

Factorización de una Gramática libre de Contexto y Eliminación de Recursividad Izquierda de una Gramática

Olger V

Universidad de Costa Rica, Escuela de Ciencias de la Computación e Informática,

San José, Costa Rica

olger031@hotmail.com

Resumen

Este artículo trata de exponer, de manera resumida, dos de los temas analizados en el curso de Autómatas y Compiladores, impartido en la Universidad de Costa Rica, específicamente en la Escuela de Ciencias de la Computación e Informática, durante el segundo semestre del año 2009. Tal artículo se divide en varias secciones las cuales son: una introducción a la temática del artículo y la utilidad de los temas tratados, un desarrollo donde se detallan los temas propuestos, una sección de conclusiones donde se resume la utilidad del contenido de los temas, y una sección final de referencias.

Palabras claves: Gramática, Factorización, Eliminación de la Recursividad.

INTRODUCCIÓN

Ante la importancia que representan los analizadores sintácticos en la creación de compiladores, es indispensable conocer que el principal objeto de su análisis, que es una gramática independiente del contexto, no puede ser interpretada de buena manera, por un analizador, si no sigue un conjunto de restricciones para ello. Tanto la factorización, como la eliminación de la recursividad de una gramática, son métodos que hacen posible que una gramática que no es correcta para un analizador llegue a serlo, el primero se utiliza para hacer posible el análisis sintáctico de tipo predictivo. El segundo es sumamente importante porque los métodos de análisis sintáctico descendente no son capaces de manejar gramáticas recursivas por la izquierda, lo que requiere la eliminación de recursividad. A continuación se tratan ambos métodos, que hacen que una gramática se convierta en una gramática correcta para el análisis sintáctico, se detallan sus casos de uso y sus algoritmos.

DESARROLLO

2.1 FACTORIZACIÓN UNA GRAMÁTICA LIBRE DE CONTEXTO

Como se mencionó anteriormente en la introducción de este artículo, el método de factorización lo que permite es una transformación gramatical útil para el análisis sintáctico de tipo predictivo. Esto ocurre cuando se tienen dos producciones alternativas para un mismo no terminal y no se sabe cual de ellas utilizar para ampliarlo. Un ejemplo que permite visualizar este problema es el representado por las producciones:

$$A \rightarrow \alpha\beta_1 \mid \alpha\beta_2 = A \rightarrow \alpha\beta_1 \\ A \rightarrow \alpha\beta_2$$

Aquí si se pretende expandir el no terminal A, no se sabe si expandirlo a $\alpha\beta_1$ o a $\alpha\beta_2$, ya que ambas empiezan con la misma cadena no vacía α .

Para resolver este problema se plantea un algoritmo general de factorización, este recibe una gramática, y produce como salida una gramática factorizada por la izquierda. Consiste en tomar cada no terminal igual, sea A el no terminal del ejemplo planteado, y encontrar el prefijo más largo que esté presente en todas las producciones para A, es decir, en el ejemplo, el símbolo α , ya que éste es el símbolo común en las dos producciones. Si el prefijo común no es igual a ϵ , que se cumple en el ejemplo pues es α , entonces se sustituye:

$$A \rightarrow \alpha\beta_1 \mid \alpha\beta_2 \mid \dots \mid \alpha\beta_n \mid \gamma \quad , \text{ con } \gamma = \text{todas las alternativas distintas al prefijo común}$$

por las producciones:

$$A \rightarrow \alpha A' \mid \gamma \\ A' \rightarrow \beta_1 \mid \beta_2 \mid \dots \mid \beta_n$$

A' es un nuevo no terminal, este procedimiento se debe aplicar hasta que todos los no terminales iguales cumplan con el hecho de no tener un prefijo en común en sus producciones. En el ejemplo la gramática resultante sería:

$$A \rightarrow \alpha A' \\ A' \rightarrow \beta_1 \mid \beta_2$$

Se omite γ pues no existe en la gramática original, ya que las producciones empiezan con α .

2.2 ELIMINACIÓN DE LA RECURSIVIDAD DE UNA GRAMÁTICA

La necesidad de este método, mencionada en la introducción del artículo, es que el análisis descendente no puede manejar una gramática recursiva.

¿Qué es una gramática recursiva? R/ Es aquella en la que al menos un no terminal deriva en cadenas que empiezan con el mismo no terminal, es decir existen producciones por ejemplo con el no terminal A, en las que van hacia una cadena que empieza con A.

¿Cómo resolver el problema?

R/ Se agrupan todas las producciones de A que existan, las que presentan recursividad y las que no, de la siguiente manera:

$$A \rightarrow A\alpha_1 \mid A\alpha_2 \mid \dots \mid A\alpha_m \mid \beta_1 \mid \beta_2 \mid \dots \mid \beta_n$$

Las producciones de A que van a β son lógicamente las que no empiezan con el mismo símbolo no terminal A. Después de hacer esto se sustituyen las producciones de A, primero las que empiezan con β , por:

$$A \rightarrow \beta_1 A' \mid \beta_2 A' \mid \dots \mid \beta_n A'$$

y después las producciones que tienen recursividad, por:

$$A' \rightarrow \alpha_1 A' \mid \alpha_2 A' \mid \dots \mid \alpha_m A' \mid \varepsilon$$

CONCLUSIONES

Ante la posibilidad, en la vida real, de encontrar gramáticas independientes del contexto, que no se ajusten a los requerimientos de los analizadores sintácticos, es necesario conocer cuales son las formas de lograr que esas gramáticas, lleguen a cumplir esos requerimientos. Los métodos de factorización y eliminación de la recursividad son métodos sencillos y relativamente fáciles de implementar, en el presente artículo se consiguió más que describirlos, llegarlos a entender y saber en que casos son necesarios y cual es el procedimiento que cada uno sigue.

REFERENCIAS

1. Aho, A. Sethi, R. Ullman, J. *Compiladores: Principios, Técnicas y Herramientas*. México: Pearson Educación (1990).