# Algoritmo CKY

Clase 19

IIC 2223

Prof. Cristian Riveros

# ¿cómo verificar si $w \in \mathcal{L}(\mathcal{G})$?

Dado un lenguaje libre de contexto $L$ y una palabra $w$:

¿cómo verificamos si $w \in L$?

¿cómo determinar si la palabra $bbaba \in \mathcal{L}(\mathcal{G})$?

$$
\begin{array}{lcl}
\mathrm{S} & \rightarrow & \mathrm{XY} \mid \mathrm{YZ} \\
\mathrm{X} & \rightarrow & \mathrm{YY} \mid \mathrm{a} \\
\mathrm{Y} & \rightarrow & \mathrm{YX} \mid \mathrm{b} \\
\mathrm{Z} & \rightarrow & \mathrm{XZ} \mid \mathrm{XX} \mid \mathrm{a}
\end{array}
$$

# ¿cómo verificar si $w \in \mathcal{L}(\mathcal{G})$?

Dado un lenguaje libre de contexto $L$ y una palabra $w$:

**¿cómo verificamos si $w \in L$?**

- Convertimos $\mathcal{G}$ en formal normal de Chomsky.
- Probamos todas las derivaciones de altura a lo mas $|w| + 1$.
- Si encontramos una derivación retornamos `TRUE`.

**¿cuántas derivaciones debemos probar?**

# Algoritmo CKY

- Inventado por:

| | |
|---|---|
| John | **C**ocke |
| Tadao | **K**asami |
| Daniel | **Y**ounger |

- Algoritmo **cúbico** en $|w|$ y **lineal** en $|\mathcal{G}|$:

Tiempo: $\mathcal{O}(|w|^3 \cdot |\mathcal{G}|)$

- Un ejemplo del uso de **programación dinámica**.

Por simplicidad asumiremos que
las gramáticas esta en **Forma Normal de Chomsky** (CNF)

... CKY se puede adaptar para producciones mayores que 2

# Tabla del algoritmo CKY

Para una palabra $w = a_1 a_2 \dots a_n$ y una gramática $\mathcal{G} = (V, \Sigma, P, S)$ construimos la **tabla CKY**:

$$
\begin{array}{|lllll}
X_{15} & & & & \\
X_{14} & X_{25} & & & \\
X_{13} & X_{24} & X_{35} & & \\
X_{12} & X_{23} & X_{34} & X_{45} & \\
X_{11} & X_{22} & X_{33} & X_{44} & X_{55} \\
\hline
a_1 & a_2 & a_3 & a_4 & a_5
\end{array}
$$

- Cada $X_{ij}$ es un subconjunto de variables en $V$ con $1 \le i \le j \le n$.
- Se cumple que $A \in X_{ij}$ si, y solo si, $A \underset{\mathcal{G}}{\overset{\star}{\Rightarrow}} a_i \dots a_j$.

# Tabla del algoritmo CKY

## Ejemplo

Considere la palabra *bbaba* y la gramática:

$$
\begin{array}{lcl}
S & \rightarrow & XY \mid YZ \\
X & \rightarrow & YY \mid a \\
Y & \rightarrow & YX \mid b \\
Z & \rightarrow & XZ \mid XX \mid a
\end{array}
$$

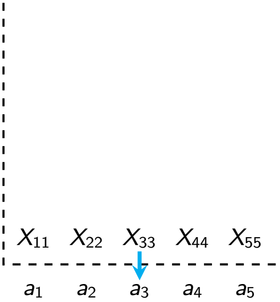
| | | | | |
|---|---|---|---|---|
| $\{S, Y\}$ | | | | |
| $\{S, Y\}$ | $\{X, Z\}$ | | | |
| $\{X, Z\}$ | $\{X\}$ | $\{S\}$ | | |
| $\{X\}$ | $\{S, Y\}$ | $\{S\}$ | $\{S, Y\}$ | |
| $\{Y\}$ | $\{Y\}$ | $\{X, Z\}$ | $\{Y\}$ | $\{X, Z\}$ |
| $b$ | $b$ | $a$ | $b$ | $a$ |

¿cómo **verificamos** si $w \in \mathcal{L}(\mathcal{G})$ usando la tabla CKY?

¿cómo **construimos** la tabla CKY?

# Algoritmo CKY: construcción de la tabla CKY

## Paso 0 (inicial)




Para cada $i$, construimos el conjunto $X_{ii} \subseteq V$ tal que:

$$X_{ii} \quad = \quad \left\{ X \in V \mid X \to a_i \in P \right\}$$

# Algoritmo CKY: construcción de la tabla CKY

## Ejemplo (Paso 0)

Considere la palabra *bbaba* y la gramática:

$$
\begin{array}{lcl}
S & \rightarrow & XY \mid YZ \\
X & \rightarrow & YY \mid a \\
Y & \rightarrow & YX \mid b \\
Z & \rightarrow & XZ \mid XX \mid a
\end{array}
$$

| | | | | |
|---|---|---|---|---|
| $\{Y\}$ | $\{Y\}$ | $\{X, Z\}$ | $\{Y\}$ | $\{X, Z\}$ |
| $b$ | $b$ | $a$ | $b$ | $a$ |

# Algoritmo CKY: construcción de la tabla CKY

## Paso 1

$$
\begin{array}{|ccccc}
X_{12} & X_{23} & X_{34} & X_{45} & \\
X_{11} & X_{22} & X_{33} & X_{44} & X_{55} \\
\hline
a_1 & a_2 & a_3 & a_4 & a_5
\end{array}
$$

Para cada $i$, construimos el conjunto $X_{i\,i+1} \subseteq V$ tal que:

$$X_{i\,i+1} \quad = \quad \left\{ X \in V \mid X \to YZ \in P \text{ para algún } Y \in X_{ii} \wedge Z \in X_{i+1\,i+1} \right\}$$

# Algoritmo CKY: construcción de la tabla CKY

## Ejemplo (Paso 1)

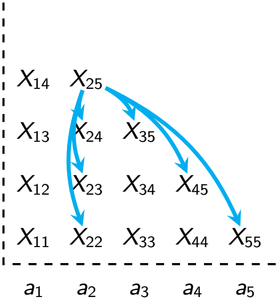
Considere la palabra *bbaba* y la gramática:

$$
\begin{array}{lcl}
S & \rightarrow & XY \mid YZ \\
X & \rightarrow & YY \mid a \\
Y & \rightarrow & YX \mid b \\
Z & \rightarrow & XZ \mid XX \mid a
\end{array}
$$

| | | | | |
|---|---|---|---|---|
| $\{X\}$ | $\{S, Y\}$ | $\{S\}$ | $\{S, Y\}$ | |
| $\{Y\}$ | $\{Y\}$ | $\{X, Z\}$ | $\{Y\}$ | $\{X, Z\}$ |
| $b$ | $b$ | $a$ | $b$ | $a$ |

# Algoritmo CKY: construcción de la tabla CKY

## Paso $k$ ($k > 0$))

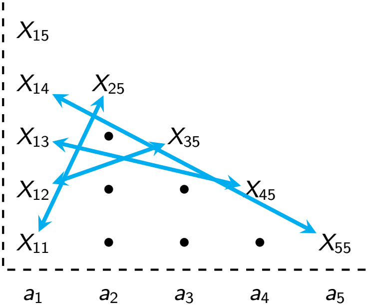



Para cada $i$, construimos el conjunto $X_{i\,i+k} \subseteq V$ tal que:

$$X_{i\,i+k} \quad = \quad \left\{ \; X \in V \;\middle|\; \begin{array}{ll} \exists j \in [i, i+k). & X \to YZ \in P \\ & \text{para algún } Y \in X_{ij} \wedge Z \in X_{j+1\,i+k} \end{array} \right\}$$

# Algoritmo CKY: construcción de la tabla CKY

Paso $k$ ($k > 0$))




Para cada $i$, construimos el conjunto $X_{i\,i+k} \subseteq V$ tal que:

$$X_{i\,i+k} = \left\{ X \in V \;\middle|\; \begin{array}{ll} \exists j \in [i, i+k). & X \to YZ \in P \\ & \text{para algún } Y \in X_{ij} \wedge Z \in X_{j+1\,i+k} \end{array} \right\}$$

# Algoritmo CKY: construcción de la tabla CKY

## Ejemplo (Paso k)

Considere la palabra *bbaba* y la gramática:

$$
\begin{array}{lcl}
S & \rightarrow & XY \mid YZ \\
X & \rightarrow & YY \mid a \\
Y & \rightarrow & YX \mid b \\
Z & \rightarrow & XZ \mid XX \mid a
\end{array}
$$

| | | | | |
|---|---|---|---|---|
| $\{S, Y\}$ | | | | |
| $\{S, Y\}$ | $\{X, Z\}$ | | | |
| $\{X, Z\}$ | $\{X\}$ | $\{S\}$ | | |
| $\{X\}$ | $\{S, Y\}$ | $\{S\}$ | $\{S, Y\}$ | |
| $\{Y\}$ | $\{Y\}$ | $\{X, Z\}$ | $\{Y\}$ | $\{X, Z\}$ |
| $b$ | $b$ | $a$ | $b$ | $a$ |

# Algoritmo CKY: construcción de la tabla CKY

## Otro ejemplo

Considere la palabra *abba* y la gramática:

$$
\begin{array}{lcl}
S & \rightarrow & XY \mid YX \mid SS \mid \\
 & & XZ \mid YW \\
X & \rightarrow & a \\
Y & \rightarrow & b \\
Z & \rightarrow & SY \\
W & \rightarrow & SX
\end{array}
$$

| | | | |
|---|---|---|---|
| $\{S\}$ | | | |
| $\{Z\}$ | $\varnothing$ | | |
| $\{S\}$ | $\varnothing$ | $\{S\}$ | |
| $\{X\}$ | $\{Y\}$ | $\{Y\}$ | $\{X\}$ |
| $a$ | $b$ | $b$ | $a$ |

# Algoritmo CKY

**input** : Una gramática $\mathcal{G} = (V, \Sigma, P, S)$ y una palabra $w = a_1 a_2 \dots a_n$
**output:** `TRUE` ssi $w = a_1 a_2 \dots a_n \in \mathcal{L}(\mathcal{G})$

**Function** `AlgoritmoCKY` *($\mathcal{G}$, $w$)*

- **for** $i \leftarrow 1$ **to** $n$ **do**
  - **let** $X_{ii} = \varnothing$
  - **for** $X \rightarrow c \in P$ **do**
    - **if** $c = a_i$ **then let** $X_{ii} = X_{ii} \cup \{X\}$
- **for** $k \leftarrow 1$ **to** $n - 1$ **do**
  - **for** $i \leftarrow 1$ **to** $n - k$ **do**
    - **let** $X_{i\,i+k} = \varnothing$
    - **for** $j \leftarrow i$ **to** $i + k - 1$ **do**
      - **for** $X \rightarrow YZ \in P$ **do**
        - **if** $Y \in X_{i\,j} \wedge Z \in X_{j+1\,i+k}$ **then let** $X_{i\,i+k} = X_{i\,i+k} \cup \{X\}$
- **return check** $S \in X_{1n}$

# Análisis algoritmo CKY

## Correctitud algoritmo CKY

Para toda gramática $\mathcal{G}$ y para toda palabra $w \in \Sigma^*$ se tiene que:

$$\texttt{AlgoritmoCKY}(\mathcal{G}, w) = \texttt{TRUE} \quad \text{ssi} \quad w \in \mathcal{L}(\mathcal{G})$$

Demostración (ejercicio)

Si el input es de tamaño $|w|$ y la gramática es de tamaño $|\mathcal{G}|$, entonces:

$$\text{Tiempo Algoritmo CKY:} \;\; \mathcal{O}(|w|^3 \cdot |\mathcal{G}|)$$

¿es posible verificar si $w \in \mathcal{L}(\mathcal{G})$ en **tiempo lineal** en $|w|$?