

Report on the Visit to Department of Computer Sciences, Yale University

Wenbiao Wu
ESD Lab, Department of Electronics
Royal Institute of Technology
Sweden

1. Introduction

Department of Computer Science at Yale University has been an integral part of the development and implementation of the Haskell programming language since its inception. They have helped to design Hugs (The Haskell Interpreter)^[1], write the Haskell Report^[2] (the document defining the language), write the tutorial on Haskell^[3], and most recently Professor Paul Hudak has published a textbook using Haskell: The Haskell School of Expression.^[4]

From March 5th to May 27th, I had a study visit to this Department, in particularly Professor Paul Hudak's Haskell research group. The purpose of this visit is to establish a research contact and to utilize that groups long and profound experience in functional languages and program transformations.

2. Department of Computer Science, Yale University

There are mainly four research groups at Computer Science Department. Haskell research group, FLINT research group, Linda research group and Computational Vision and Control group. Here I'd like to give a brief introduction of each group.

Haskell Group^[5]

Professor Paul Hudak heads the Haskell group in Yale. Today the focus of their group is on putting Haskell to work in the real world and they have used Haskell to construct a variety of Domain-Specific Embedded Languages: languages that use Haskell as a framework within which to build abstractions specific to an underlying domain. The followings are the active projects in this group at the moment:

- **Functional Reactive Programming (FRP)**

FRP^[6] describes interactive systems containing both continuous elements, or behaviors, and discrete elements, or events. Ongoing projects include investigating the core semantics of FRP and implementations of FRP.

- **Robocup 2000**

They are entering a team in the Robocup Robotic Soccer Tournament (www.robocup.org) and currently they are building hardware (small robots controlled via a radio link) and software. This system uses Frob (Functional Robotics), a DSL (Domain-Specific Embedded Languages) for robot control, to drive the robots.

- **Mobile Autonomous Robotics**

This research covers a wide variety of problems, from software methodologies for low level control systems design to on-board vision processing to planning and cooperation. Frob^[7] is used in this research to building a set of simple autonomous robots.

- **Computer Vision**

In collaboration with Greg Hager (Johns Hopkins) they are developing a Haskell-based DSL for computer vision. They have built a library of Haskell types and functions called FVision that provides a high-level interface to the lower-level C++ library code (XVision). In addition, They use Functional Reactive Programming (FRP) to express interaction in a purely functional manner. The resulting system demonstrates the viability of mixed-language programming: visual tracking programs continue to spend most of their time executing low-level image-processing code, while Haskell's advanced features allow them to develop and test systems quickly and with confidence.

- **Multi-Media**

They are working with two multi-media DSLs: Fran^[8], a system for describing functional animation, and Haskore^[9], a language of computer music. They use FRP to build richly interactive systems, adding interactive graphics and computer vision to the traditional vocabulary of interaction.

FLINT Project^[10]

This project focuses on building a state-of-the-art systems environment for multiple HOT languages (Higher-Order and Typed). The key idea is the use of a common typed intermediate language (named FLINT) to model the semantics and interactions of various HOT features. FLINT is based on a predicative variant of Girard and Reynolds's polymorphic lambda calculus (Fomega), extended with a very rich set of primitive type constructors and functions. FLINT provides a common compiler backend that can be quickly adapted to generate compilers for new HOT languages. Also, with its single unified type system, FLINT serves as a great platform for reasoning about cross-language interoperations. FLINT types act as a glue to connect language features that complicate interoperability, such as mixed data representations,

multiple function calling conventions, and different memory management protocols. This project is headed by Professor Zhong Shao.

Linda Research Group:^[11]

This group explores a variety of research topics in parallel and distributed computing, adaptive computation, parallel programming languages, groupware and information systems. Much of this work centers around the Linda programming language or ideas that have been spawned from work on the Linda system.

Computational Vision and Control (CVC)^[12]

The CVC exists as a collaboration between researchers in several departments, especially Computer Science, Electrical Engineering, and Diagnostic Radiology. Their research includes image process and pattern recognition, AI, computer vision and control etc. Concurrently, there are a couple of research projects ongoing in the group.

3. My activities at Yale

- Course study

I attended one course (CS 430/530 Formal Semantics) at the department. This course is based on John Renolds' newest semantics book: Theories of Programming Languages, Cambridge University Press, 1998, Robert Harper's draft notes on Type Systems for Programming Languages and Benjamin C. Pierce's draft notes on Type Systems. The main topics in this courses includes Lambda calculus, semantics for imperative languages and functional languages, type systems and so on.

- Attending two workshops

During my stay, there was one workshop held by New Jersey Programming Languages and Systems Seminar at IBM Thomas J. Watson Research Center. The New Jersey Programming Languages and Systems Seminar Series is an informal forum that promotes interaction amongst programming languages and systems researchers in the New Jersey area. The series provides an opportunity to present unfinished research-in-progress and receive feedback and constructive criticism. There are 6 presentations at that day and most of the presentation focused on Java, Haskell and ML. There is also a homepage for this workshop^[13].

Another workshop that I attended was sponsor by the department. The title of this workshop is Programming Languages: Theory vs. Practice. The four lectures are:

- James Gosling (from Sun)
Title: Java: an Exercise in Research Avoidance

- Robert Harper (from Carnegie Melon Univ.)
Title: The Practice of Type Theory
- Randal L. Schwartz (from Stonehenge Consulting Services, Inc.)
Title: Perl: the 'P' stands for 'Practical'
- Phil Wadler (from Bell Labs)
Title: From Frege to Gosling: 19'th century logic and 21'st century programming languages

The homepage for this workshop is still available at [14].

- Mark Tullsen's PATH and the Ultra tool

Mark Tullsen is Professor Hudak's PhD student. His research interests include functional programming and program transformation. And currently he is in the process of building a program transformation system for Haskell called PATH (Programmer Assistant for Transforming Haskell). An intermediate-meta language is constructed and used for the transformation in PATH. The basic principle underlying his approach is that it has integrated different approaches i.e. Squiggol, Unfold/fold and expression procedures. ^{[15][16]}

Mark also recommended me to use the tool Ultra^[17] which is provided by University of Ulm at Germany. Ultra is an interactive program transformation system to transform Haskell programs. As related to our work, we may use part of it for our interactive transformation tools in the future.

- Learning Monadic programming

Monads are widely used in Haskell programming community. Both PATH and ULTRA system are based on monadic programming style. The novel feature of the monadic approach is that side-effects can be incorporated into a pure functional language by means of monads which are mathematical notions from category theory and represent computations. So I spent quite a long time at Yale to understand what monad is. Three articles that I studied during the time are [18][19][20].

4. Conclusion

Three months' stay at Yale is not a long time. The most immediate benefit of this visit is that our project has been strengthened because it let us know what other people are doing research in this area and also establish a research contact for future work. Yale is one of the most important research labs in computer science in the US, and some of their ideas may be very influential in the near future. The established research contact with Yale Haskell group will hopefully be useful for me, KTH and

the SAVE project. And finally, the course at Yale is also very good; it broadened my knowledge in the area of programming languages.

5. References

- [1] Hugs: <http://www.haskell.org/hugs/>
- [2] Haskell Report: <http://www.haskell.org/definition/>
- [3] Haskell Tutorial: <http://www.haskell.org/tutorial/>
- [4] The Haskell School of Expression: Learning Functional Programming through Multimedia
<http://www.haskell.org/soe/>
- [5] Yale Haskell Group: <http://www.cs.yale.edu/haskell/yale-fp.html>
- [6] Zhanyong Wan and Paul Hudak, Functional Reactive Programming from First Principles Proc. 2000 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '00), pp. 240 -- 250. (<http://pantheon.yale.edu/~zw23/frp-1st.ps.gz>)
- [7] Haskell for Vision and Robotics: <http://www.haskell.org/frob/>
- [8] Functional Reactive Animation: <http://www.research.microsoft.com/~conal/Fran/>
- [9] The Haskore Computer Music System: <http://www.haskell.org/frob/haskore/>
- [10] FLINT Project: <http://flint.cs.yale.edu/>
- [11] Linda Project: <http://www.cs.yale.edu/Linda/linda.html>
- [12] CVC Group at Yale: <http://cvc.yale.edu/cvctest/index.html>
- [13] <http://www.research.ibm.com/compsci/plansoft/conferences/plday.html>
- [14] Programming Languages --- Theory vs. Practice:
<http://www.cs.yale.edu/admin/Files%20not%20in%20Use/perlis00.html>
- [15] Mark Tullsen and Paul Hudak. Shifting Expression Procedures Into Reverse. Proceedings of the ACM SIGPLAN Workshop on Partial Evaluation and Semantics-Based Program Manipulation, Technical report BRICS-NS-99-1, University of Aarhus, pages 95--104, San Antonio, Texas, January 1999. <http://cs-www.cs.yale.edu/homes/tullsen/shifting.ps>
- [16] Mark Tullsen and Paul Hudak. An Intermediate Meta-language for Program Transformation. Technical Report YALEU/DCS/RR-1154, Yale University, June 1998. <http://cs-www.cs.yale.edu/homes/tullsen/imlforpt.ps>
- [17] The ULTRA Home Page: <http://www.informatik.uni-ulm.de/pm/ultra/>
- [18] Graham Hutton and Erik Meijer, Monadic parser combinators, Technical report NOTTCS-TR-96-4, Department of Computer Science, University of Nottingham, 1996.
<http://www.cs.nott.ac.uk/~gmh/monparsing.ps>
- [19] Graham Hutton and Erik Meijer, Monadic parsing in Haskell, Journal of Functional Programming, 8(4):437-444, Cambridge University Press, July 1998. <http://www.cs.nott.ac.uk/~gmh/pearl.ps>
- [20] Philip Wadler, Monads for functional programming, *Advanced Functional Programming*, Springer Verlag, LNCS 925, 1995. <http://cm.bell-labs.com/cm/cs/who/wadler/papers/marktoberdorf/marktoberdorf.ps>