Valentina Balderas Torres

Wilialdo Oceguera Pérez

Paulina Osuna Guzmán

Práctica 5 - Recursividad

Objetivo: Revisar la gramática de su propuesta de lenguaje para asegurar la generación de árboles sintácticos por el método descendente.

Instrucciones:

1. Escriba la gramática completa de su lenguaje y valide que pueda generar todas las cadenas que desea analizar en su código.

*programa 🡪* **inicio** *listaInstr* **fin**

*listaInst 🡪 instr listaInstr | instr | Ɛ*

*instr 🡪 id = numero |if comparacion instr else instr | if comparacion instr else listaInstr |* **mostrar** *opcion |* **keyboard** *id |*

*comparacion 🡪 opcion op opcion*

*op 🡪* < | > | == |! =| <= | >=

*opcion 🡪 text | id | numero*

1. Factorice la gramática y especifique cuáles sentencias se “redujeron” o aclare que no fue necesario demostrando por qué.

*programa* 🡪**inicio** *listaInstr* **fin**

*listaInstr* 🡪 *instr listainstr’ | Ɛ*

*listainstr’* 🡪 *listaInstr | Ɛ*

*instr* 🡪 *id =numero | if comparacion instr else instr’ |mostrar opcion | keyboard id*

*instr’* 🡪 *instr | listaInstr*

*comparacion* 🡪 *opcion op opcion*

*op 🡪* < | > | == |! =| <= | >=

*opcion 🡪 text | id | numero*

1. Elimine la recursividad por la izquierda y escriba la gramática ampliada. Si no presenta recursividad especifíquelo.

*No tiene recursividad por la izquierda.*

1. Asegúrese de que la gramática final no es ambigua eligiendo dos sentencias: una sencilla y una semi compleja (condición, repetición) para su representación. Aunque eso no lo asegure del todo, será suficiente para la práctica.