Propuesta de Tesis de Licenciatura

λPage

Un bloc de notas para desarrolladores Haskell

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Alumno

Fernando Benavides

Libreta Universitaria 470/01

greenmellon@gmail.com

Directores

Dr. Diego Garbervetsky

Lic. Daniel Gorín

Índice

1. Motivación

Actualmente, gracias al éxito de varios proyectos desarrollados en Erlang, como CouchDB, ejabberd y el chat de Facebook, los lenguajes del paradigma funcional han ganado un renovado interés entre los desarrolladores.

Si bien como lenguajes, tanto Haskell como Erlang son maduros, confiables y presentan abundantes ventajas en comparación con los lenguajes tradicionales del paradigma imperativo, aquellos desarrolladores que deciden hacer el cambio de paradigma se encuentran generalmente con el problema de la falta de ciertas herramientas a las que ya están acostumbrados.

La comunidad Haskell ha generado muchas herramientas (algunas de ellas, incluso, no habituales en los demás lenguajes) para ayudar al programador. Entre ellas podemos encontrar:

Cabal Un sistema de construcción y generación de paquetes de librerías y ejecutables hechos en Haskell. Permite distribuir, catalogar y organizar aplicaciones de manera consistente y portable.

Cabal-install Una aplicación que permite instalar paquetes en formato *Cabal* verificando sus dependencias y realizando las acciones necesarias para su compilación.

Hackage Un repositorio centralizado de paquetes haskell. Cada paquete se encuentra en formato *Cabal* y puede ser instalado utilizando *Cabal-install*.

Darcs Un sistema de administración de la configuración distribuido. Muy similar a *git* pero orientado a proyectos hechos en haskell.

Haddock Una herramienta para generar documentación a partir del código. Integrada con *Cabal* y *Hackage* permite publicar documentación para las librerías generadas en la propia página web de *Hackage*.

Hayoo! Un sitio web en el que se puede navegar toda la documentación generada con *Haddock* sobre los paquetes *Cabal* subidos a *Hackage*

Sin embargo, lenguajes como *Objective-C*, *Java*, *C#* o *SmallTalk* proveen otras herramientas con las que los desarrolladores haskell no cuentan aún, como:

- Entornos de desarrollo de aplicaciones con, por ejemplo, *IntelliSense* (Por ejemplo: XCode para Objective-C, Eclipse para Java, etc.)
- Herramientas para el diseño de interfaces gráficas (Por ejemplo: Interface Builder para Objective-C, Visual Studio para C#, etc.)
- Herramientas para la ejecución y testeo de pequeñas porciones de código (Por ejemplo: jPage para Java, Workspace para Smalltalk, etc.)

En otro orden de cosas, cabe notar que haskell es un lenguaje que requiere un gran nivel de abstracción por parte del programador. Es considerado un lenguaje "write-only", en el sentido de que, cuando un programador lee un código haskell escrito hace suficiente tiempo (aún si lo escribió él mismo) es probable que le cueste mucho entender qué es lo que ese código hace y cómo. Para descifrarlo, el enfoque usual es recurrir a la consola *ghci* o *hugs* e ir realizando pequeñas pruebas sobre distintas porciones del código que permitan comprender gradualmente el código completo.

2. Propuesta de Tesis

2.1. Objetivo

El objetivo de esta tesis es brindar a los desarrolladores haskell una herramienta similar al Workspace de Smalltalk, que les permita trabajar con pequeñas porciones de código haskell, evaluarlas, observar su tipo y su clase.

En el espíritu de las antes mencionadas herramientas provistas por la comunidad de desarrolladores haskell, es nuestra idea que *λPage*, tal es el nombre de la herramienta que nos proponemos crear, sea desarrollada en haskell y se integre con *Cabal*, *Hayoo!* y demás herramientas ya existentes.

2.2. Plan de Trabajo

Paso a paso, cómo lo hicimos, ponele

A. Referencias

A hint, wxhaskell, etc...