Lenguajes Libres de Contexto

Más allá de los Autómatas Finitos.

Universidad de Cantabria

Esquema

Introducción

Hemos visto que los lenguajes regulares son demasiado simples. Ahora tenemos que dar un paso más allá y estudiar los lenguajes libres de contexto.

Introducción

Los lenguajes libres de contexto tienen una aplicación a los compiladores, aunque existen otras aplicaciones como la compartición de información.

Introducción

¿Por qué no ir un paso más allá? ¿Por qué no se estudian las gramáticas sensibles al contexto?

- Para gramáticas sensibles al contexto, el problema de decidir si el lenguaje que genera es vacío o no también es un problema indecidible.
- En cuanto al Problema de Palabra para gramáticas sensibles al contexto, el problema es **PSPACE**–completo para ciertas subclases, con lo que se hace impracticable para su uso en la vida real.

Introducción

¿Por qué no ir un paso más allá? ¿Por qué no se estudian las gramáticas sensibles al contexto?

- Para gramáticas sensibles al contexto, el problema de decidir si el lenguaje que genera es vacío o no también es un problema indecidible.
- En cuanto al Problema de Palabra para gramáticas sensibles al contexto, el problema es **PSPACE**–completo para ciertas subclases, con lo que se hace impracticable para su uso en la vida real.

Nota

- PSPACE se refiere a la clase de problemas que pueden ser resueltos usando un número "eficiente" **de espacio**.
- PSPACE-Completo es la clase de problemas más difíciles de PSPACE.

Definición

Definición (Gramáticas libres de contexto o de Tipo 2)

Llamaremos gramática libre de contexto a toda gramática $G = (V, \Sigma, Q_0, P)$ tal que todas las producciones de P son del tipo siguiente:
$A \mapsto \omega$, donde $A \in V$ y $\omega \in (\Sigma \cup V)^*$.
Un lenguaje **libre de contexto** *es un lenguaje generado por una gramática libre de contexto.*

Definición

El sistema de transición asociado a una gramática libre de contexto es el mismo que asociamos a una gramática cualquiera. Usaremos el símbolo $C \vdash_G C'$ para indicar que la configuración C' es deducible de la configuración C mediante computaciones de G.

Árboles de Derivación

Definición

Llamamos formas sentenciales a todos los elementos ω *de* $(V \cup \Sigma)^*$*. Llamaremos formas terminales a las formas sentenciales que sólo tienen símbolos en el alfabeto de símbolos terminales, es decir, a los elementos de* Σ^**.*

Árboles de Derivación

Definición (Árbol de Derivación)

Sea $G := (V, \Sigma, Q_0, P)$ una gramática libre de contexto, sea $A \in V$ una variable. Diremos que un árbol $\mathcal{T}_A := (V, E)$ etiquetado es un árbol de derivación asociado a G si verifica las propiedades siguientes:

- *La raíz del árbol es un símbolo no terminal (i.e., una variable) y los otros nodos interiores están etiquetados con una variable y las hojas con un símbolo terminal o λ.*
- *Si un nodo está etiquetado con una variable X y sus descendientes (leídos de izquierda a derecha) en el árbol son $X_1, \ldots, X_k$, entonces hay una producción $X \mapsto X_1 \cdots X_k$ en G.*

Árboles de Derivación

Ejemplo

Hallar árboles de derivación para las gramáticas siguientes:

- $A \mapsto BF, B \mapsto EC, E \mapsto a, C \mapsto b, F \mapsto c$, *de tal manera que la raíz sea* A *y las hojas estén etiquetadas con* a, b, c *en este orden.*
- $Q_0 \mapsto zABz, B \mapsto CD, C \mapsto c, D \mapsto d, A \mapsto a$, *de tal manera que la raíz sea* Q_0 *y las hojas estén etiquetadas (de izquierda a derecha) mediante* z, a, c, d, z.

Árboles de Derivación

Sea $G := (V, \Sigma, Q_0, P)$ una gramática libre de contexto, sea $A \in V$ una variable. Sea $\mathcal{T}_A$ un árbol asociado a la gramática con raíz A. Sea $\omega \in (V \cup \Sigma)^*$ la forma sentencial obtenida leyendo de izquierda a derecha los símbolos de las hojas de $\mathcal{T}_A$. Entonces, $A \vdash_G \omega$. En particular, las formas sentenciales alcanzables desde el símbolo inicial Q_0 están representadas por los árboles de derivación de raíz Q_0.

Árboles de Derivación

Las palabras sobre el alfabeto Σ están en el lenguaje $L(G)$ generado por una gramática G si y solamente si existe un árbol de derivación cuyas hojas (leídas de izquierda a derecha) producen la palabra ω.

El Problema de la Vacuidad

El problema de decidir si es o no vacío el lenguaje generado por una gramática libre de contexto se puede resolver algorítmicamente.

Resultado

Teorema (Vacuidad de un lenguaje libre de contexto)

El problema de la vacuidad de los lenguajes generados por gramáticas libres de contexto es decidible. Es decir, existe un algoritmo que toma como entrada una gramática libre de contexto G y devuelve una respuesta afirmativa si $L(G) \neq \emptyset$ y negativa en caso contrario.

Idea del Algoritmo

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**

        $M := N$

        $N := \{A \in V \; : \; (A \mapsto \omega) \in P, \; \omega \in (N \cup \Sigma)^*\} \cup N.$

    finaliza mientras

si $Q_0 \in N$, **entonces devuelve** SI

en otro caso devuelve NO

finaliza si

Idea de la Demostración

Consideremos una cadena de subconjuntos N_i de V que reflejan los sucesivos pasos por el ciclo mientras.

Idea de la Demostración

Escribamos $N_0 = \emptyset$ y denotemos por N_i al conjunto obtenido en el $i-$ésimo paso por el ciclo mientras, sin considerar la condición de parada. Esto es,

$$N_i := \{A \in V \ : \ (A \mapsto \omega) \in P, \ \omega \in (N_{i-1} \cup \Sigma)^*\} \cup N_{i-1}.$$

Idea de la Demostración

Por construcción observamos, además, que si existe un paso i tal que $N_i = N_{i+1}$, entonces, $N_i = N_m$ para todo $m \geq i + 1$.

Idea de la Demostración

Si definimos la altura por el número de nodos atravesados en el camino más largo, se tiene que:

> Una variable $X \in V$ verifica que $X \in N_i$ si y solamente si existe un árbol de derivación de G de altura $i+1$ que tiene X como raíz y cuyas hojas están etiquetadas con símbolos en $\Sigma \cup \{\lambda\}$.