i \text{ y } w_i \notin L(G_i)\}.$$

- L está bien definido y de hecho es recursivo.
- Para ver esto, construimos un algoritmo de pertenencia.
- Dado w_i, comprobamos para ver si define un gramatica sensible al contexto G_i.

Teorema 4 III

- Introduzcamos un orden propio en $\{0,1\}^+$, de modo que podamos escribir cadenas en el orden w_1, w_2, etc.
- Una cadena dada w_j puede no definir una gramática sensible al contexto; pero si lo hace, llame a la gramática G_j.
- A continuación, definimos un lenguaje L mediante

$$L = \{w_i : w_i \text{ define una gramática sensible al contexto } G_i \text{ y } w_i \notin L(G_i)\}.$$

- L está bien definido y de hecho es recursivo.
- Para ver esto, construimos un algoritmo de pertenencia.
- Dado w_i, comprobamos para ver si define un gramatica sensible al contexto G_i.
- Si no, entonces $w_i \notin L$.

Teorema 4 IV

- Si la cadena define una gramática, entonces $L(G_i)$ es recursivo, y podemos usar el algoritmo de pertenencia del Teorema 3 para averiguar si $w_i \in L(G_i)$.

Teorema 4 IV

- Si la cadena define una gramática, entonces $L(G_i)$ es recursivo, y podemos usar el algoritmo de pertenencia del Teorema 3 para averiguar si $w_i \in L(G_i)$.
- Si no es así, entonces w_i pertenece a L.

Teorema 4 IV

- Si la cadena define una gramática, entonces $L(G_i)$ es recursivo, y podemos usar el algoritmo de pertenencia del Teorema 3 para averiguar si $w_i \in L(G_i)$.
- Si no es así, entonces w_i pertenece a L.
- Pero L no es sensible al contexto.

Teorema 4 IV

- Si la cadena define una gramática, entonces $L(G_i)$ es recursivo, y podemos usar el algoritmo de pertenencia del Teorema 3 para averiguar si $w_i \in L(G_i)$.
- Si no es así, entonces w_i pertenece a L.
- Pero L no es sensible al contexto.
- Si lo fuera, existiría un w_j tal que $L = L(G_j)$.

Teorema 4 IV

- Si la cadena define una gramática, entonces $L(G_i)$ es recursivo, y podemos usar el algoritmo de pertenencia del Teorema 3 para averiguar si $w_i \in L(G_i)$.
- Si no es así, entonces w_i pertenece a L.
- Pero L no es sensible al contexto.
- Si lo fuera, existiría un w_j tal que $L = L(G_j)$.
- Entonces podemos preguntar si w_j está en $L(G_j)$.

Teorema 4 IV

- Si la cadena define una gramática, entonces $L(G_i)$ es recursivo, y podemos usar el algoritmo de pertenencia del Teorema 3 para averiguar si $w_i \in L(G_i)$.
- Si no es así, entonces w_i pertenece a L.
- Pero L no es sensible al contexto.
- Si lo fuera, existiría un w_j tal que $L = L(G_j)$.
- Entonces podemos preguntar si w_j está en $L(G_j)$.
- Si asumimos que $w_j \in L(G_j)$, entonces, por definición, w_j no está en L.

Teorema 4 IV

- Si la cadena define una gramática, entonces $L(G_i)$ es recursivo, y podemos usar el algoritmo de pertenencia del Teorema 3 para averiguar si $w_i \in L(G_i)$.
- Si no es así, entonces w_i pertenece a L.
- Pero L no es sensible al contexto.
- Si lo fuera, existiría un w_j tal que $L = L(G_j)$.
- Entonces podemos preguntar si w_j está en $L(G_j)$.
- Si asumimos que $w_j \in L(G_j)$, entonces, por definición, w_j no está en L.
- Pero $L = L(G_j)$, entonces tenemos una contradicción.

Teorema 4 IV

- Si la cadena define una gramática, entonces $L(G_i)$ es recursivo, y podemos usar el algoritmo de pertenencia del Teorema 3 para averiguar si $w_i \in L(G_i)$.
- Si no es así, entonces w_i pertenece a L.
- Pero L no es sensible al contexto.
- Si lo fuera, existiría un w_j tal que $L = L(G_j)$.
- Entonces podemos preguntar si w_j está en $L(G_j)$.
- Si asumimos que $w_j \in L(G_j)$, entonces, por definición, w_j no está en L.
- Pero $L = L(G_j)$, entonces tenemos una contradicción.
- Por el contrario, si asumimos que $w_j \notin L(G_j)$, entonces, por definición, $w_j \in L$ y tenemos otra contradicción.

Teorema 4 IV

- Si la cadena define una gramática, entonces $L(G_i)$ es recursivo, y podemos usar el algoritmo de pertenencia del Teorema 3 para averiguar si $w_i \in L(G_i)$.
- Si no es así, entonces w_i pertenece a L.
- Pero L no es sensible al contexto.
- Si lo fuera, existiría un w_j tal que $L = L(G_j)$.
- Entonces podemos preguntar si w_j está en $L(G_j)$.
- Si asumimos que $w_j \in L(G_j)$, entonces, por definición, w_j no está en L.
- Pero $L = L(G_j)$, entonces tenemos una contradicción.
- Por el contrario, si asumimos que $w_j \notin L(G_j)$, entonces, por definición, $w_j \in L$ y tenemos otra contradicción.
- Por lo tanto, debemos concluir que L no es sensible al contexto.

Relación entre lenguajes recursivos y sensibles al contexto II

- El resultado del Teorema 4 indica que los autómatas acotados linealmente son de hecho menos poderosos que las máquinas de Turing, ya que sólo aceptan un subconjunto propio de los lenguajes recursivos.

Relación entre lenguajes recursivos y sensibles al contexto II

- Se sigue del mismo resultado que los autómatas acotados linealmente son más poderosos que los autómatas de pila.

Relación entre lenguajes recursivos y sensibles al contexto II

- Los lenguajes libres de contexto, que se generan mediante gramáticas libres de contexto, son un subconjunto de los lenguajes sensibles al contexto.

Relación entre lenguajes recursivos y sensibles al contexto II

- Como muestran varios ejemplos, son un subconjunto propio.

Relación entre lenguajes recursivos y sensibles al contexto II

- Debido a la equivalencia esencial de los autómatas acotados linealmente y los lenguajes sensibles al contexto por un lado, y los autómatas de pila y los lenguajes libres de contexto por el otro, vemos que cualquier lenguaje aceptado por un autómata de pila también es aceptado por algún autómata acotado linealmente, pero que hay lenguajes aceptados por algunos autómatas acotados linealmente para los que no hay autómatas de pila.

Ejercicios

Construya gramáticas sensibles al contexto para los siguientes lenguajes:

1. $L = \{a^{n+1}b^n c^{n-1} : n \geq 1\}$.
2. $L = \{a^n b^n a^{2n} : n \geq 1\}$.
3. $L = \{a^n b^m c^n d^m : n \geq 1, m \geq 1\}$.
4. $L = \{ww : w \in \{a, b\}^+\}$.
5. $L = \{a^n b^n c^n d^n : n \geq 1\}$.

La Jerarquía de Chomsky I

- Ahora nos hemos encontrado con una serie de familias de lenguajes, entre ellos los lenguajes recursivamente enumerables (L_{RE}), los lenguajes sensibles al contexto (L_{CS}), los lenguajes libres de contexto (L_{CF}) y los lenguajes regulares (L_{REG}).

La Jerarquía de Chomsky I

- Una forma de mostrar la relación entre estas familias es mediante la **Jerarquía de Chomsky**.

La Jerarquía de Chomsky I

- Noam Chomsky, uno de los fundadores de la teoría de lenguajes formales, proporcionó una clasificación inicial con cuatro tipos de lenguajes, del tipo 0 al tipo 3.

La Jerarquía de Chomsky I

- Esta terminología original ha persistido y se encuentran frecuentes referencias a ella, pero los tipos numéricos son en realidad nombres diferentes para las familias de lenguajes que hemos estudiado.

La Jerarquía de Chomsky I

- Los lenguajes de tipo 0 son los generados por gramáticas sin restricciones, es decir, los lenguajes enumerables recursivamente.

La Jerarquía de Chomsky I

- El tipo 1 consta de los lenguajes sensibles al contexto, el tipo 2 consta de los lenguajes libres de contexto y el tipo 3 consta de los lenguajes regulares.

La Jerarquía de Chomsky II

- Como hemos visto, cada familia de lenguajes de tipo i es un subconjunto propio de la familia de tipo $i-1$.

La Jerarquía de Chomsky II

- La figura 1 muestra la jerarquía de Chomsky original.

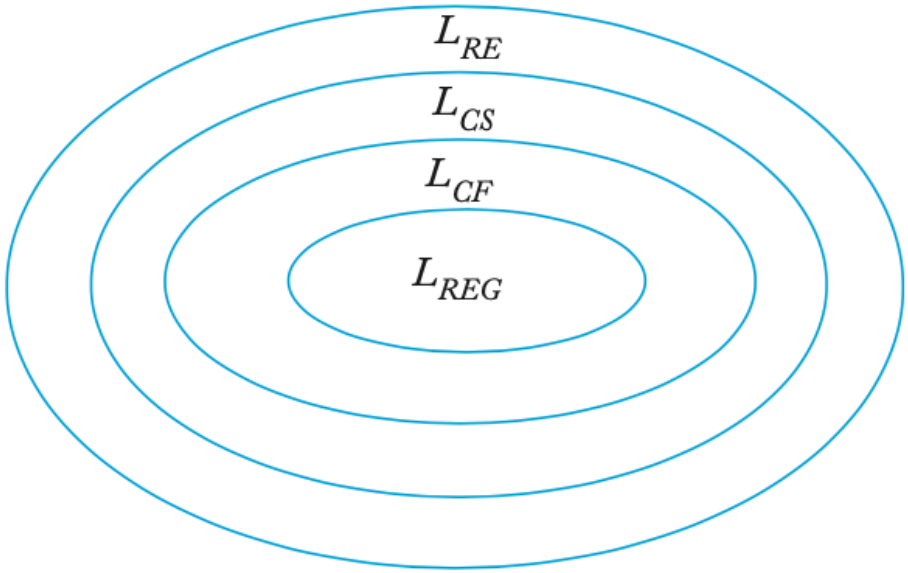

Figure 1: Jerarquía de Chomsky original.

La Jerarquía de Chomsky III

- También hemos conocido otras familias de lenguajes que pueden encajar en esta imagen.

La Jerarquía de Chomsky III

- Incluyendo las familias de lenguajes deterministas libres de contexto (L_{DCF}) y lenguajes recursivos (L_{REC}), llegamos a la jerarquía extendida que se muestra en la Figura 2.

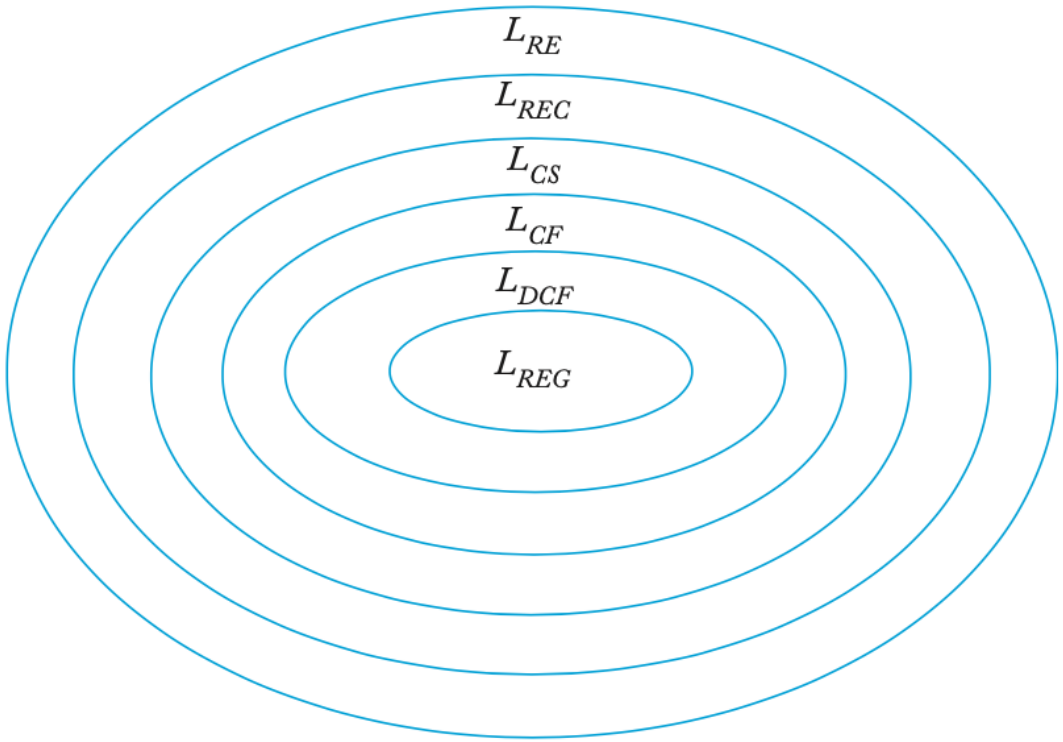

Figure 2: Jerarquía de Chomsky extendida.

La Jerarquía de Chomsky IV

- Para resumir, hemos explorado las relaciones entre varias familias de lenguajes y sus autómatas asociados.

La Jerarquía de Chomsky IV

- Al hacerlo, establecimos una jerarquía de lenguajes y clasificamos a los autómatas por su poder como aceptadores de lenguajes.

La Jerarquía de Chomsky IV

- Las máquinas de Turing son más poderosas que los autómatas acotados linealmente.

La Jerarquía de Chomsky IV

- Estos, a su vez, son más poderosos que los autómatas de pila.

La Jerarquía de Chomsky IV

- En la parte inferior de la jerarquía se encuentran los aceptadores finitos, con los que comenzamos nuestro estudio.