Ciencias de la Computación II

Equivalencias y Árboles de Derivación

Eduardo Contrera Schneider

Universidad de la Frontera

13 de octubre de 2016

Equivalencias

Volvamos un poco a las gramáticas regulares y estudiemos métodos constructivos entre autómatas finitos y éstas. Supongamos que L es un lenguaje regular. Se puede obtener una gramática regular que genere L por medio de un AFD $M = (Q, \Sigma, s, F, \delta)$ para el cual $L = L(M)$. Definimos $G = (N, \Sigma, S, P)$ como

- $N = Q$
- $\Sigma = \Sigma$
- $S = s$
- $P = \{(q, ap) | \delta(q, a) = p\} \cup \{(q, \epsilon) | q \in F\}$

En esta gramática los q_i son no terminales y no es difícil demostrar que genera el lenguaje aceptado por M.

También es posible construir un AFN M a partir de una gramática regular de forma que $L(G) = L(M)$. Sea $G = (N, \Sigma, S, P)$ una gramática regular; se define $M = (Q, \Sigma, s, F, \Delta)$ mediante

- $Q = N \cup \{f\}$, donde f es un símbolo nuevo
- $s = S$
- $F = \{f\}$

y Δ se construye como se indica a continuación, a partir de las producciones de P:

- Si $A \to \sigma_1, \sigma_2, ..., \sigma_n B$ es una producción de P con A y B como no terminales, entonces se añadirán a Q los nuevos estados $q_1, q_2, ..., q_{n-1}$ y las transformaciones siguientes

$$\Delta(A, \sigma_1, \sigma_2, ..., \sigma_n) = \Delta(q_1, \sigma_2, ..., \sigma_n) = ... = \Delta(q_{n-1}, \sigma_n) = B$$

- Si $A \to \sigma_1, \sigma_2, ..., \sigma_n$ es una producción de P, entonces se añadirán a Q los nuevos estados $q_1, q_2, ..., q_{n-1}$ y a Δ, las transciones siguientes

$$\Delta(A, \sigma_1, ..., \sigma_n) = \Delta(q_1, \sigma_2, ..., \sigma_n) = ... = \Delta(q_{n-1}, \sigma_n) = f$$

Árboles de Derivación

Cuando se realiza el proceso de sustitución al crear una cadena mediante las producciones especificadas por la gramática, éste puede ser descrito mediante un árbol que muestra la manera en que cada no terminal contribuye a formar la cadena final de símbolos terminales. Éste se construye creando un nodo raíz que se etiqueta con el símbolo inicial. Todo nodo etiquetado con un no terminal también tiene unos nodos hijos etiquetados con los símbolos del lado derecho de la producción usada para sustituir ese no terminal. Los nodos que no tienen hijos deben ser etiquetados con símbolos terminales.

De los ejemplos precedentes podemos observar que para una misma cadena pueden haber árboles distintos. Al respecto podemos decir lo siguiente:

Ambigüedad

Una gramática se dice que es **ambigua** si hay dos o más árboles de derivación distintos para la misma cadena. Una gramática en la cual, para toda cadena w, todas las derivaciones de w tienen el mismo árbol de derivación es **no ambigua**.

La ambigüedad es un problema que también se da en los lenguajes naturales. Consideremos la frase "Juan ve a un hombre con un telescopio". La parte çon un telescopio"puede describir al hombre que Juan ve o la técnica que utiliza Juan para ver al hombre. Este fenómeno sucede debido a que el lenguaje depende de su estructura y ésta admite más de una descomposición.

Sólo en algunos casos donde se presenta la ambigüedad, se puede encontrar una gramática que produzca el mismo lenguaje pero que no sea ambigua. En caso contrario, es decir, si todas las gramáticas independientes del contexto para un lenguaje son ambiguas, se dice que el lenguaje es un **lenguaje independiente del contexto inherentemente ambiguo**.

Derivaciones por la Izquierda y por la Derecha

Una **derivación por la izquierda** es aquella donde el no terminal que se expande es, siempre, el del extremo izquierdo. Una **derivación por la derecha** es aquella donde el no terminal que se expande es, siempre, el del extremo derecho.

Las gramáticas ambiguas se caracterizan por tener dos (o más) derivaciones por la izquierda (o derecha) para la misma cadena.

Ejemplo

- Pruebe que la siguiente gramática es ambigua
  - $S \rightarrow bA|aB$
  - $A \rightarrow a|aS|bAA$
  - $B \rightarrow b|bS|aBB$