

# Tarea 3

## Diseño de Compiladores

Alberto Oliart

marzo 2015

Para esta tarea es necesario que tengan funcionando las siguientes herramientas: flex y gcc. La tarea deben entregarla el martes 10 de marzo de 2015.

### Problema 1

El objetivo del ejercicio es que construyan un reconocedor sintáctico descendente predictivo de una gramática que genera un lenguaje de programación muy básico. La gramática es la siguiente:

$stmt$	$\rightarrow$	$id := expr$
		$if\ expr\ then\ stmt$
		$while\ expr\ do\ stmt$
		$begin\ opt\_stmts\ end$
$opt\_stmts$	$\rightarrow$	$stmt\_lst \mid \varepsilon$
$stmt\_lst$	$\rightarrow$	$stmt ; stmt\_lst \mid stmt$
$expr$	$\rightarrow$	$expr + term$
		$expr - term$
		$term$
$term$	$\rightarrow$	$term * factor$
		$term / factor$
		$factor$
$factor$	$\rightarrow$	$(\ expr\ )$
		$id$
		$num$

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*. Las expresiones regulares que determinan **id** y **num** son las siguientes:

- **id** =  $[a - z, A - Z]([0 - 9][a - z, A - Z])^*$
- **num** =  $[1 - 9][0 - 9]^*$

Se asume que todos los números serán números enteros, para facilitar el ejercicio.

El primer paso es construir una versión **LL(1)** de la gramática. En clase hicimos parte de este trabajo, sólo deben terminarlo para esta caso particular.