

INF155 - Informática Teórica

Ayudantía #5

NP-Complete Warriors

Semestre 2022-2



UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

DEPARTAMENTO
DE INFORMÁTICA

Definición: Derivación

Definición

Sea $G = (N, \Sigma, P, S)$ una gramática. La *relación de derivación* de G entre formas sentenciales $u, v \in V^*$ se define como:

$$u \Rightarrow_G v$$

donde

$$u \Rightarrow v$$

$$x\alpha y \Rightarrow x\beta y$$

cuando $\alpha \rightarrow \beta \in P$

En términos humanos, si $\alpha \rightarrow \beta$ es una producción de G , y aparece α en una forma sentencial, entonces podemos reemplazar α por β .

Definición: Lenguaje Generado

Definición

Sea $G = (N, \Sigma, P, S)$ una gramática. El *lenguaje generado* por G se define como:

$$\mathcal{L}(G) = \{\alpha \in \Sigma^* : S \Rightarrow_G^* \alpha\}$$

Clasificación de gramáticas - Jerarquía de Chomsky

Sea $G = (N, \Sigma, P, S)$ una gramática. Se dice que es:

- **Tipo 0 (Irrestricta):** Como definido.
- **Tipo 1 (Sensible al contexto):** Todas las producciones $\alpha \rightarrow \beta$ cumplen:

$$|\alpha| \leq |\beta|$$

Es decir, las formas sentenciales nunca se acortan. Una manera alternativa de describirlas es por producciones de la forma $\alpha A \beta \rightarrow \alpha \gamma \beta$, con $\alpha, \beta \in (N \cup \Sigma)^*$, $A \in N$, $\gamma \in (N \cup \Sigma)^+$; o sea, podemos reemplazar A por γ , pero solo en el contexto dado por α y β .

Jerarquía de Chomsky

- **Tipo 2 (Contexto libre):** Todas las producciones tienen la forma:

$$A \rightarrow \alpha$$

con $A \in N$, $\alpha \in (N \cup \Sigma)^+$.

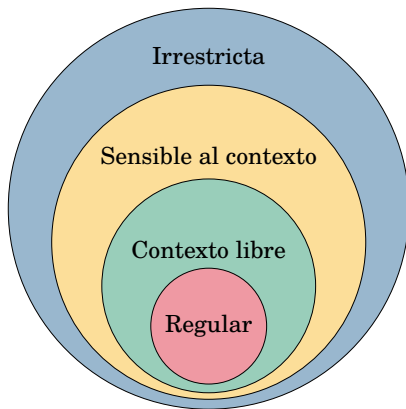
- **Tipo 3 (Regular):** Todas las producciones son de una de las formas:

$$A \rightarrow \alpha B, \quad \text{con } A, B \in N, \alpha \in \Sigma^*$$

$$A \rightarrow \alpha \quad \text{con } A \in N, \alpha \in \Sigma^+$$

Jerarquía de Chomsky

Observación importante: Las gramáticas de tipo 1 son un caso particular de tipo 0, las de tipo 2 particulares de tipo 1 y así sucesivamente.



Gramáticas de Contexto Libre (CFG)

Sea $G = (N, \Sigma, P, S)$ una gramática de contexto libre (CFG). Algunas convenciones que deben considerarse se presentan a continuación:

- Las mayúsculas son símbolos no-terminales (A, B, \dots, S, \dots)
- Otros símbolos (como minúsculas o números) son terminales.
- Se anotan sólo las producciones, y el símbolo no terminal de partida es S .
- Se agrupan a la derecha todas las producciones para un mismo no-terminal, separándolas con un *pipe* (barra vertical).

Árbol de derivación

Sea $G = (N, \Sigma, P, S)$ una CFG. El *árbol de derivación* de $\sigma \in \mathcal{L}(G)$ se construye de la derivación de σ poniendo el símbolo de partida como raíz, y como hijo de cada no-terminal los símbolos que deriva (en orden).

Árbol de derivación

Consideremos la gramática regalona:

$$E \rightarrow E + T \quad (1)$$

$$E \rightarrow T \quad (2)$$

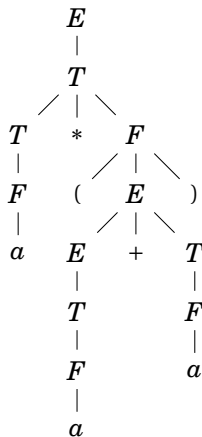
$$T \rightarrow T * F \quad (3)$$

$$T \rightarrow F \quad (4)$$

$$F \rightarrow (E) \quad (5)$$

$$F \rightarrow a \quad (6)$$

¿Cómo sería el árbol de derivación de la palabra $a * (a + a)$?



Derivaciones de extrema izquierda y derecha

- En una *derivación de extrema izquierda* (también llamada *canónica*) en cada paso aplicamos una producción al no-terminal más a la izquierda en la forma sentencial (postorden).
- En una *derivación de extrema derecha* en cada paso aplicamos una producción al no-terminal más a la derecha en la forma sentencial.

En las gramáticas de contexto libre siempre tenemos la libertad de aplicar las producciones en distinto orden, ya que las producciones de un no-terminal no dependen de su contexto.

Simplificación de Gramáticas

La idea es eliminar las producciones nulas. Para ello, debemos identificar todos los símbolos no-terminales A tales que $A \Rightarrow^* \varepsilon$. A estos símbolos se les llama *nullable symbols*.

Luego, creamos nuevas producciones cuyos lados derechos omitirán las apariciones de los no-terminales correspondientes.

Finalmente, se eliminan todas las producciones nulas. Si $S \Rightarrow^* \varepsilon$, solo se debe reponer la producción $S \rightarrow \varepsilon$ para generar el lenguaje original (ε incluido).

Simplificación de Gramáticas

Determinar el conjunto de los no-terminales que producen ε (llamémosle \mathcal{A}) se logra de la siguiente forma:

- Si $A \rightarrow \varepsilon \Rightarrow A \in \mathcal{A}$
- Si $A \rightarrow B_1 B_2 \dots B_n \wedge \forall B_i, B_i \in \mathcal{A} \Rightarrow A \in \mathcal{A}$

Simplificación de Gramáticas

Con el conjunto \mathcal{A} podemos eliminar las producciones nulas y generar los reemplazos que generan el mismo lenguaje.

- 1 $\forall A \in \mathcal{A}$ reemplazamos A por ε en las producciones P y escribimos esa nueva producción
- 2 Si en una producción hay 2 o más no-terminales que pertenecen a \mathcal{A} entonces reemplazamos 1 o más de esos símbolos no-terminales por ε en todas las combinaciones posibles y los resultados los escribimos como nuevas producciones
- 3 Eliminamos todas las producciones $A \rightarrow \varepsilon$

Simplificación de Gramáticas

Definición

Sea $G = (N, \Sigma, P, S)$ una gramática. Un símbolo $X \in V$ se llama *generante* si, para $\alpha \in \Sigma^*$

$$X \Rightarrow^* \alpha$$

Definición

Sea $G = (N, \Sigma, P, S)$ una gramática. Un símbolo $X \in V$ se dice *alcanzable* si existe una derivación tal que

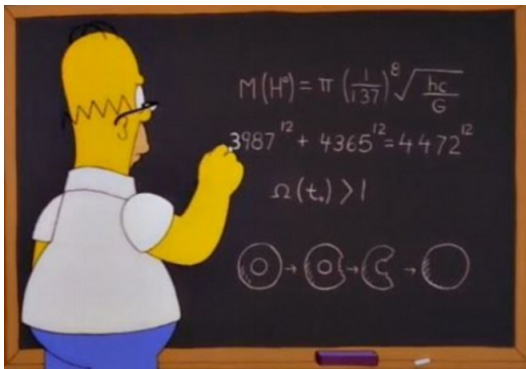
$$S \Rightarrow^* \alpha X \beta$$

Simplificación de Gramáticas

Para simplificar una gramática G debemos:

- 1 Eliminar todos los *nullable symbols*
- 2 Eliminar símbolos no generantes
- 3 Eliminar símbolos no alcanzables

Ejercicios



Ejercicio 1

Dada la siguiente gramática G :

$$S \rightarrow XT$$

$$X \rightarrow 1X2Z \mid 12L$$

$$Z2 \rightarrow 2Z$$

$$L2 \rightarrow 2L$$

$$ZT \rightarrow T3$$

$$LT \rightarrow 3$$

Describe el lenguaje que genera G

Solución Ejercicio 1

La idea es ir desarrollando y observar el patrón que se va generando:

XT
 $1X2ZT$
 $112L2ZT$
 $112L2T3$
 $1122LT3$
 112233

Luego, es claro que el lenguaje es $\mathcal{L}(G) = \{1^n 2^n 3^n : n \geq 1\}$

Buscar el resto de palabras queda para EQQAER

Ejercicio 2

Considere la gramática G definida sobre $\Sigma = \{0, 1\}$ y $N = \{S, A, B\}$, con las siguientes producciones:

$$S \rightarrow 0A|1B$$

$$A \rightarrow 0AA|1S|1$$

$$B \rightarrow 1BB|0S|0$$

Además, considere $\sigma \in \mathcal{L}(G)$, tal que $\sigma = 001101$. Demuestre que G es ambigua.

Solución Ejercicio 2

Basta encontrar dos generaciones distintas. Para σ , las sucesiones de producciones son las siguientes:

$$S \Rightarrow 0A \Rightarrow 00AA \Rightarrow 001S1 \Rightarrow 0011B1 \Rightarrow 001101$$

$$S \Rightarrow 0A \Rightarrow 00AA \Rightarrow 0011S \Rightarrow 00110A \Rightarrow 001101$$

¿Dudas?



INF155 - Informática Teórica

Ayudantía #5

NP-Complete Warriors

Semestre 2022-2



UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

DEPARTAMENTO
DE INFORMÁTICA