

# Lenguajes y Compiladores

## Laboratorio 1 - 2013

March 24, 2013

Al terminar este laboratorio habremos definido en Haskell primero un tipo de dato para representar la gramática abstracta del lenguaje imperativo, y en segundo lugar, un parser para construir elementos de ese tipo de datos a partir de una cadena de símbolos.

### i Generalidades

**Cómo instalarlo** Usar el administrador de paquetes de tu sistema operativo o bajar “The Haskell Platform”.

**Qué leer para recordar** “Real World Haskell”, “StackOverflow” (tag “Haskell”).

**Entregas** Las entregas serán por mail a las direcciones `miguel.pagano@gmail.com` y `hector.gramaglia@gmail.com` hasta las 23:59 (hora argentina) del día correspondiente. El asunto del mail tiene que ser

LyC 2013-*i*: *Apellido, Nombre*

donde *i* es el número de laboratorio. Se debe adjuntar al mail un archivo `.tar.bz2`<sup>1</sup> que contenga un directorio con su apellido y nombre (sin espacios) y que tenga la estructura de directorios de la primera entrega.

### ii Tarea

Descargar el archivo `lab1.tar.gz` de la wiki, allí se encuentra un ejemplo de un tipo de datos para expresiones enteras y booleanas y un parser para ellas. La tarea consiste en definir tipos de datos para representar el lenguaje definido por la siguiente gramática concreta (notar que el tipo de datos no debe reflejar las sintacticalidades concretas que facilitan el parseo, por ejemplo el `fi` en la producción del condicional). Luego se deben desarrollar un lexer y un parser para ese lenguaje, para estas tareas se debe utilizar la librería `parsec`.

---

<sup>1</sup>para crearlo `$ tar jcf apellido.tar.bz2 ApellidoNombre`

$\langle \text{intexp} \rangle ::= \langle \text{var} \rangle \mid \mathbf{0} \mid \mathbf{1} \mid \dots$   
 $\quad \mid \langle \text{intexp} \rangle * \langle \text{intexp} \rangle \mid \langle \text{intexp} \rangle / \langle \text{intexp} \rangle \mid \langle \text{intexp} \rangle \% \langle \text{intexp} \rangle$   
 $\quad \mid -\langle \text{intexp} \rangle \mid \langle \text{intexp} \rangle + \langle \text{intexp} \rangle \mid \langle \text{intexp} \rangle - \langle \text{intexp} \rangle$   
 $\langle \text{assert} \rangle ::= \mathbf{true} \mid \mathbf{false}$   
 $\quad \mid \langle \text{intexp} \rangle = \langle \text{intexp} \rangle \mid \langle \text{intexp} \rangle \neq \langle \text{intexp} \rangle \mid \langle \text{intexp} \rangle < \langle \text{intexp} \rangle$   
 $\quad \mid \langle \text{intexp} \rangle \leq \langle \text{intexp} \rangle \mid \langle \text{intexp} \rangle > \langle \text{intexp} \rangle \mid \langle \text{intexp} \rangle \geq \langle \text{intexp} \rangle$   
 $\quad \mid \neg \langle \text{assert} \rangle \mid \langle \text{assert} \rangle \wedge \langle \text{assert} \rangle \mid \langle \text{assert} \rangle \vee \langle \text{assert} \rangle \mid$   
 $\quad \mid \langle \text{assert} \rangle \Rightarrow \langle \text{assert} \rangle \mid \langle \text{assert} \rangle \Leftrightarrow \langle \text{assert} \rangle$   
 $\langle \text{comm} \rangle ::= \mathbf{skip} \mid \langle \text{var} \rangle := \langle \text{intexp} \rangle \mid \langle \text{comm} \rangle ; \langle \text{comm} \rangle$   
 $\quad \mid \mathbf{if} \langle \text{assert} \rangle \mathbf{then} \langle \text{comm} \rangle \mathbf{else} \langle \text{comm} \rangle \mathbf{fi}$   
 $\quad \mid \mathbf{newvar} \langle \text{var} \rangle := \langle \text{intexp} \rangle \mathbf{in} \{ \langle \text{comm} \rangle \}$   
 $\quad \mid \mathbf{while} \langle \text{assert} \rangle \mathbf{do} \langle \text{comm} \rangle \mathbf{od}$