



16-6-2010

11:30  
Aula 13  
Facultad de  
Informática

Kostis Sagonas is Associate Professor at School of Electrical and Computer Engineering of National Technical University of Athens. Before joining NTUA he has been on the Dept. of Information Technology of Uppsala University, Sweden where he still maintains a part time position. His current research interests are in the area of programming languages and systems; mostly on implementation of functional and logic programming languages. He has worked on Prolog, being one of the main developers of XSB, a logic programming system integrating ideas from deductive databases, logic programming, and non-monotonic reasoning. He has led the development of HiPE (High Performance Erlang), the native code compiler for Erlang, and developed Dialyzer, a static analysis tool that detects type and concurrency errors in Erlang applications, and Tidier, an automatic refactoring tool for Erlang programs.

# SEMINARIO

## Departamento de Sistemas Informáticos y Computación

Facultad de Informática  
Universidad Complutense de Madrid

Using Static Analysis to Detect  
Type and Concurrency Errors  
in Erlang Programs

Konstantinos Sagonas  
National Technical University of Athens

We will present the key ideas in the design and implementation of Dialyzer, a static analysis tool that detects bugs in Erlang programs. Dialyzer started as a defect detection tool using a rather ad hoc dataflow analysis to detect type errors in Erlang programs, but it adopted a more disciplined approach to detecting definite type clashes in dynamically typed languages. Namely, an approach based on using a constraint-based analysis to infer success typings which are enhanced with optional contracts supplied by the programmer.

In the first part of the talk, we describe this constraint-based approach to type inference and explain how it differs with past and recent attempts to type check programs written in dynamic languages. In the second part, we will present important recent additions to Dialyzer, namely analyses that detect concurrency defects (such as race conditions) in Erlang programs.

For a number of years now, Dialyzer has been part of the Erlang/OTP system and has been actively used by its community. Based on this experience, we will also critically examine Dialyzer's design choices, show interesting cases of Dialyzer's use, and distill the main lessons learned from using static analysis in open source as well as commercial code bases of significant size.