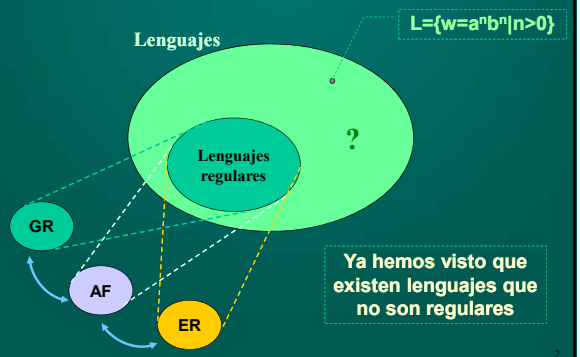
Teoría de la Computabilidad

Módulo 2:
Lenguajes Libres de Contexto y Sensibles al Contexto

Departamento de Cs. e Ing. de la Computación
Universidad Nacional del Sur
Bahía Blanca, Argentina

1

Lenguajes, gramáticas y autómatas



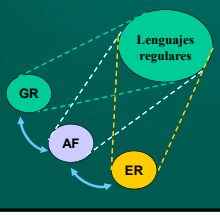
Ya hemos visto que existen lenguajes que no son regulares

2

Lenguajes, gramáticas y autómatas

Con las herramientas de las que disponemos, no podemos construir un autómata que reconozca las cadenas de la forma a^nb^n, ni una gramática para producir estas cadenas...

Evidentemente hemos alcanzado un límite en nuestro poder de cómputo...



Decimos que encontramos un "límite" pues conocemos una tarea que no podemos realizar.

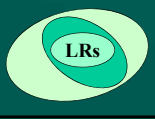
3

Lenguajes, gramáticas y autómatas

Es lógico entonces preguntarnos...
¿tiene el lenguaje $L=\{a^nb^n|n>0\}$ una gramática que lo genere?
¿y un autómata que reconozca sus cadenas?

Veremos que así como existen lenguajes regulares, existen otros tipos de lenguajes, con sus respectivas gramáticas y autómatas.

De esta forma expandimos nuestro poder de cómputo para poder trabajar con estos lenguajes.



¿cuál será el límite definitivo?
¿existe?
¿hay algo que nunca podremos computar?

4

Leng. Libre de Contexto (LLC) y Sensibles al Contexto (LSC)

Por ejemplo, un AF (o una ER, o una GR): ¿podrá reconocer elementos tales como una expresión aritmética en un lenguaje de programación?

Veremos a continuación lenguajes algo más complicados que los lenguajes regulares.

Para ello estudiaremos otras clases de gramáticas, que surgen de restringir las reglas de producción permitidas, usando distintos criterios.

Estas gramáticas nos permiten generar lenguajes más complejos.

5

Gramáticas Libres de Contexto (GLC)

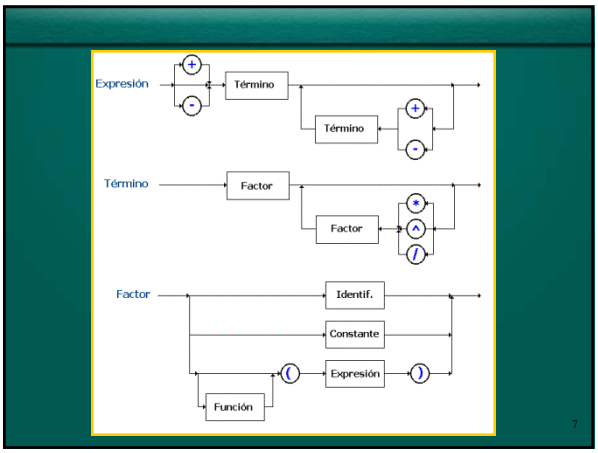
Las gramáticas libres de contexto (GLC) son muy importantes para describir sintaxis de lenguajes de programación.

Se dicen "*libres de contexto*" pues las reglas de producción prescinden del entorno: dicen cómo reemplazar un no terminal sin hacer referencia al contexto en el que se encuentra:

La parte izquierda de $\alpha \rightarrow \beta$ es siempre un símbolo no terminal.

Ej: los diagramas de Conway de Pascal pueden asociarse a una GLC.

6

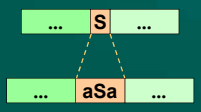


Gramáticas Libres de Contexto (GLC)

- **Gramáticas con libertad de contexto**

$P = \{ \; S \rightarrow a\,S\,a, \;\; S \rightarrow b\,S\,b, \;\; S \rightarrow c \; \}$

De lado izquierdo, siempre UN símbolo no terminal



No importa el "contexto" en el que se encuentra S

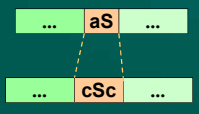
Se reemplaza el no terminal por la cadena indicada en la regla de producción

Gramáticas Sensibles al Contexto (GSC)

- **Gramáticas con sensibilidad al contexto**

$P = \{ \; aS \rightarrow cSc, \;\; S \rightarrow aSb, \;\; S \rightarrow c \; \}$

De lado izquierdo, símbolos terminales y/o no terminales



Se hace referencia al "contexto" en el que se encuentra S

Se reemplaza el no terminal y su contexto por la cadena indicada en la regla de producción

Libre de Contex. vs. Sensible al Contex.

- **Gramáticas con libertad de contexto**

$P = \{ S \rightarrow a\,S\,a, \;\; S \rightarrow b\,S\,b, \;\; S \rightarrow c \}$

$S \xrightarrow{G} aSa \xrightarrow{G} abSba \xrightarrow{G} abcba$

- **Gramáticas con sensibilidad al contexto**

$P = \{ S \rightarrow aSb, \;\; aS \rightarrow cSc, \;\; S \rightarrow c \}$

$S \xrightarrow{G} aSb \xrightarrow{G} cScb \xrightarrow{G} cccb$

Clasificación de gramáticas de Chomsky

Entre 1950 y 1960, el lingüista norteamericano Noam Chomsky desarrolló una clasificación de las gramáticas. Su idea: modelar los lenguajes naturales, con miras a traducción automática

Noam Chomsky

Gramáticas (repaso)

Recordemos....

Def.: Una **gramática estructurada por frases (GEF)** es una 4-upla **G=(Vn,Vt,S,P)**, donde

V_n = conj.finito de **símbolos no terminales** (o auxiliares)

V_t = conj. finito de **símbolos terminales**

S = **símbolo inicial** ,**S** $\in V_n$

P = conj.finito de **reglas de producción** $\alpha \rightarrow \beta$.

Estudiaremos clases de gramáticas que surgen al tipificar las reglas de producción permitidas.

Gramáticas: tipos 0,1,2 y 3

Def.: Una **gramática** $G=(V_n,V_t,S,P)$. Diremos que G es de:

Tipo 0: si la única restricción en cada regla $\alpha\rightarrow\beta$ es $\alpha \in (V_n \cup V_t)^* V_n (V_n \cup V_t)^*$

Tipo 1: si es de tipo 0, y se verifica $long(\alpha) \leq long(\beta)$

Tipo 2: si es de tipo 1, y en cada regla $\alpha\rightarrow\beta$, se cumple que $\alpha \in V_n$

Tipo 3: si es de tipo 2, y en cada regla $\alpha\rightarrow\beta$, se cumple que $\alpha \in V_n$, y $\beta \in V_t$ o bien $\beta \in (V_t \bullet V_n)$

13

Gramáticas: tipos 0,1,2 y 3

Restricciones para $\alpha\rightarrow\beta$	Gramática
$\alpha \in (V_n \cup V_t)^* V_n (V_n \cup V_t)^*$	Tipo 0
$long(\alpha) \leq long(\beta)$	Tipo 1
$\alpha \in V_n$	Tipo 2
$\beta \in V_t$ o bien $\beta \in (V_t \bullet V_n)$	Tipo 3

14

Gramáticas tipos 3 y 2: características

Las **gramáticas tipo 3 o regulares** generan los lenguajes especificados por las ERs.

$S\rightarrow 1$, $S\rightarrow 0S$, $S\rightarrow 1S$

Las **gramáticas tipo 2 o libres de contexto** permiten, en la izquierda de sus producciones, un único simbolo $\alpha \in V_n$. El contexto no influye al aplicar una regla de producción.

$S\rightarrow a$, $S\rightarrow b$, $S\rightarrow aSa$, $S\rightarrow bSb$

15

Gramáticas tipo 1: características

Las **gramáticas tipo 1 o sensibles al contexto** permiten producciones donde a la izquierda puede haber varios símbolos ("contexto") rodeando a un no terminal.

Si existe una regla $C_iA\ C_d \rightarrow C_i'\beta\ C_d'$, donde

$A \in (V_n \cup V_t)^* \bullet V_n \bullet (V_n \cup V_t)^*$, y

$C_i, C_d, \beta, C_i', C_d' \in (V_n \cup V_t)^*$

Aquí C_i y C_d forman el contexto a izq. y der. al cual es sensible el no terminal A. En otras palabras, A puede ser reemplazado por β cuando esté entre C_i y C_d .

16

Ejemplo 1

Defina una gramática para generar las cadenas de

$L=\{wcz \mid w \in \{a,b\}^*, z=\text{reverso de } w\}$.

Ejemplos: aacaa, bacab $\in$L.

$G=(V_n,V_t,S,P)$,

$V_n=\{S,T_1,T_2,T_3\}$, $V_t=\{a,b,c\}$,

$P=\{ S\rightarrow a\ S\ a,\ \ S\rightarrow b\ S\ b,\ \ S\rightarrow c\ \}$

Esta es una gramática libre de contexto. La parte izq. de cada regla es un V_n y la parte derecha pertenece a $(V_n \cup V_t)^*$

17

Ejemplo 2

Supongamos querer una gramática para generar las cadenas de

$L=\{a^nb^nc^n \mid n>0\}$.

Ejemplos: aabbcc, abc $\in$L.

$G=(V_n, V_t, S, P)$,

$V_n=\{S, A, C\}$, $V_t=\{a,b,c\}$,

$P=\{ S\rightarrow abc \mid aAbc$, $A\rightarrow abC \mid aAbC$,

$Cb\rightarrow bC$, $Cc\rightarrow cc\}$

Esta es una gramática sensible al contexto (ej: el no terminal C sólo puede ser reescrito como c si en alguna cadena aparece una c adelante).

18

Ejemplo 3

Supongamos querer una gramática para generar las cadenas de $L=\{w=a^{2n} | n \geq 0\}$ (Ej: aa, aaaa, ...)

$G=(V_n, V_t, S, P)$,

$V_n=\{S, X, Y, Z\}$, $V_t=\{a\}$,

$P=\{ \; S \rightarrow \lambda, \quad S \rightarrow YXXY, \quad YX \rightarrow YZ,$

$ZX \rightarrow XZ, \quad ZY \rightarrow XXXY, \quad X \rightarrow a, \quad Y \rightarrow \lambda \}$

Esta es una gramática donde long(α) ≤long(β) se verifica casi para todas las reglas, pero no para $S \rightarrow \lambda$ e $Y \rightarrow \lambda$. Luego ni siquiera es tipo 1, sino de tipo 0.

19

Cadena nula: un caso especial...

En las gramáticas de tipo 1 y 2, debía respetarse que $\beta \in (V_t \cup V_n)^*$, y esto implica como caso particular que $\beta = \lambda$

Recordemos que las reglas de tipo $A \rightarrow \lambda$ son reglas borradoras, y su admisión dificulta predecir el comportamiento de los lenguajes afectados.

Convención usual adoptada: aceptar reglas $S \rightarrow \lambda$ sólo si S no aparece en la parte derecha de las producciones. Luego $S \rightarrow \lambda$ solo es usado en un único paso de una derivación (el inicial).

Nota: la jerarquía de Chomsky indica que las gramáticas tipo 1 no pueden tener reglas borradoras (y por ende tampoco las de tipo 2 y tipo 3).

20

Ejemplo con regla borradora

Consideremos una gramática para

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

$G=(V_n, V_t, S, P)$,

$V_n=\{S,R\}$, $V_t=\{a,b\}$,

$P=\{S \rightarrow \lambda, \; S \rightarrow R, \quad R \rightarrow a\,R\,b, \quad R \rightarrow ab\}$

Esta es una gramática **libre de contexto**. Si bien hay una regla borradora $S \rightarrow \lambda$, ésta respeta la condición antes mencionada.

21

Teorema de la cadena nula en G^{SC}

Teo. de cadena nula: Si L es un lenguaje generado por una gramática sensible al contexto, entonces $L-\{\lambda\}$ y $L \cup \{\lambda\}$ también lo es.

22

Lenguajes y gramáticas

Las gramáticas generan lenguajes, y al clasificar las gramáticas obtenemos también una clasificación de lenguajes

¿De qué tipo es un lenguaje L?

Debe distinguirse entre el **tipo de lenguaje** y la **gramática asociada** a él.

Buscaremos una gramática del **mayor tipo posible** para determinar la pertenencia de un lenguaje a una cierta clase.

Analogía: ¿en qué cjto. ubicamos al 2?

23

Tres gramáticas alternativas - Ejemplo

Sea L={program} (la palabra reservada de Pascal). Veamos gramáticas alternativas:

$G_1=(V_n, V_t, S, P)$, $V_n=\{S,A,B\}$, $V_t=\{p,r,o,g,r,a,m\}$,

$P=\{S \rightarrow AB, \quad A \rightarrow pro, \quad B \rightarrow gram\}$

$G_2=(V_n, V_t, S, P)$, $V_{n1}=\{S\}$, $V_t=\{p,r,o,g,r,a,m\}$,

$P_1=\{S \rightarrow program\}$

$G_3=(V_n, V_t, S, P)$, $V_n=\{S,R,O,G,R_2,A,M\}$,

$V_t=\{p,r,o,g,r,a,m\}$,

$P=\{ \; S \rightarrow pR, R \rightarrow rO, O \rightarrow oG, G \rightarrow gR_2, R_2 \rightarrow rA,$

$A \rightarrow aM, M \rightarrow m \quad \}$

24

¡Tres gramáticas alternativas!

Sea $L=\{w=a^{2n} \mid n>0\}$.

$G_1=(V_{n1},V_{t1},S,P)$, $V_{n1}=\{S,X,Y,Z\}$, $V_{t1}=\{a\}$,
$P_1=\{ S\rightarrow \lambda,\ S\rightarrow YXXY,\ YX\rightarrow YZ,\ ZX\rightarrow XZ,$
$ZY\rightarrow XXXY,\ X\rightarrow a,\ Y\rightarrow \lambda\}$

$G_2=(V_{n1},V_{t1},S,P_1)$, $V_{n2}=\{S\}$, $V_{t2}=\{a\}$,
$P_2=\{ S\rightarrow aa,\ S\rightarrow aSa\}$

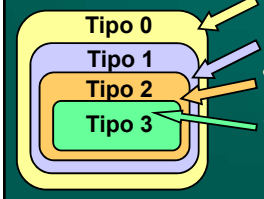
$G_3=(V_{n2},V_{t2},S,P_3)$, $V_{n3}=\{S\}$, $V_{t3}=\{a\}$,
$P_3=\{ S\rightarrow aaS,\ S\rightarrow aa\}$

25

Lenguaje de Tipo i

Def: Sea L un lenguaje.Diremos que **L es de tipo i** si existe una gramática G de tipo i tq. L =L(G), con i es maximal en la jerarquía de Chomsky.

$L=\{a^nb^n \mid n \geq 0\}$
¿de qué tipo es?



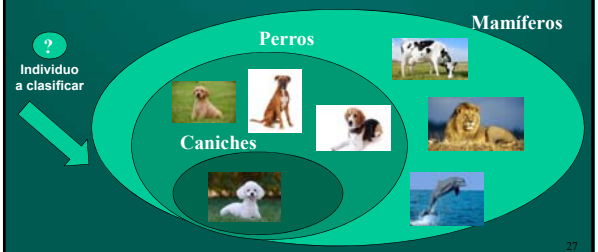
¿Lo genero con G de tipo 0?
¿Lo genero con G de tipo 1?
¿Lo genero con G de tipo 2?
¿Lo genero con G de tipo 3?

$L=\{a^nb^n \mid n \geq 0\}$ es tipo 2

26

Analogía: clasificando un ser vivo

Imaginemos que queremos clasificar un animal según sus características y arrancamos de lo más específico a lo más general (ej. primero vemos si es un caniche; si no lo es, vemos si es un perro, y así siguiendo sucesivamente).



27

Gramáticas Libres de Contexto

Def.: Una **gramática libre de contexto** G=(Vn,Vt,S,P), donde

V_n = conj.finito de **símbolos no terminales**
V_t = conj. finito de **símbolos terminales**
S = **símbolo inicial** ,$S \in V_n$
P = conj.finito de **reglas de producción** $\alpha\rightarrow\beta$, donde $\alpha \in V_n$ y $\beta \in (V_n \cup V_t)^*$

28

Ejemplo

Sea G una gramática
libre de contexto
$G_1=(V_n,V_t,O,P)$,
$V_n=\{O,A,S,V,F\}$,

$V_t=\{$robot, naranja, pelota , patea, ataja$\}$,

$P=\{ O\rightarrow FVF,\ F\rightarrow S,\ F\rightarrow SA,\ A\rightarrow$naranja,
$S\rightarrow$pelota , $S\rightarrow$ robot, $V\rightarrow$ ataja, $V\rightarrow$ patea $\}$

Derivación:

$O \overset{G}{\Rightarrow} FVF \overset{G}{\Rightarrow} SVF \overset{G}{\Rightarrow} SVS \overset{G}{\Rightarrow}$ robot VS $\overset{G}{\Rightarrow}$

$\overset{G}{\Rightarrow}$ robot patea S $\overset{G}{\Rightarrow}$ robot patea pelota

29

Capacidades de las G^{LC}

Con GLCs, podemos resolver problemas tales como:

- Cadenas de paréntesis balanceados en expresiones aritméticas
- Apareamiento de begin-ends
- Correspondencia if-then-else-end if

30

Ejemplo

Consideremos el alfabeto $V_t = \{(,),+,*,x_1,x_2\}$,

$G = (V_n, V_t, E, P)$,

$V_n = \{E, T, F\}$,

$P = \{E \rightarrow E+T,\ E \rightarrow T,\ T \rightarrow T*F,\ T \rightarrow F,\ F \rightarrow (E),\ F \rightarrow x_1,\ F \rightarrow x_2\}$

Derivación para $(x_1 * x_2 + x_1) * (x_1 + x_2)$

$E \xrightarrow{G} T \xrightarrow{G} T*F \xrightarrow{G} F*F \xrightarrow{G} (E)*F \xrightarrow{G} (E+T)*F \xrightarrow{G} (T+T)*F \xrightarrow{G} (T*F+T)*F \xrightarrow{G} (F*F+T)*F \xrightarrow{G} (x_1*F+T)*F \xrightarrow{G} (x_1*x_2+T)*F \xrightarrow{G+} (x_1*x_2+x_1)*(x_1+x_2)$

31

Aplicaciones de las G^{LC}

Aplicaciones de Gramáticas Libres de Contexto:

- Especificación de la sintaxis de los lenguajes de programación.
- Generadores de analizadores YACC.
- Lenguajes de marcado tipo HTML
- XML – DTD (describen la "semantica" del texto que se especifica en un HTML)

32