Teoria computacional

Alumno: *Ángel López Manríquez*
Tema: *Limpieza de gramaticas*
Fecha: *17 de junio de 2019*
Grupo: *2CV1*

Prof. Luz Maria Sanchez Garcia

Instituto Politecnico Nacional
Escuela Superior de Cómputo

Índice

Limpieza de gramaticas

Lopez Manriquez Angel 2CM5

17 de junio de 2019

1. Introduccion

Las gramáticas de los lenguajes de programación están formadas por un conjunto de reglas BNF, cuyo número suele ser bastante amplio, lo cual incide en la ocultación de distintos problemas que pueden producirse, tales como

- tener reglas que produzcan símbolos que no se usen después
- cadenas inalcanzables

Todo esto se puede solventar realizando la transformación de la gramática inicial "sucia" a una gramática "limpia".

2. Definiciones

- **Símbolo muerto** (superfluo): es un símbolo no terminal que no genera ninguna cadena de símbolos terminales.
- **Símbolo vivo** : es un símbolo no terminal del cual se puede derivar una cadena de símbolos terminales. Todos los símbolos terminales son símbolos vivos. Es decir son símbolos vivos lo que no son muertos.
- **Símbolo inaccesible** : es un símbolo no terminal al que no se puede llegar por medio de producciones desde el símbolo inicial.
- **Símbolo accesible** : es un símbolo que aparece en una cadena derivada del símbolo inicial. Es decir, aquel símbolo que no es inaccesible.
- **Símbolo extraño** : se denomina así a todo símbolo muerto o inaccesible.
- **Gramática sucia** : es toda gramática que contiene símbolos extraños.
- **Gramática limpia** : es toda gramática que no contiene símbolos extraños.

toda gramática en bruto ha de limpiarse con el objetivo de eliminar todos los símbolos extraños.

El método de limpiar las gramáticas sucias consiste en detectar en primer lugar todos los símbolos muertos, y a continuación se detectan todos los símbolos inaccesibles. Es importante seguir este orden, puesto que la eliminación de símbolos muertos puede generar nuevos símbolos inaccesibles.

Los algoritmos que se utilizan en la limpieza de gramáticas se basan en los teoremas que se enuncian a continuación

3. Teoremas

3.1. Teorema de los símbolos vivos

Si todos los símbolos de la parte derecha de una producción son vivos, entonces el símbolo de la parte izquierda también lo es.

3.2. algoritmo

Algoritmo para detectar simbolos vivos

1. Hacer una lista de no terminales que tengan al menos una producción sin símbolos no terminales en la parte derecha.
2. Dada una producción, si todos los no-terminales de la parte derecha pertenecen a la lista, entonces podemos incluir al no terminal de la parte izquierda.
3. Cuando no se puedan incluir más símbolos mediante la aplicación del paso 2, la lista contendrá todos los símbolos vivos, el resto serán muertos.

3.3. Teorema de los simbolos accesibles

Si el símbolo no terminal de la parte izquierda de una producción es accesible, entonces todos los símbolos de la parte derecha también lo son.

3.4. algoritmo

Algoritmo para detectar simbolos accesibles

1. Se comienza la lista con un único no terminal, el símbolo inicial.
2. Si la parte izquierda de la producción está en la lista, entonces se incluyen en la misma a todos los no terminales que aparezcan en la parte derecha.
3. Cuando ya no se puedan incluir más símbolos mediante la aplicación del paso 2 , la lista contendrá todos los símbolos accesibles, y el resto será inaccesible.

4. Análisis automático de la limpieza de gramáticas

Una gramatica esta limpia si no tiene

1. símbolos muertos
2. símbolos inaccesibles
3. reglas innecesarias

Una gramatica esta bien formada si

1. Está limpia.

- Sin símbolos muertos
- Sin símbolos inaccesibles
- Sin reglas innecesarias

2. No tiene reglas no generativas ($A \to \epsilon, A \neq S$).

3. No tiene reglas de redenominación ($A \to B$).

En algunas ocasiones es imprescindible que las gramáticas se hallen dispuestas de una forma especial. Es decir, se trata de obtener una gramática equivalente, que genera el mismo lenguaje, pero que debe cumplir unas especificaciones determinadas.

Mas adelante en el curso se hablara acerca de otras formas normales como la de Chomsky o Greibach, asi como practicas.