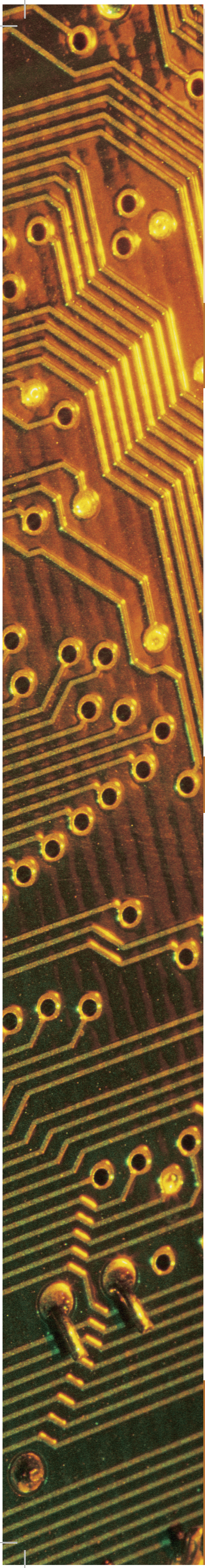

University of Wisconsin
LA CROSSE

Distinguished Lecture Series in Computer Science

Monday, Oct. 24, 2011

Leslie Valiant

was educated at King's College, Cambridge; Imperial College, London; and at Warwick University where he received his Ph.D. in computer science in 1974. He is currently T. Jefferson Coolidge Professor of Computer Science and Applied Mathematics in the School of Engineering and Applied Sciences at Harvard University, where he has taught since 1982. Before coming to Harvard he had taught at Carnegie Mellon University, Leeds University, and the University of Edinburgh.

His work has ranged over several areas of theoretical computer science, particularly complexity theory, computational learning, and parallel computation. He also has interests in computational neuroscience, evolution and artificial intelligence.

He received the Nevanlinna Prize at the International Congress of Mathematicians in 1986, the Knuth Award in 1997, the European Association for Theoretical Computer Science EATCS Award in 2008, and the 2010 A. M. Turing Award. He is a Fellow of the Royal Society (London) and a member of the National Academy of Sciences (USA).

Schedule of Events

10:30 a.m. **Registration**
Cleary Alumni & Friends Center | UW-L Campus

11 a.m. **Symposium**

Do Parallel Algorithms and Programs Need to be Parameter Aware?

For more than half a century programmers have enjoyed efficiently universal sequential languages and algorithms, which permitted them to write programs independent of machines. We suggest that for parallel or multicore computers this will no longer be generally possible, and that the algorithms that run on them will need to be designed to be aware of the resource parameters of the particular machines on which they are to run. Parameter-free parallel languages or models, and algorithms that are oblivious to some parameters for some reason, will continue to have roles in restricted domains. We suggest, however, that they will always remain inadequate for the full range of tasks that need to be computed.

With parameter-awareness the main promise is that of portable algorithms, those that contain efficient designs for all reasonable ranges of the basic resource parameters and input sizes. Such portable algorithms need to be designed just once, but, once designed, they can be compiled to run efficiently on any machine. In this way the considerable intellectual effort that goes into parallel program design becomes reusable.

To permit such portable algorithms some standard bridging model is needed - a common understanding between hardware and algorithm designers of what the costs of a computation are. We shall suggest the Multi-BSP model as a candidate for this bridging role. We show that for several basic problems, namely matrix multiplication, fast Fourier Transform, and sorting, portable algorithms do exist that are optimal in a defined sense, for all combinations of input size and parameter values for this model.

4:30 p.m. **Registration**
Cleary Alumni & Friends Center
UW-L Campus

5 p.m. **Keynote**

Biological Evolution as a Form of Learning

Living organisms function according to protein circuits. Darwin's theory of evolution suggests that these circuits have evolved through variation guided by natural selection. However, the question of which circuits can so evolve in realistic population sizes and within realistic numbers of generations has remained essentially unaddressed.

We suggest that computational learning theory offers the framework for investigating this question, of how circuits can come into being adaptively from experience, without a designer. We formulate evolution as a form of learning from examples. The targets of the learning process are the functions of highest fitness. The examples are the experiences. The learning process is constrained so that the feedback from the experiences is Darwinian. We formulate a notion of evolvability that distinguishes function classes that are evolvable with polynomially bounded resources from those that are not. The dilemma is that if the function class, say for the expression levels of proteins in terms of each other, is too restrictive, then it will not support biology, while if it is too expressive then no evolution algorithm will exist to navigate it. We shall review current work in this area.

For further information about the lecture contact:

Steve Senger, Ph.D., Computer Science Department
University of Wisconsin-La Crosse | 1725 State St., La Crosse, WI 54601
608.785.6805 | email: *compsci@uwlax.edu*

Co-sponsored by the
University of Wisconsin-La Crosse Foundation Inc.
Department of Computer Science • College of Science and Health