

# Tarea 4

## Compiladores

Alberto Oliart

marzo 2016

Para esta tarea es necesario que tengan funcionando las siguientes herramientas: flex y bison. La tarea deben entregarla el jueves 7 de abril de 2015.

### Problema 1

El objetivo de este ejercicio es hacer un reconocedor del lenguaje generado por la gramática presentada más abajo.

Lo que aparece en ***negritas*** son los símbolos terminales y se refiere a lo que debe reconocer el reconocedor léxico. El reconocedor léxico deben hacerlo usando *flex*. Las expresiones regulares que determinan **id** y **num** son las siguientes:

- **id** =  $[a - z, A - Z]([0 - 9][a - z, A - Z])^*$
- **num** =  $[0 - 9][0 - 9]^*$
- **numf** =  $[0 - 9][0 - 9]^*.[0 - 9][0 - 9]^*$

El reconocedor debe construir el árbol sintáctico que represente al programa usando *bison*.

Se debe construir una tabla de símbolos con las variables que se declaran. El reconocedor debe reportar como error el uso de identificadores no declarados. Debe reportar también como errores los errores de por tipo. Esto es, el lenguaje es un lenguaje con un sistema de tipos fuerte que no permite la operación entre expresiones o variables de diferentes tipos.

Al finalizar el reconocimiento el programa debe mostrar la tabla de símbolos y debe recorrer el árbol sintáctico mostrando cómo está representado el programa que acaba de reconocer.

<i>prg</i>	→	<i>opt_decls</i> <b>begin</b> <i>opt_stmts</i> <b>end.</b>
<i>opt_decls</i>	→	<i>decls</i>   $\varepsilon$
<i>decls</i>	→	<i>decls</i> ; <i>dec</i>   <i>dec</i>
<i>dec</i>	→	<i>tipo</i> <b>id</b>
<i>tipo</i>	→	<b>int</b>   <b>float</b>
<i>stmt</i>	→	<b>id</b> := <i>expr</i>   <b>if</b> <i>expresion</i> <b>then</b> <i>stmt</i>   <b>while</b> <i>expresion</i> <b>do</b> <i>stmt</i>   <b>repeat</b> <i>stmt</i> <b>until</b> <i>expresion</i>   <b>begin</b> <i>opt_stmts</i> <b>end</b>   <b>print</b> <i>expr</i>   <b>read</b> <i>id</i>
<i>opt_stmts</i>	→	<i>stmt_lst</i>   $\varepsilon$
<i>stmt_lst</i>	→	<i>stmt_lst</i> ; <i>stmt</i>   <i>stmt</i>
<i>expresion</i>	→	<i>expr</i> < <i>expr</i>   <i>expr</i> > <i>expr</i>   <i>expr</i> = <i>expr</i>
<i>expr</i>	→	<i>expr</i> + <i>term</i>   <i>expr</i> - <i>term</i>   <i>term</i>
<i>term</i>	→	<i>term</i> * <i>factor</i>   <i>term</i> / <i>factor</i>   <i>factor</i>
<i>factor</i>	→	( <i>expr</i> )   <b>id</b>   <b>nument</b>   <b>numf</b>