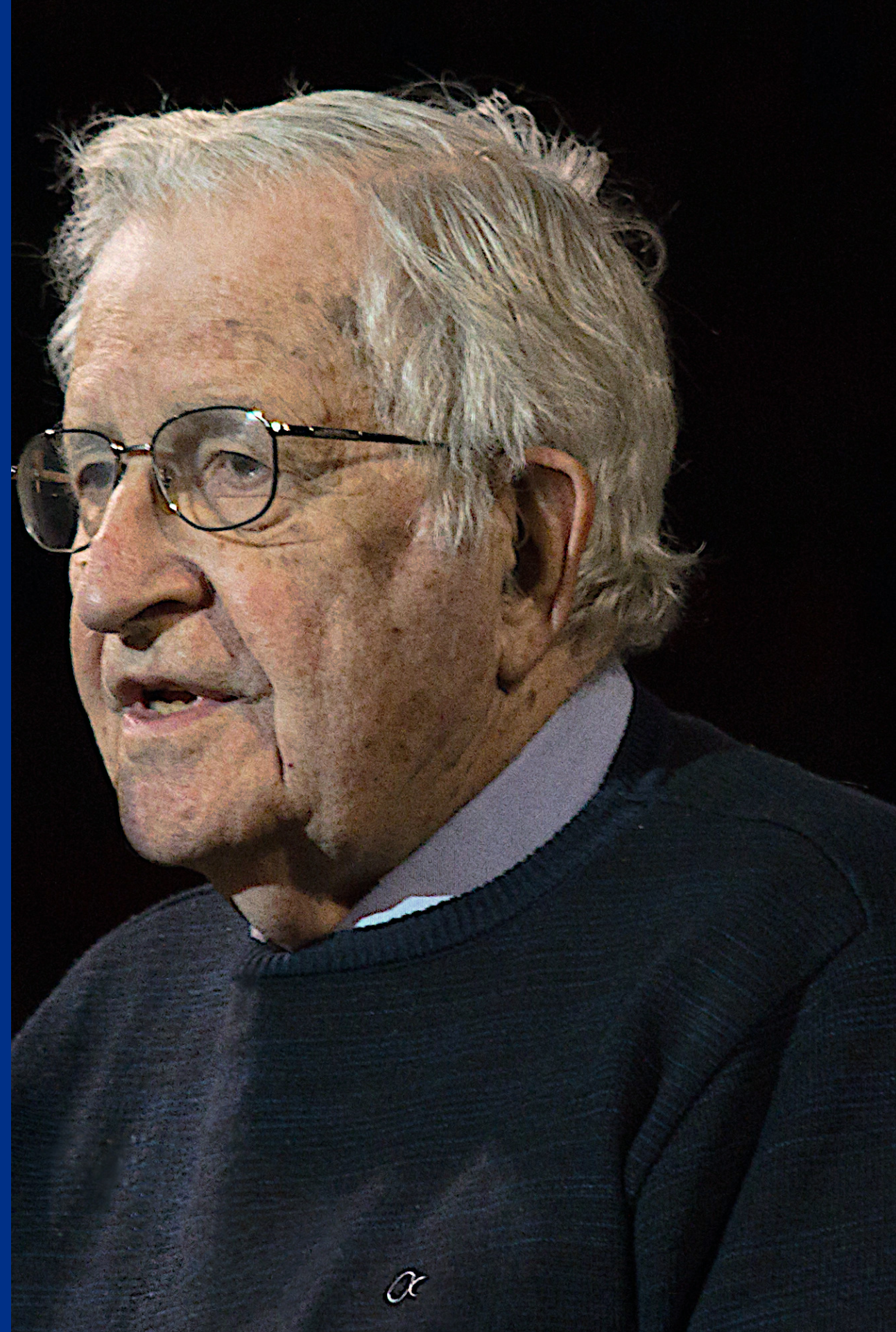


Gramáticas libres de contexto



UNIVERSIDAD
DE CHILE

Arturo Merino



Recuerdo / Plan

■ Recuerdo/Plan

- **Recuerdo:**

Recuerdo/Plan

- **Recuerdo:**
 - Lenguajes no regulares

- **Recuerdo:**
 - Lenguajes no regulares
 - El lema de bombeo

Recuerdo/Plan

- **Recuerdo:**
 - Lenguajes no regulares
 - El lema de bombeo
- **Plan:**

Recuerdo/Plan

- **Recuerdo:**

- Lenguajes no regulares
- El lema de bombeo

- **Plan:**

- Gramáticas libres de contexto

Recuerdo/Plan

- **Recuerdo:**

- Lenguajes no regulares
- El lema de bombeo

- **Plan:**

- Gramáticas libres de contexto
- Lenguajes libre de contexto

Recuerdo/Plan

- **Recuerdo:**

- Lenguajes no regulares
- El lema de bombeo

- **Plan:**

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

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.

Gramática libre de contexto

- **Objetivo del curso:** formalizar el modelo de computación.
 - Logramos entender poder y límites de los AF/ER.

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.

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*.

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.

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 sintaxis? / ¿Parser?

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 sintaxis? / ¿Parser?
 - ¿Chequear si un programa tiene paréntesis bien balanceados?

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 $\#)$.

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 sintaxis? / ¿Parser?
 - ¿Chequear si un programa tiene paréntesis bien balanceados?
 - **Imposible** via ER $\leftarrow$ no pueden chequear que $\#($ sea igual a $\#)$.
- **Extendamos** dichos modelos.

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 sintaxis? / ¿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 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 sintaxis? / ¿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**.

Variables y gramáticas

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

Variables y gramáticas

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

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

Variables y gramáticas

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

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

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

Variables y gramáticas

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

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

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

$\text{Exp} \rightarrow 0 \mid 1$

Variables y gramáticas

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

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

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

$\text{Exp} \rightarrow 0 \mid 1$

$\text{Op} \rightarrow + \mid - \mid \times \mid /$

Variables y gramáticas

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

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

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

$\text{Exp} \rightarrow 0 \mid 1$

$\text{Op} \rightarrow + \mid - \mid \times \mid /$

- **Ej:** $(0 + 1) \times 0 + 1, (0 + \times)0.$

Variables y gramáticas

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

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

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

$\text{Exp} \rightarrow 0 \mid 1$

$\text{Op} \rightarrow + \mid - \mid \times \mid /$

- **Ej:** $(0 + 1) \times 0 + 1, (0 + \times)0.$
- Derivación en gramáticas $\leftrightarrow$ producción en recursión

Variables y gramáticas

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

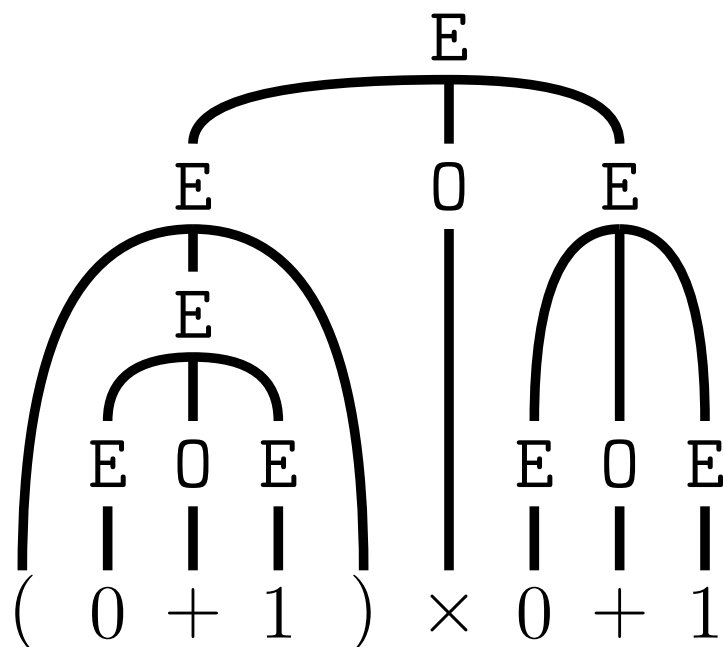
$\text{Exp} \rightarrow \text{Exp Op Exp}.$

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

$\text{Exp} \rightarrow 0 \mid 1$

$\text{Op} \rightarrow + \mid - \mid \times \mid /$

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



Definición formal

Def.

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

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*

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*.

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 $w \in (V \cup \Sigma)^*$.

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*.

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*

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*

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

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.

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.

¿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 $uAv \Rightarrow uwwv$.

¿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

¿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

$$u \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \dots \Rightarrow u_k \Rightarrow v$$

¿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 $u \Rightarrow^* v$.

¿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 $u \Rightarrow^* v$.
- La gramática G *deriva* $w \in \Sigma^*$ si $S \Rightarrow^* w$.

¿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 $u \Rightarrow^* v$.
- La gramática G *deriva* $w \in \Sigma^*$ si $S \Rightarrow^* w$.

- Las gramáticas describen lenguajes.

¿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

¿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 $u \Rightarrow^* v$.
- La gramática G *deriva* $w \in \Sigma^*$ si $S \Rightarrow^* w$.

- 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?

Entendiendo gramáticas vía sus lenguajes

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

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

Entendiendo gramáticas vía sus lenguajes

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

- $A \rightarrow 0A1 \mid B$

- $B \rightarrow \varepsilon$

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

Entendiendo gramáticas vía sus lenguajes

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

- $A \rightarrow 0A1 \mid B$

- $B \rightarrow \varepsilon$

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

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

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 $L \subseteq \Sigma^*$.

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 $\mathcal{L}(\Sigma) = L$.

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?

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?

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.

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

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.

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:

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:**
 - $L_1 = \{a^i b^j \mid i \geq j\}$

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:**
 - $L_1 = \{a^i b^j \mid i \geq j\}$
 - $L_2 = \{a^i b^j c^j d^i \mid i, j \geq 0\}$

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:**
 - $L_1 = \{a^i b^j \mid i \geq j\}$
 - $L_2 = \{a^i b^j c^j d^i \mid i, j \geq 0\}$
 - $L_3 = \{a^i \# a^j \# a^{i+j} \mid i, j \geq 0\}$

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

Relación con los lenguajes regulares

- Las GLC se sienten más poderosas que las ER

Teo.

Todo lenguaje regular es libre de contexto.

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.

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:**

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?

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.

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?

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.

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:

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:

$$\circ L_1 L_2, L_1 \cup L_2 \text{ y } (L_1)^*$$

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.

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:

$$\circ L_1 L_2, L_1 \cup L_2 \text{ y } (L_1)^*$$

- Libre de contexto cerrados bajo: concatenación, unión y estrella.
- **Dem:**

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:

$$\circ L_1 L_2, L_1 \cup L_2 \text{ y } (L_1)^*$$

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

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:

$$\circ L_1 L_2, L_1 \cup L_2 \text{ y } (L_1)^*$$

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

Dudas/Preguntas

Wrap-Up

- **Hoy:**

Wrap-Up

- **Hoy:**
 - Gramáticas libres de contexto

Wrap-Up

- **Hoy:**
 - Gramáticas libres de contexto
 - Lenguajes libre de contexto

Wrap-Up

- **Hoy:**
 - Gramáticas libres de contexto
 - Lenguajes libre de contexto
 - Relación con los lenguajes regulares

Wrap-Up

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

Wrap-Up

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

Wrap-Up

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