Tecnológico de Monterrey

Tarea 5
Diseño de Compiladores

Para esta tarea es necesario que tengan funcionando las siguientes herramientas: flex, bison y gcc. La tarea deben entregarla el jueves 28 de mayo de 2020. Debe estar documentada y debe dar instrucciones de cómo compilar y correr el programa. Debe decir también qué se espera como salida.

Problema 1

El objetivo de esta tarea es que construyan un intérprete del lenguaje cuya gramática se encuentra más abajo en este documento. El intérprete debe hacerse usando flex, bison y el lenguaje de programación C y debe ser una extensión del reconocedor de la tarea 4.

Para hacer el intérprete primero debe construir el árbol sintáctico reducido, tal como se discutió en clase. La interpretación se debe hacer recorriendo el árbol sintáctico reducido y usando la tabla de símbolos como forma de cambiar los valores de las variables involucradas, también como se discutió en clase.

El intérprete debe recibir como entrada un archivo de texto que contenga un programa en el lenguaje de programación descrito y la salida debe ser la interpretación del programa evaluado, si no hay errores en la parte del reconocimiento sintáctico. En caso de haber errores, se debe reportar y terminar la ejecución del intérprete. Parte de la salida puede ser la tabla de símbolos como queda al final de la interpretación.

Lo que aparece en ***negritas*** son los símbolos terminales, y obviamente se refiere a lo que debe reconocer el reconocedor léxico. El reconocedor léxico deben hacerlo usando *flex*.

$$
\begin{array}{lcl}
prog & \rightarrow & \textbf{program id \{ } opt\_decls \textbf{ \}} \ stmt \\
opt\_decls & \rightarrow & decls \mid \varepsilon \\
decls & \rightarrow & dec \ \textbf{;} \ decls \mid dec \\
dec & \rightarrow & \textbf{var id:} \ tipo \\
tipo & \rightarrow & \textbf{int} \mid \textbf{float} \\
stmt & \rightarrow & assign\_stmt \\
 & \mid & if\_stmt \\
 & \mid & iter\_stmt \\
 & \mid & cmp\_stmt \\
assig\_stmt & \rightarrow & \textbf{set id} \ expr\textbf{;} \\
 & \mid & \textbf{read} \ id\textbf{;} \\
 & \mid & \textbf{print} \ expr\textbf{;} \\
if\_stmt & \rightarrow & \textbf{if (}expresion\textbf{)} \ stmt \\
 & \mid & \textbf{ifelse (}expresion\textbf{)} \ stmt \ stmt \\
iter\_stmt & \rightarrow & \textbf{while (}expresion\textbf{)} \ stmt \\
 & \mid & \textbf{for set id} \ expr \ \textbf{to} \ expr \ \textbf{step} \ expr \ \textbf{do} \ stmt \\
cmp\_stmt & \rightarrow & \textbf{\{ \}}|\textbf{\{} \ stmt\_lst \ \textbf{\}} \\
stmt\_lst & \rightarrow & stmt \mid stmt\_lst \ stmt \\
expr & \rightarrow & expr + term \\
 & \mid & expr \ \textbf{-} \ term \\
 & \mid & term \\
term & \rightarrow & term \ \textbf{*} \ factor \\
 & \mid & term \ / \ factor \\
 & \mid & factor \\
factor & \rightarrow & (\ expr\ ) \\
 & \mid & \textbf{id} \mid \textbf{numi} \mid \textbf{numf} \\
expresion & \rightarrow & expr < expr \\
 & \mid & expr > expr \\
 & \mid & expr = expr \\
 & \mid & expr <= expr \\
 & \mid & expr >= expr
\end{array}
$$