

Ambigüedad y forma normal de Chomsky



UNIVERSIDAD
DE CHILE

Arturo Merino



Recuerdo / Plan

Recuerdo/Plan

- **Recuerdo:**

- Equivalencia AP-GLC
- $GLC \rightarrow AP$: simular las producciones por un lado.
- $AP \rightarrow GLC$: programación dinámica.

- **Plan:**

- Ambigüedad
- Forma normal de Chomsky
- Wrap-up lenguajes libres de contexto

Ambigüedad

Ambigüedad

- Strings con derivación “única”
 - Podemos definir funciones/propiedades sobre la derivación
 - Interpretabilidad

Def.

Un string w será *derivado ambiguamente* por G si tiene dos o más derivaciones por la izquierda (o dos o más árboles de derivación).

- Ej:
 - aaa por $S \rightarrow SS \mid a$.
 - $a + a \times a$ por $E \rightarrow E + E \mid E \times E \mid a$.
 - $()()()$ por $S \rightarrow SS \mid (S) \mid \varepsilon$.

Def.

Una gramática será *ambigua* si deriva algún string ambiguamente.

Desambiguando gramáticas

- Nos gustaría diseñar gramáticas no-ambiguas
- **Construcción usual:** dar roles distintos a la fuente de ambigüedad.
- **Ej:**
 - $S \rightarrow SS \mid a.$
 - $E \rightarrow E + E \mid E \times E \mid a.$
 - $S \rightarrow SS \mid (S) \mid \varepsilon.$
- ¿Algoritmo?
- No siempre se puede: hay lenguajes inherentemente ambiguos.
- **Ej:**
 - $\{a^i b^j c^k : i = j \text{ o } j = k\}.$

Forma normal de Chomsky

Forma normal de Chomsky

- Para AP es útil asumir formas específicas
- ¿Algo similar en GLC?
 - Útil al diseñar algoritmos sobre GLC
 - Diseño de parsers para GLC
- Forma simplificada: Forma normal de Chomsky.

Def.

Una GLC G está en *forma normal de Chomsky* si

- La variable inicial no aparece al lado derecho de las reglas,
- Las reglas son de la forma $A \rightarrow BC$ para variables A, B, C , o
- De la forma $A \rightarrow a$ para variables A y terminales a , o
- De la forma $S \rightarrow \varepsilon$ (sólo si $\varepsilon \in \mathcal{L}(G)$)

Forma normal de Chomsky

Teo.

Toda GLC tiene una GLC equivalente en forma normal de Chomsky.

- **Idea:** corregir S al LD, $A \rightarrow \varepsilon$, $A \rightarrow B$, $A \rightarrow aBC \dots$, $A \rightarrow ABC \dots$
 - S al LD: agregar nueva variable inicial.
 - $A \rightarrow \varepsilon$: reemplazar dónde puede aparecer A por ε y eliminar.
 - $A \rightarrow B$: si $A \xRightarrow{*} B$, podemos agregar las reglas de B a A .
 - $A \rightarrow aBC \dots$: podemos agregar una nueva variable T_a , reemplazar la regla por $A \rightarrow T_a BC \dots$ y agregar la regla $T_a \rightarrow a$.
 - $A \rightarrow ABC \dots$: podemos separar en varias subreglas.
- **Dem:** Inducción estructural.
- **Proc:** Agregar nueva variable inicial, Eliminar ε , eliminar reglas unitarias, aislar terminales, binarizar reglas.
- **Ej:**
 - $S \rightarrow aSb \mid ab$
 - $S \rightarrow aAB \mid B, A \rightarrow a \mid \varepsilon, B \rightarrow b$.

Dudas/Preguntas

Recapitulación sobre lenguajes libres de contexto

Lenguajes libres de contexto

- **Lenguaje libre de contexto:** aceptado por AP o generado por GLC.
- Diseñar alguno para probar que un lenguaje es libre de contexto.
- Cerrados bajo operaciones regulares: $\cup$, concatenación y $*$.
- $\text{GLC} \rightarrow \text{AP}$.
 - Adivinar reglas de producción
 - Usar el stack para mantener la derivación por la izquierda
- $\text{AP} \rightarrow \text{GLC}$.
 - Simplificación: AP con único estado final, sin replace y acepta con pila vacía.
 - **PD:** sobre A_{pq} = "ir de p a q con pila vacía"
 - Para todo r intermedio, agregar $A_{pq} \rightarrow A_{pr}A_{rq}$
 - Apilamos y desapilamos el mismo símbolo: $A_{pq} \rightarrow aA_{rs}b$
 - $A_{pp} \rightarrow \varepsilon$
- **Tópicos en GLC:** Ambigüedad y forma normal de Chomsky.

Wrap-Up

- **Próx. clase:**
 - Lenguajes no libres de contexto