La Representación de las Gramáticas

El Generador del Lenguaje

Universidad de Cantabria

Esquema

Introducción

Las gramáticas libres de contexto son más generales que las gramáticas regulares. Empecemos poniendo un ejemplo de los siguientes problemas que pueden surgir:

- Puede estar “sobre-definida”, en el sentido de que haya información inútil.
- Dada una palabra, es difícil decidir como ha sido generada por la gramática.

Introducción

Consideremos la gramática $G := (\{Q_0, A, B\}, \{a, b\}, Q_0, P)$, donde las producciones de P son dadas por:

$$P := \{Q_0 \mapsto a \mid A, A \mapsto AB, B \mapsto b\}.$$

¿Qué utilidad tiene el símbolo A?

Introducción

Consideremos la gramática $G := (\{Q_0, A, B\}, \{a, b\}, Q_0, P)$, donde las producciones de P son dadas por:

$$P := \{Q_0 \mapsto a|aQ_0, A \mapsto BB \mid Q_0, B \mapsto b\}.$$

¿Qué utilidad tiene el símbolo A?

Definición de Símbolo Útil

Definición (Símbolos Útiles)

Sea $G := (V, \Sigma, Q_0, P)$ *una gramática libre de contexto. Llamamos símbolos útiles dec* G *a todos los símbolos (terminales o no)* $X \in V \cup \Sigma$ *tales que existen* $\alpha, \gamma \in (V \cup \Sigma)^*$ *y* $\omega \in \Sigma^*$ *de tal modo que:*

$$Q_0 \vdash_G \alpha X \gamma, \quad \text{y} \quad \alpha X \gamma \vdash_G \omega.$$

Los símbolos inútiles son los que no son útiles.

Una Pequeña Nota

Encontrar los símbolos inútiles es difícil a primera vista. Dividamos nuestro problema definiendo dos subclases de símbolo inútil.

Definición

Definición

Sea $G := (V, \Sigma, Q_0, P)$ una gramática libre de contexto.

- *Llamamos símbolos productivos (o fecundos) de G a todos los símbolos no terminales $X \in V$ tales que existe $\omega \in \Sigma^*$ tal que $X \vdash_G \omega$. Son improductivos (o infecundos) a los que no satisfacen esta propiedad.*
- *Llamamos símbolos accesibles de G a todos los símbolos (terminales o no) $X \in V \cup \Sigma$ tales que existen $\alpha, \gamma \in (V \cup \Sigma)^*$ de tal modo que:*

$$Q_0 \vdash_G \alpha X \gamma.$$

Se llaman inaccesibles a los que no son accesibles.

Unas Aclaraciones

Nótese que si X es un símbolo útil, se han de producir dos propiedades. De una parte, la propiedad $Q_0 \vdash_G \alpha X \gamma$ que nos dice que X es accesible. De otra parte, sabemos que $\alpha X \gamma \vdash_G \omega$. Y por tanto, ha de existir $\beta \in \Sigma^*$ tal que $X \vdash_G \beta$.

Proposición

Teorema

Si $G := (V, \Sigma, Q_0, P)$ es una gramática libre de contexto, entonces los símbolos útiles son productivos y accesibles. El recíproco no es cierto, en general.

Unas Aclaraciones

Lo que nos dice el teorema es que no podemos simplemente eliminar los símbolos que no sean productivos y los que no sean accesibles. Pero no mucho más.

Otro Resultado

Teorema

Si $G := (V, \Sigma, Q_0, P)$ es una gramática libre de contexto, y libre de símbolos infecundos, entonces todo símbolo es útil si y solamente si es accesible.

Algoritmo de Símbolos Infecundos

Toda gramática libre de contexto es equivalente a una gramática libre de contexto sin símbolos infecundos. Además, dicha equivalencia puede hacerse de manera algorítmica.

Algoritmo de Símbolos Infecundos

La prueba del resultado es consecuencia directa del algoritmo. No responderemos a preguntas tan esenciales como la memoria necesaria o el número de operaciones, aunque son algoritmos polinomiales.

Algoritmo de Símbolos Infecundos

Entrada: Una gramática libre de contexto $G = (V, \Sigma, Q_0, P)$.
$M := \emptyset$
$N := \{A \in V \ : \ (A \mapsto \omega) \in P, \ \omega \in \Sigma^*\}$
mientras $N \neq M$ **hacer**
$M := N$
$N := \{A \in V \ : \ (A \mapsto \omega) \in P, \ \omega \in (N \cup \Sigma)^*\} \cup N.$
mientras fin
Salida $\bar{G} := (V \cap N, \Sigma, Q_0, \bar{P})$, donde $\bar{P}$ son las producciones de P que involucran solamente símbolos en $(V \cap N) \cup \Sigma \cup \{\lambda\}$

Notas

Esta claro que este algoritmo funciona, y que acaba después de un número de pasos finito, pero es fundamental notar la idea.

Podríamos habernos preguntado también lo siguiente: ¿Se podría haber resuelto este problema con un autómata finito?

Notas

Esta claro que este algoritmo funciona, y que acaba después de un número de pasos finito, pero es fundamental notar la idea.
Podríamos habernos preguntado también lo siguiente:
¿Se podría haber resuelto este problema con un autómata finito?

Eliminación de Símbolos Inaccesibles

Como ocurre con los símbolos no productivos, toda gramática libre de contexto es equivalente a una gramática libre de contexto sin símbolos inaccesibles. Además, dicha equivalencia puede hacerse de manera algorítmica.

Eliminación de Símbolos Inaccesibles

Entrada: Una gramática libre de contexto $G = (V, \Sigma, Q_0, P)$.
$M := \{Q_0\}$
$N := \{X \in V \cup \Sigma \ : \ \exists A \in M, \ \exists \alpha, \beta \in (V \cup \Sigma)^* , \text{ con } A \mapsto \alpha X \beta \text{ en } P\}$.

mientras $N \neq M$ **hacer**

$M := N$

$N := \{X \in V \cup \Sigma \ : \ \exists A \in M, \ \exists \alpha, \beta \in (V \cup \Sigma)^* , \text{ con } A \mapsto \alpha X \beta \text{ en } P\}$.

mientras fin

Salida: La gramática $\bar{G} = (\bar{V}, \bar{\Sigma}, Q_0, \bar{P})$, con
$\bar{V} := N \cap V, \ \ \bar{\Sigma} := N \cap \Sigma$,
$\bar{P} :=$
$\{\text{Las producciones de } P \text{ que sólo contienen los elementos de } \bar{V} \cup \bar{\Sigma}\}$.

Conclusiones

Sabemos que hay símbolos que no son necesarios:

- Los símbolos inútiles, que no generan palabras. Estos son difíciles de detectar en cambio, dos subclases son fáciles de detectar:
  - Los símbolos inaccesibles.
  - Los símbolos improductivos.
- Además, un resultado nos dice como eliminar los símbolos que no son útiles por medio de algoritmos.