La Forma Normal de Chomsky

Algoritmos Polinomiales para el Problema de la Palabra en CFL

Universidad de Cantabria

Esquema

Introducción

Hemos visto hasta aquí como demostrar si una palabra esta dentro de un lenguaje libre de contexto (CFL). El problema es que el algoritmo es de naturaleza exponencial. ¿Como reducir la complejidad de ese algoritmo?

Introducción

El problema básico es que, si tenemos una gramática muy general, no parece viable otra posibilidad que probar con todas las combinaciones de reglas gramaticales.

Por lo tanto, hay que utilizar otro tipo de gramáticas, que sean menos generales que las propias.

Introducción

El problema básico es que, si tenemos una gramática muy general, no parece viable otra posibilidad que probar con todas las combinaciones de reglas gramaticales.
Por lo tanto, hay que utilizar otro tipo de gramáticas, que sean menos generales que las propias.

Forma Normal de Chomsky

Definición (Forma Normal de Chomsky)

Una gramática libre de contexto $G := (V, \Sigma, Q_0, P)$ se dice que está en forma normal de Chomsky si es λ-libre y las únicas producciones (exceptuando, eventualmente, la única $\lambda-$producción $Q_0 \mapsto \lambda$), son exclusivamente de uno de los dos tipos siguientes.

- $A \mapsto a$, *con* $A \in V$ *y* $a \in \Sigma$,
- $A \mapsto CD$, *con* $A, C, D \in V$.

Preguntas

- ¿Son estas gramáticas $\lambda-$libres?
- ¿Tienen producciones simples?
- ¿Tienen símbolos inútiles?
- ¿Se pueden detectar utilizando un autómata finito?

Algoritmo

Teorema (Transformación a Forma Normal de Chomsky)

Toda gramática libre de contexto es equivalente a una gramática libre de contexto en Forma Normal de Chomsky. Además, esta equivalencia es algorítmicamente computable.

Algoritmo

Supongamos que tenemos una gramática $G = (V, \Sigma, Q_0, P)$ propia. Procederemos del modo siguiente:
Definamos un par de clases $\bar{V}$ y $\bar{P}$ de símbolos no terminales y producciones.

Algoritmo

Inicializar con $\bar{V} := V$, $\bar{P} = \emptyset$ y añadimos estas producciones por defecto:

- Si $Q_0 \mapsto \lambda$ está en P, añadir $Q_0 \mapsto \lambda$ a $\bar{P}$ sin modificar $\bar{V}$.
- Si en P hay una producción del tipo $A \mapsto a \in \Sigma$ entonces, añadir $A \mapsto a$ a $\bar{P}$ sin modificar $\bar{V}$.
- Si en P hay una producción del tipo $A \mapsto CD$, con $C, D \in V$, entonces, añadir $A \mapsto CD$ a $\bar{P}$ sin modificar $\bar{V}$.

Simplemente, añadimos las producciones que cumplan la definición.

Algoritmo

Finalmente, para cada producción en P del tipo

$$A \mapsto X_1 \cdots X_k,$$

con $X_i \in V \cup \Sigma$ que no sea de ninguno de los tres tipos anteriores realizar las tareas siguientes:

- Para cada i tal que $X_i \in V$, no modificar $\bar{V}$
- Para cada i tal que $X_i \in \Sigma$, añadir a $\bar{V}$ una nueva variable $\bar{X}_i$, distinta a todas las que ya estuvieran en $\bar{V}$. Añadir a $\bar{P}$ la producción $\bar{X}_i \mapsto X_i$ en este caso.
- Añadir a $\bar{P}$ la producción $A \mapsto X_1' \cdots X_k'$, donde X_i' viene dada por:

$$X_i' := \begin{cases} X_i, & \text{si } X_i \in V \\ \bar{X}_i, & \text{en otro caso} \end{cases}$$

- Si $k = 2$, no modificar.
- Si $k > 2$, reemplazar en $\bar{P}$, la producción $A \mapsto X'_1 \cdots X'_k$ por una cadena de producciones:

$$A \mapsto X'_1 Y_2,\ Y_2 \mapsto X'_2 Y_3, \cdots, Y_{k-1} \mapsto X'_{k-1} X'_k,$$

añadiendo a $\bar{V}$ las variables $\{Y_2, \ldots, Y_{k-1}\}$.

Esto termina con el algoritmo.

Nota

Obsérvese que los árboles de derivación asociados a gramáticas en forma normal de Chomsky son árboles binarios.

Nota

Este proceso algorítmico es polinomial, otra vez, pero:

- ¿Cuantas producciones nuevas se añaden en el peor de los casos?
- ¿Se añaden variables inútiles durante el proceso?
- ¿Por qué pedimos que la gramática sea propia como entrada del algoritmo?

Conclusiones

Hemos estudiado la forma normal de Chomsky:

- La gramática transformada sigue siendo propia.
- Las gramáticas pueden añadir nuevas derivaciones a una misma palabra.
- El árbol de derivación siempre será binario, esto es muy importante.