HEIDELBERG LAUREATE FORUM

Abstracts of Laureates' Presentations 2013 (alphabetic)

Michael Atiyah: Advice to a Young Mathematician

Having spent a long life in mathematics I thought it would be useful to give some general advice to the younger generation based on my experience.

I will discuss many aspects of a life in research, both personal and scientific. Motivation, psychology, curiosity will be some of the topics raised. I will also discuss style, the role of proof and the importance of grounding oneself with interesting examples.

Edmund Melson Clarke: Model Checking and the Curse of Dimensionality

Model Checking is an automatic verification technique for large state transition systems. It was originally developed for reasoning about finite-state concurrent systems. The technique has been used successfully to debug complex computer hardware and communication protocols. Now, it is beginning to be used for software verification as well. The major disadvantage of the technique is a phenomenon called the State Explosion Problem. This problem is impossible to avoid in worst case. However, by using sophisticated data structures and clever search algorithms, it is now possible to verify state transition systems with astronomical numbers of states.

John Hopcroft: Future Directions in Computer Science Research

Over the last 40 years, computer science research was focused on making computers useful. Areas included programming languages, compilers, operating systems, data structures and algorithms. These are still important topics but with the merging of computing and communication, the emergence of social networks, and the large amount of information in digital form, focus is shifting to applications such as the structure of networks and extracting information from large data sets. This talk will give a brief vision of the future and then an introduction to the science base that needs to be formed to support these new directions.

HEIDELBERG LAUREATE FORUM

William M. Kahan: Desperately Needed Remedies for the Undebuggability of Large-Scale Floating-Point Computations in Science and Engineering

How long does it take to either allay or confirm suspicions, should they arise, about the accuracy of a computed result? Often diagnosis has been overtaken by the end of a computing platform's service life. Diagnosis could be sped up by at least an order of magnitude if more users and developers of numerical software knew enough to demand the needed software tools. Almost all these have existed though not all together in one place at one time. These tools cope with vulnerabilities peculiar to Floating-Point, namely roundoff and arithmetic exceptions. Programming languages tend to turn exceptions into branches which are prone to error. In particular, unanticipated events deemed ERRORs are handled in obsolete ways inherited from the era of batch computing. There are better ways. They would have prevented the crash of Air France #447 in June 2009, among other things.

This long and evolving document has been posted at <www.eecs.berkeley.edu/~wkahan/Boulder.pdf>.

Richard Manning Karp: The Computational Lens on the Sciences

The computational lens is a metaphor for a relationship that is emerging between the theory of computation and the physical, biological, engineering and social sciences. Viewing natural or engineered systems in therms of their computational requirements or capabilities provides new insights and ways of thinking. We discuss the influence of this view on quantum computing, statistical physics, biology, mathematics and economic and social processes supported by the Worldwide Web.

Alan Kay: Putting Turing to Work

Many computer programs today are enormously large -- 10s and 100s of millions of lines of code -- and distressingly messy and buggy -- many to the point that programmers are afraid to make major revisions, but must resort to adding still more code "around the outside". Though it is hard to prove interesting and useful things about computations, analogies to some of the expressive ideas in Mathematics can be used to make programs much more expressive and much smaller. Similarly, though the scales are not commensurate to really use some of the deep ideas from Biology, many of the architectural principles that produce life from inanimate matter can be adapted to make large systems which "find ways to work and keep working".

HEIDELBERG LAUREATE FORUM

Butler Lampson: Hints and Principles for Computer System Design

I have many hints that are often helpful in designing computer systems, and I also know a few principles. There are several ways to organize them:

- Goals (What you want)—simple, timely, efficient, adaptable, dependable, yummy.
- Methods (How to get it)—approximate, increment, iterate, indirect, divide and conquer.
- Phases (When to apply them)—requirements, architecture, process, techniques.

Of course the goals are in conflict, and engineering is the art of making tradeoffs, for instance among features, speed, cost, dependability, and time to market. Some simpler oppositions are:

- For adaptable, between evolving and fixed, long-lived and one-shot, monolithic and extensible, scalable and bounded.
- For dependable, between deterministic and non-deterministic, volatile and persistent, precise and sloppy, reliable and flaky, consistent and eventual.
- For incremental, between indirect and inline, dynamic and static, experiment and plan, discover and prove.

It also helps to choose the right coordinate system, just as center of mass coordinates make many dynamics problems easier. You can view the system state as a name→value map, or as an initial state and a sequence of operations that transform the state. You can view a function as code or as a table or as a sequence of partial functions. Notation, vocabulary, and syntax are other kinds of coordinates.

In the complex process of designing systems, both principles and hints can only be justified by examples of what has worked and what has not.

Curtis T. McMullen: Billiards and Moduli Spaces

The motion of a billiard ball on a rectangular table is related to complex analysis on a torus. Billiards in more general polygons can be studied using surfaces of higher genus.

We will present this connection as a gateway to current research on Riemann surfaces and their moduli spaces.

HEIDELBERG LAUREATE FORUM

Silvio Micali: Rational Proofs

We unify the treatment of asymmetry of information in theoretical computer science and economics.

We put forward a new type of proof system, where an unbounded Prover and a bounded Verifier interact, on inputs a string x and a function f, so that the Verifier may learn f(x). In our setting Provers are not “honest” or “malicious”, but RATIONAL, that is, trying to maximize their utility. In essence, (1) the Verifier gives the Prover a reward in [0,1] determined by the transcript of their interaction; (2) the Prover wishes to maximize his expected reward; and (3) the reward is maximized only if the verifier correctly learns f(x).

Rational proofs are as powerful as interactive ones, but can be amazingly more efficient in the amount of communication involved: that is, the number of bits exchanged and the number of rounds of interaction.

Joint work with Pablo Azar.

Michael O. Rabin: Miracles of Cryptography, Preventing Collusion in Auctions

We present novel algorithms enabling an auctioneer in a sealed bid auction to prove to bidders who won without revealing any bid values. These methods were extended together with Silvio Micali to enable bidders to submit bids in a deniable uncontrollable manner. One application is solution to the important open problem of prevention of collusion in auctions.

Raj Reddy: Who invented the Computer: Babbage, Atanasoff, Zuse, Turing or von Neumann?

In this talk we examine the evolution of automation of computation and the events that are central to the advancement of computers. We will examine the concepts and technologies that were seminal, the necessary conditions, to the development of computers and the people that were principal players in the emergence of the modern computers.

Joseph Sifakis: System Design Science

Design is the process that leads to artifacts meeting given requirements. We propose a decomposition of a design process into three main steps each one addressing a corresponding characteristic problem. We identify principles and associated scientific challenges for moving design from an art to a science.

Endowing design with scientific foundations would meet an urgent demand for cost-effectively building complex, trustworthy artifacts. Failure in this endeavor, would seriously limit our capability to master the techno-structure and its further development intended to address urgent global challenges for optimal resource management and enhanced services.

The envisioned goal is both intellectually challenging and culturally enlightening. It is at least of equal importance as the quest for scientific discovery in natural sciences.

Stephen Smale: Protein Folding

The National Research Council Report on Mathematics to 2025 has drawn attention to 5 areas in science for mathematicians; Protein Folding is one.

Here I will give some mathematical setting of this problem addressed to scientists without background in biology. We will lead up to our new results on this topic. A main idea is to introduce a geometry on the space of proteins.

Madhu Sudan: Reliable Meaningful Communication

Around 1940, engineers working on communication systems encountered a new challenge: How can one preserve the integrity of digital data, where minor errors in transmission can have catastrophic effects? The resulting theories of information (Shannon 1948) and error-correcting codes (Hamming 1950) created a ''marriage made in heaven'' between mathematics and its applications. On the one hand emerged a profound theory that could measure information and preserve it under a variety of adversarial injections of errors; and on the other hand the practical consequences propelled telephony, satellite communication, digital hardware and the internet. Today, as we allow computers and computational devices to interact freely with each other and control complex engineering systems, all with limited human intervention, we encounter a new challenge: How can we ensure that computers interpret the messages they receive from each other correctly so that they do not cause catastrophic actions due to misinterpretation? The resulting class of questions poses new challenges to mathematics: both in modelling, and in analysis. In this talk I will give a brief survey of the history of reliable communication, and outline the challenges of communicating meaningfully.

Leslie Valiant: Learning as the Source of Life's Phenomena

The assertion that all aspects of living organisms are determined by some combination of learning during an individual life and evolution beforehand is close to tautologous. Recent work that unifies within a computational framework Darwinian evolution on the one hand with learning on the other, can be viewed as a unified approach to how life is determined. We shall present this viewpoint and summarize some of its broader implications.

Srinivasa Varadhan: Scaling Limits

We often model the evolution of a complex system at the lowest or micro level because that is what makes the most physical sense. But if the system is large, we invariably want answers to questions posed on the larger macro or global scale. This then involves the study of equations with a large number of variables and extracting useful information from their solutions. The macroscopic quantities of interest may not contain complete information about the microscopic variables that drive their evolution. Something has to be done to obtain a closed system of equations for quantities of interest. Different contexts require different approaches. We will look at several examples where rigorous results have been obtained. The process involves some averaging and the presence of some randomness in the evolution helps.

Vladimir Voevodsky: Univalent Foundations of Mathematics

Set-theoretic approach to foundations of mathematics work well until one starts to think about categories since categories cannot be properly considered as sets with structures due to the required invariance of categorical constructions with respect to equivalences rather than isomorphisms of categories. On the other hand the underlying groupoids of categories are invariant (up to an equivalence) with respect to equivalences of categories. Hence, it is natural to think of categories as groupoids with structures. The same applies to higher categories. By Grothendieck's insight higher groupoids correspond to homotopy types which suggests that mathematics of all levels may be thought of as study of structures on homotopy types. Univalent foundations use the language of Martin-Lof type theories to directly work with structures on homotopy types without reducing them to sets.

HEIDELBERG LAUREATE FORUM

Avi Wigderson: Randomness

Is the universe inherently deterministic or probabilistic? Perhaps more importantly - can we tell the difference between the two?

Humanity has pondered the meaning and utility of randomness for millennia.
There is a remarkable variety of ways in which we utilize perfect coin tosses to our advantage: in statistics, cryptography, game theory, algorithms, gambling... Indeed, randomness seems indispensable! Which of these applications survive if the universe had no randomness in it at all? Which of them survive if only poor quality randomness is available, e.g. that arises from "unpredictable" phenomena like the weather or the stock market?

A computational theory of randomness, developed in the past three decades, reveals (perhaps counter-intuitively) that very little is lost in such deterministic or weakly random worlds - indeed, most application areas above survive!
In the talk I'll explain the main ideas and results of this theory. A key notion is pseudorandomness, whose understanding impacts large areas in mathematics and computer science.