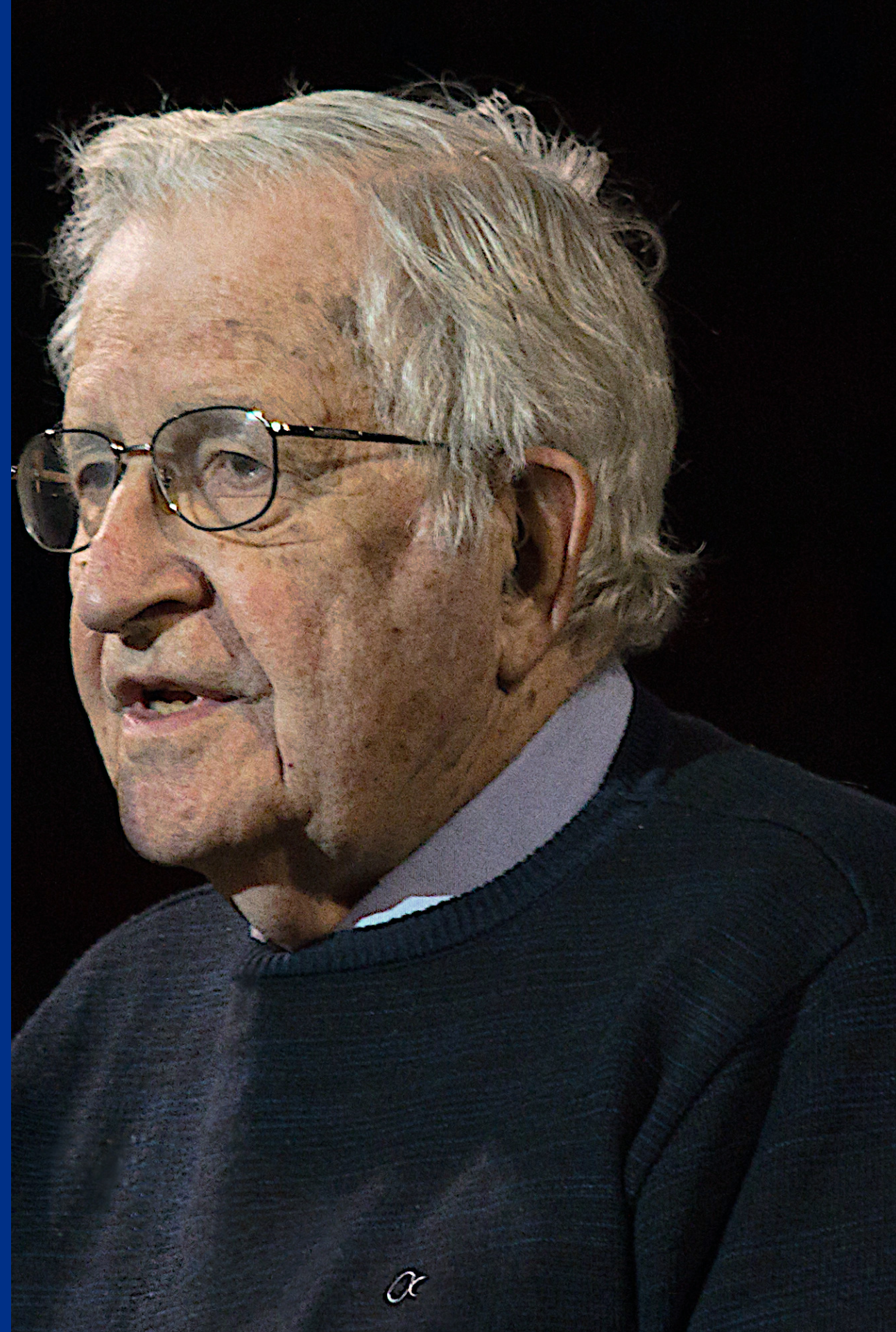


Gramáticas libres de contexto



UNIVERSIDAD
DE CHILE

Arturo Merino



Recuerdo / Plan

Recuerdo/Plan

- **Recuerdo:**

- Lenguajes no regulares
- El lema de bombeo

- **Plan:**

- Gramáticas libres de contexto
- Lenguajes libre de contexto
- Relación con los lenguajes regulares
- Propiedades de cerradura

Gramáticas libre de contexto

Gramática libre de contexto

- **Objetivo del curso:** formalizar el modelo de computación.
 - Logramos entender poder y límites de los AF/ER.
 - Muy débiles como modelo general de computación.
- Chequear texto *sintácticamente*.
 - url's, RUT's, tokenización, etc $\rightarrow$ ER.
 - ¿Chequear si un programa matchea la sintáxis? / ¿Parser?
 - ¿Chequear si un programa tiene paréntesis bien balanceados?
 - **Imposible** via ER $\leftarrow$ no pueden chequear que $\#($ sea igual a $\#)$.
- **Extendamos** dichos modelos.
- ER $\leftarrow$ pattern-matching sin variables.
- Gramática = ER + variables.

Variables y gramáticas

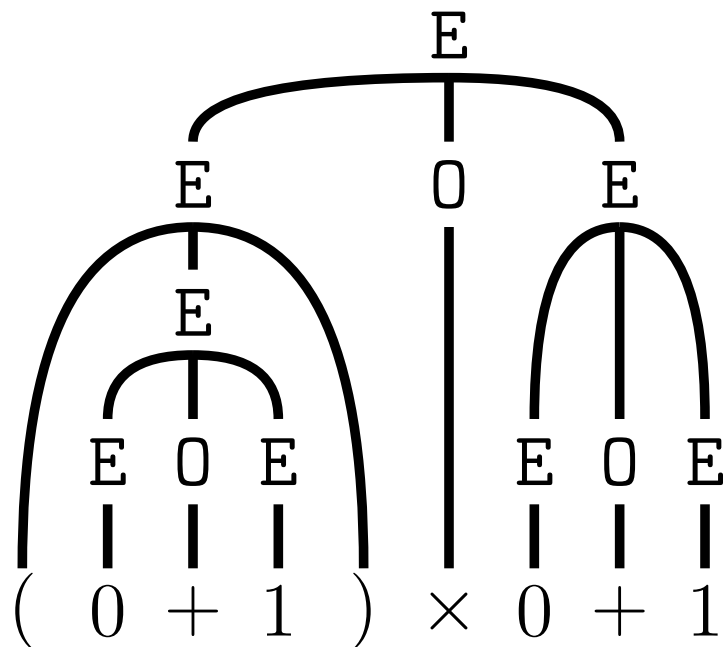
- Gramática: **variables y reglas de reemplazo.**
- Gramática que genere expresiones aritméticas

$$\text{Exp} \rightarrow \text{Exp Op Exp}.$$
$$\text{Exp} \rightarrow (\text{Exp})$$

Exp	$\rightarrow 0$	1
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0
12	0	0
13	0	0
14	0	0
15	0	0
16	0	0
17	0	0
18	0	0
19	0	0
20	0	0
21	0	0
22	0	0
23	0	0
24	0	0
25	0	0
26	0	0
27	0	0
28	0	0
29	0	0
30	0	0
31	0	0
32	0	0
33	0	0
34	0	0
35	0	0
36	0	0
37	0	0
38	0	0
39	0	0
40	0	0
41	0	0
42	0	0
43	0	0
44	0	0
45	0	0
46	0	0
47	0	0
48	0	0
49	0	0
50	0	0
51	0	0
52	0	0
53	0	0
54	0	0
55	0	0
56	0	0
57	0	0
58	0	0
59	0	0
60	0	0
61	0	0
62	0	0
63	0	0
64	0	0
65	0	0
66	0	0
67	0	0
68	0	0
69	0	0
70	0	0
71	0	0
72	0	0
73	0	0
74	0	0
75	0	0
76	0	0
77	0	0
78	0	0
79	0	0
80	0	0
81	0	0
82	0	0
83	0	0
84	0	0
85	0	0
86	0	0
87	0	0
88	0	0
89	0	0
90	0	0
91	0	0
92	0	0
93	0	0
94	0	0
95	0	0
96	0	0
97	0	0
98	0	0
99	0	0
100	0	0

Op $\rightarrow$ + | - | $\times$ | /

- **Ej:** $(0 + 1) \times 0 + 1, (0 + \times)0$.
- Derivación en gramáticas $\leftrightarrow$ producción en recursión
- $E \Rightarrow EOE \Rightarrow EOEEOE \Rightarrow (E)OEEOE \Rightarrow (EOE)OEEOE \xRightarrow{*} (0 + 1) \times 0 + 1$



Definición formal

Def.

Una *gramática libre de contexto (GLC)* es una 4-tupla (V, Σ, R, S) , donde

- V es un conjunto finito llamado *variables*
- Σ es un conjunto finito, disjunto de V , llamado los *terminales*.
- $R \subseteq V \times (V \cup \Sigma)^*$ es un conjunto finito de *reglas*, denotadas de la forma $A \rightarrow w$ con $A \in V$ y $w \in (V \cup \Sigma)^*$.
- $S \in V$ es la *variable de partida*.

- Usualmente descrita por un *listado de reglas*

$$\text{Exp} \rightarrow \text{Exp Op Exp.}$$
$$\text{Exp} \rightarrow (\text{Exp})$$
$$\text{Exp} \rightarrow 0 \mid 1$$
$$\text{Op} \rightarrow + \mid - \mid \times \mid /$$

- Primera variable en aparecer es la inicial.
- Uso de $|$ para abreviar

¿Cómo computa/deriva palabras?

Def.

Sea $G = (V, \Sigma, R, S)$ una GLC.

- Si $A \rightarrow w \in R$, diremos que uAv *produce* $uwwv$, escrito $uAv \Rightarrow uwwv$.
- Diremos que u *deriva* v si existe una secuencia $u_1, \dots, u_k$ tal que
$$u \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \dots \Rightarrow u_k \Rightarrow v$$
- Si u deriva v , lo anotamos por $u \xRightarrow{*} v$.
- La gramática G *deriva* $w \in \Sigma^*$ si $S \xRightarrow{*} w$.

- Las gramáticas describen lenguajes.

Def.

Sea $G = (V, \Sigma, R, S)$ una GLC. El *lenguaje* de G es

$$\mathcal{L}(G) = \{w \in \Sigma^* : G \text{ deriva } w\}$$

Entendiendo gramáticas vía sus lenguajes

- **Ej:** ¿Qué lenguajes describen las siguientes gramáticas?

- $A \rightarrow 0A1 \mid B$
 $B \rightarrow \varepsilon$

- $A \rightarrow (A) \mid AA \mid \varepsilon$

- $A \rightarrow 0A0 \mid 1A1 \mid 0 \mid 1 \mid \varepsilon$

- $\text{ORACIÓN} \rightarrow \text{SUJETO PREDICADO}$

- $\text{SUJETO} \rightarrow \text{ARTÍCULO SUSTANTIVO}$

- $\text{ARTÍCULO} \rightarrow \text{el} \mid \text{un}$

- $\text{SUSTANTIVO} \rightarrow \text{niño} \mid \text{balón} \mid \text{cielo}$

- $\text{PREDICADO} \rightarrow \text{juega} \mid \text{mira SUJETO} \mid \text{toca SUJETO} \mid \text{juega con SUJETO}$

Lenguajes libre de contexto

Def.

Sea $L \subseteq \Sigma^*$.

Diremos que L es *libre de contexto* si existe una GLC con terminales en Σ tal que $\mathcal{L}(\Sigma) = L$.

- ¿Cómo probar que un lenguaje es libre de contexto?
- ¿Cómo diseñar GLC?
 - Recordar los tips para ER: separar en bloques clave + patrones forzados.
 - Usar la recursión para llevar cuentas
 - Usar distintas variables para distintas “fases” de la producción.
- **Ej:**
 - $L_1 = \{a^i b^j \mid i \geq j\}$
 - $L_2 = \{a^i b^j c^j d^i \mid i, j \geq 0\}$
 - $L_3 = \{a^i \# a^j \# a^{i+j} \mid i, j \geq 0\}$
 - $L_4 = \{w \in \{0, 1\}^* : w \text{ tiene igual cantidad de 0's que de 1's}\}$

Relación con los lenguajes regulares

- Las GLC se sienten más poderosas que las ER

Teo.

Todo lenguaje regular es libre de contexto.

Además, hay lenguajes no regulares que son libres de contexto.

- **Dem:**

- ¿Qué nos dice esto de las ER v/s las GLC?
- GLC tienen más poder de expresividad/computo.
- El poder de las **variables**.

Propiedades de cerradura

- ¿Cómo se comportan ser libre de contexto con nuestras operaciones?

Teo.

Sean $L_1, L_2 \subseteq \Sigma^*$ dos lenguajes libres de contexto.

Los siguientes son libres de contexto:

- $L_1 L_2, L_1 \cup L_2$ y $(L_1)^*$

- Libre de contexto cerrados bajo: concatenación, unión y estrella.
- **Dem:**
- **Ojo:** no son cerrados bajo intersección.
- ¿Complemento?

Dudas/Preguntas

Wrap-Up

- **Hoy:**
 - Gramáticas libres de contexto
 - Lenguajes libre de contexto
 - Relación con los lenguajes regulares
 - Propiedades de cerradura
- **Prox. clase:**
 - Automatas de pila