COMPUTER SCIENCE
North Haugh, St Andrews, Fife, KY16 9SX Scotland
Tel: (01334) 463253 Fax: (01334) 463278
www.cs.st-andrews.ac.uk/

DISTINGUISHED LECTURE SERIES

Semester 1

2010/11

Machines Reasoning about Machines

By

J Strother Moore

Department of Computer Sciences
University of Texas at Austin

Monday 15 November 2010

School of Computer Science (Room 133a/b), Jack Cole Building, North Haugh, St Andrews
Physics Building (Lecture Theatre B), North Haugh, St Andrews

---

Biography

J Strother Moore holds the Admiral B.R. Inman Centennial Chair in Computing Theory at the University of Texas at Austin. He is also a Visiting Professor at the University of Edinburgh, where he spends several months each year. He is the author of many books and papers on automated theorem proving and mechanical verification of computing systems. Along with Boyer he is a co-author of the Boyer-Moore theorem prover and the Boyer-Moore fast string searching algorithm. With Matt Kaufmann he is the co-author of the ACL2 theorem prover. Moore got his SB from MIT in 1970 and his PhD from the University of Edinburgh in 1973. Moore was a founder of Computational Logic, Inc., and served as its chief scientist for ten years. He served as chair of the UT Austin CS department for eight years. He and Bob Boyer were awarded the McCarthy Prize in 1983 and the Current Prize in Automatic Theorem Proving by the American Mathematical Society in 1991. In 1999, they were awarded the Herbrand Award for their work in automatic theorem proving. Boyer, Moore, and Kaufmann were awarded the 2005 ACM Software Systems Award for the Boyer-Moore theorem prover. Moore is a Fellow of both the American Association for Artificial Intelligence and the ACM and is a member of the National Academy of Engineering.

Programme

Abstract

Computer hardware and software can be modeled precisely in mathematical logic. If expressed appropriately, these models can be executable. The ``appropriate'' logic is an axiomatically formalized functional programming language. This allows models to be used as simulation engines or rapid prototypes. But because they are formal they can be manipulated by symbolic means: theorems can be proved about them, directly, with mechanical theorem provers. But how practical is this vision of machines reasoning about machines?

In this highly personal talk, I will describe the 40 year history of the ``Boyer-Moore Project'' and discuss progress toward making formal verification practical. Among other examples I will describe important theorems about commercial microprocessor designs, including parts of processors by AMD, Motorola, IBM, Rockwell-Collins and others. Some of these microprocessor models execute at 90% the speed of C models and have had important functional properties verified. In addition, I will describe a model of the Java Virtual Machine, including class loading and bytecode verification and the proofs of theorems about JVM methods. In the latter half of this 3-hour seminar we will look closely at how such machines are formalized and how the theorem prover is ``taught'' to reason about them, by looking at simpler examples drawn from list processing and a ``toy'' version of the JVM.

Monday 15 November 2010

10.30 – 11.00	Coffee & Tea with Biscuits
School of Computer Science , Jack Cole Building, Common area.	Break
11.00 – 12.00	**Lecture 1:** Introductory
School of Computer Science, Jack Cole Building, **R:133a/b**, North Haugh	I will briefly explain how the theorem prover works and illustrate some of its important industrial applications. We will do this by quickly surveying the 40 year history of the Boyer-Moore project, touching on the highpoints of each decade.
14.00 – 15.00	**Lecture 2**
Physics Building, Lecture Theatre B, North Haugh	In this part of the talk, I will focus on how we formalize the operational semantics of machines and how we configure the theorem prover to reason about the programming language defined by the semantics. I will use a very simple (``toy'') model of the JVM as my target machine.
15.00 – 15.30	Coffee & Tea with Biscuits
Physics Building, Lecture Theatre B, North Haugh	Break
15.30 – 16.30	**Lecture 3**
Physics Building, Lecture Theatre B, North Haugh	During the third hour I will show how the user of the theorem prover interacts with the system to construct proofs that the system cannot discover by itself. I will start with a very simple list processing theorem to illustrate the basic user behavior and then we'll jointly tackle a another list processing theorem.