

Abstracts Lectures

(sorted by date and time)

MONDAY, AUGUST 24

Stefan Hell

“Optical Microscopy: the Resolution Revolution”

(Monday, August 24, 9:00 a.m.)

Throughout the 20th century it was widely accepted that a light microscope relying on conventional optical lenses cannot discern details that are much finer than about half the wavelength of light (200-400 nm), due to diffraction. However, in the 1990s, the viability to overcome the diffraction barrier was realized and microscopy concepts defined, that can resolve fluorescent features down to molecular dimensions. In this lecture, I will discuss the simple yet powerful principles that allow neutralizing the limiting role of diffraction^{1,2}. In a nutshell, feature molecules residing closer than the diffraction barrier are transferred to different (quantum) states, usually a bright fluorescent state and a dark state, so that they become discernible for a brief period of detection. Thus, the resolution-limiting role of diffraction is overcome, and the interior of transparent samples, such as living cells and tissues, can be imaged at the nanoscale.

Manuel Blum

“Human Computation”

(Monday, August 24, 9:45 a.m.)

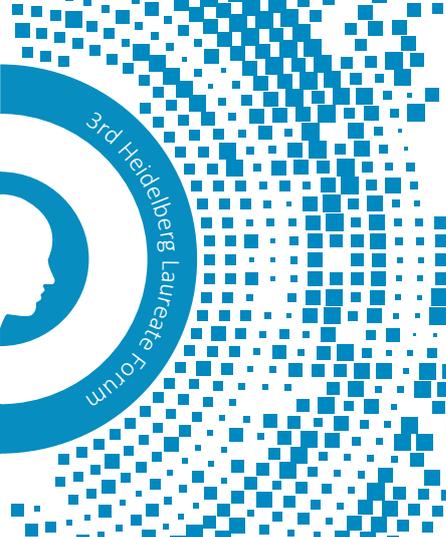
A source of power for theoretical computer science has been the interplay of algorithms and complexity: given a computational problem to be solved, algorithm design is concerned with finding faster solutions than what is currently available. Complexity theory is concerned with showing that faster solutions are impossible.

When a faster algorithm appears impossible to get, one can sometimes use that intuition to raise the complexity lower bound.

When the complexity lower bound appears impossible to raise, one can sometimes use that intuition to construct a faster algorithm.

This is the well known point and counterpoint of modern theoretical computer science.

The goal of human computability is to do the same thing - to build the same kind of theory - to get the same kind of algorithmic upper bounds and complexity theoretic lower bounds for the kinds of computations that humans can do in their heads.



Abstracts Lectures

Leslie Lamport

“A Mathematical View of Computer Systems”

(Monday, August 24, 11:30 a.m.)

Mathematics provides what I believe to be the simplest and most powerful way to describe computer systems.

Shigefumi Mori

“A compelling Desire to do Mathematics”

(Monday, August 24, 12:00 noon)

Mathematics is admittedly useful in the current digital world, though it is classified as a curiosity-driven science, which I accept. There is thus a question frequently asked by non-mathematicians.

If mathematicians do their research out of curiosity why should they be supported by tax money?

It is not easy to give a direct answer that everyone accepts.

There are various levels of curiosities depending on the scene; it can be as general as one about the principle of universe, as special as one about an actual mathematical problem, or as practical as one about an application, but it has to be enchanting for the mathematician.

How is the outcome evaluated if it is done out of curiosity? There are notions of rigor and beauty that mathematicians generally accept; while the former is objective and sometimes painful, the latter is subjective and often addictive. I believe that each of us evaluate the work by its beauty even before it is completed, and a good sense of beauty will lead him/her to a final solution. I will show examples including some of my own experiences in my career.

Along this line of arguments, I will give my answer to the first question in my talk.

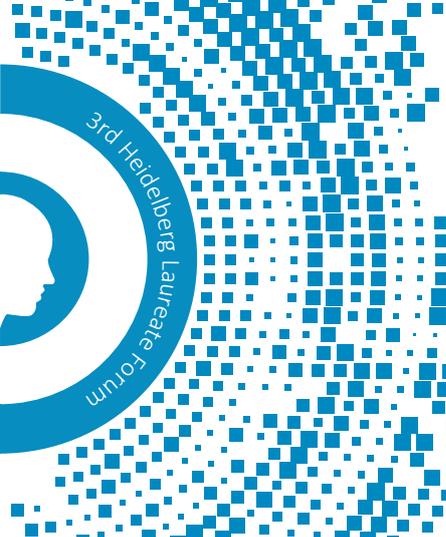
Lenore Blum

“Alan Turing and the other Theory of Computation”

(Monday, August 24, 12:30 p.m.)

Most theoretical computer scientists are familiar with Alan Turing’s 1936 seminal paper setting the stage for the foundational (discrete) theory of computation. Most however remain unaware of Turing’s 1948 seminal paper, notwithstanding its many references in the modern literature of numerical analysis and computational mathematics. The 1948 paper, introducing the notion of condition, sets the stage for a natural theory of complexity for the “other theory of computation” emanating from the classical tradition of equation solving and the continuous mathematics of calculus.

This talk will recognize Alan Turing’s work in the foundations of numerical computation (in particular, his 1948 paper “Rounding-Off Errors in Matrix Processes”), its influence in complexity theory today, and how it provides a unifying concept for the two major traditions of the Theory of Computation.



Abstracts Lectures

TUESDAY, AUGUST 25

Sir C. Antony R. Hoare

"Pioneers of Computer Science: Aristotle and Euclid"

(Tuesday, August 25, 9:00 a.m.)

In the fifteen years of this century, computer application has expanded dramatically from day to day, both in outreach and in sophistication. However, the origins of computer science itself date back several millennia, to the teachings of the most famous philosophers of the ancient world. I refer to Aristotle, the recognised founding father of biology and of logic, and to Euclid, whose Elements of geometry have been taught in schools right up to modern times. Throughout this interval, the ideas and culture of computer science have been advanced by philosophers, logicians and mathematicians. They contribute substantially to the intellectual heritage of the human race.

Edmund Melson Clarke

"Model Checking Hybrid Systems"

(Tuesday, August 25, 9:45 a.m.)

Although every undergraduate in computer science learns about Turing Machines, it is not well known that they were originally proposed as a means of characterizing computable real numbers. For a long time, formal verification paid little attention to computational applications that involve the manipulation of continuous quantities, even though such applications are ubiquitous. In recent years, however, there has been great interest in safety-critical hybrid systems involving both discrete and continuous behaviors, including autonomous automotive and aerospace applications, medical devices of various sorts, control programs for electric power plants, etc. As a result, the formal analysis of numerical computation can no longer be ignored. In this talk, we focus on one of the most successful verification techniques, bounded model checking. Current industrial model checkers do not scale to handle realistic hybrid systems. We believe that the key to handling more complex systems is to make better use of the theory of the computable reals and computable analysis. We argue that new formal methods for hybrid systems should combine existing discrete methods in model checking with new algorithms based on computable analysis. In particular, we discuss a model checker that we are currently developing along these lines.

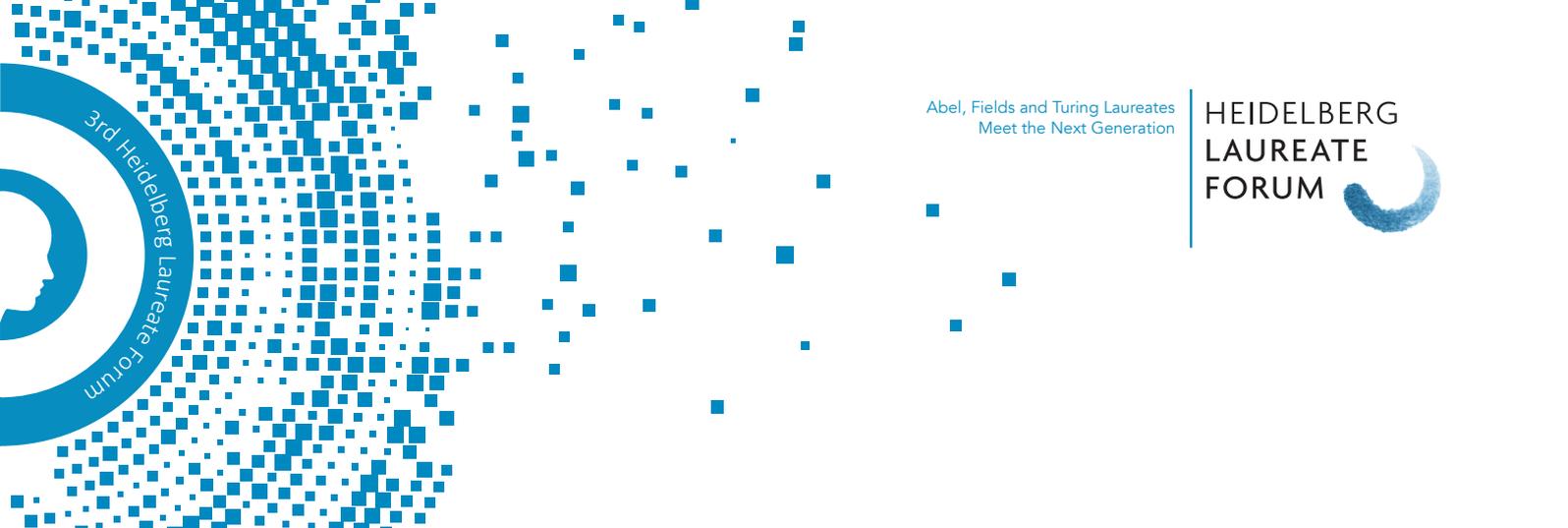
Vladimir Voevodsky

"UniMath"

(Tuesday, August 25, 11:30 a.m.)

UniMath is the name of the library of formalized mathematics and related formalization tools that Benedikt Ahrens, Daniel Grayson, Michael Warren and myself started to build a little over a year ago based in the earlier code created in 2009-2014.

The library is written entirely in the univalent formalization style and it is implemented using a very small subset of the Coq proof assistant. I will tell about some of the features of this library and about some of the projects that use this library as a starting point.



Abstracts Lectures

Fred Brooks

“A Personal History of Computers”

(Tuesday, August 25, 12:15 p.m.)

I fell in love with computers at age 13, in 1944 when Aiken (architect) and IBM (engineers) unveiled the Harvard Mark I, the first American automatic computer. A half-generation behind the pioneers, I have known many of them. So this abbreviated history is personal in two senses: it is primarily about the people rather than the technology, and it disproportionately emphasizes the parts I know personally.

THURSDAY, AUGUST 27

Ivan Sutherland

“Understanding self-timed systems”

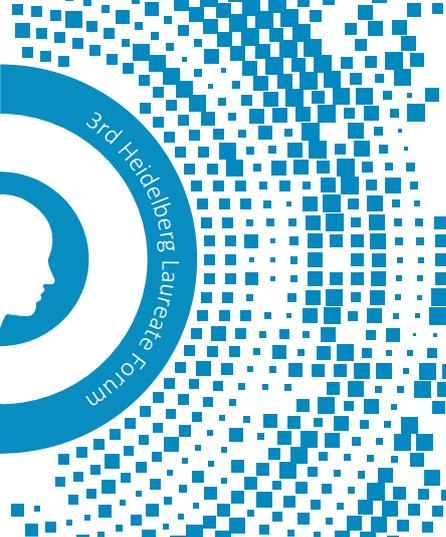
(Thursday, August 27, 9:00 a.m.)

The “clocked” paradigm for hardware design insists that the state of each and every part of a system may change only upon the “tick” of a clock. Each clock tick changes the global system state atomically; between clock ticks the global state remains stable. Global state is a convenient mental model for complex digital systems. However, each new circuit technology makes logic faster and communication relatively slower, making atomic changes to a global state harder to approximate. The clocked paradigm focuses attention on global state but lacks reality by ignoring space.

A few research teams, including our own, explore a “self-timed” alternative design paradigm. The self-timed paradigm gains modularity by allowing each task to start as soon as it gets data and to take as much or as little time as it needs for each particular data set. Global state is undefined because concurrent tasks separated in space may change local state at their own convenience. A self-timed system in operation can be as disorderly as a kindergarten playground during recess. How are we to understand, design and test self-timed systems?

Our group distinguishes two kinds of self-timed components that we call Links and Joints. Links form the edges of a directed graph and joints form its nodes. Links don’t compute; they only store and transport data values between joints. Joints don’t store data; they only use data values from their input links to compute new values for their output links. All known circuit families for self-timed systems fit our Link – Joint model.

This talk reports a fully-testable, working, chip experiment to show how our Link – Joint model for self-timed systems simplifies understanding, design, and test. Our Link – Joint model replaces evolving global state by a mental model of data elements moving through a sequence of steps distributed in space. Our Link – Joint model focuses attention on data transport, the most costly part of modern computing equipment.



Abstracts Lectures

Butler W. Lampson

“Hints and Principles for Computer System Design”
(Thursday, August 27, 9:45 a.m.)

I have many hints that can be helpful in designing computer systems, as well as a few principles. Two ways to organize them are:

- Goals (what you want)—simple, timely, efficient, adaptable, dependable, yummy.
- Methods (how to get it)—approximate, increment, iterate, indirect, divide and conquer.

Of course the goals are in conflict, and engineering is the art of making tradeoffs. It also helps to choose the right coordinate system, just as center of mass coordinates make many dynamics problems easier. For example, 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. 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.

Leonard Max Adleman

“Strata in Complex Analysis”
(Thursday, August 27, 11:30 a.m.)

What would happen if a bunch of computer scientists who knew (and perhaps still know) next to nothing about complex analysis studied it for a decade? This is not a talk about algorithms or complexity theory; it is pure math, and this is the first presentation of our results. We have looked at Riemann surfaces and cut them into flat sheets. This is nothing new, but perhaps how we have cut them and what we have done with the resulting sheets is. In particular, we have proven that the sheets, which we call strata, form a field analogous to the field of algebraic numbers and that they obey a version of the famous Kronecker-Weber theorem that all Abelian extensions of the rationals are subfields of the cyclotomic fields. Further, we have generalized the notion of power series, and used it to prove an extension of Cauchy’s famous residue theorem from integrating around poles to integrating around branch-points. We hope to use our approach to revisit Kronecker’s Jugendtraum (Hilbert’s 12th problem).

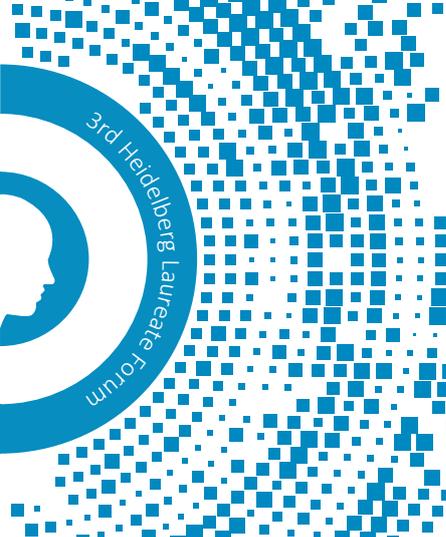
Endre Szemerédi

“The Shifting Method”
(Thursday, August 27, 12:15 p.m.)

The shifting method is a counting technique. To understand it we are going to illustrate the method with three examples.

- 1) Arithmetic progressions of length 3.
- 2) Ramsey--type problem in additive combinatorics.
- 3) Giving a tight bound for the size of a set $S \subset [1, N]$ having the property that for every $x, y \in S, x + y \neq Z^2$.

Part of the work is a joint work with S. Herdade and A. Khalfalah.
No prior knowledge is assumed.



Abstracts Lectures

FRIDAY, AUGUST 28

Peter Naur

“Science versus Philosophy”

(Friday, August 28, 9:00 a.m.)

Philosophy may be very harmful to science by making claims that have no empirical support, no support in the constitution of the world as it may be observed.

A prominent example of philosophical confusions concerns human knowing. Immanuel Kant in his philosophical treatise Critique of pure reason from 1781, in its very first paragraph of the Introduction says that *...the faculty of knowledge should be awakened ... by means of... objects which affect our senses, and partly of themselves produce representations ...*

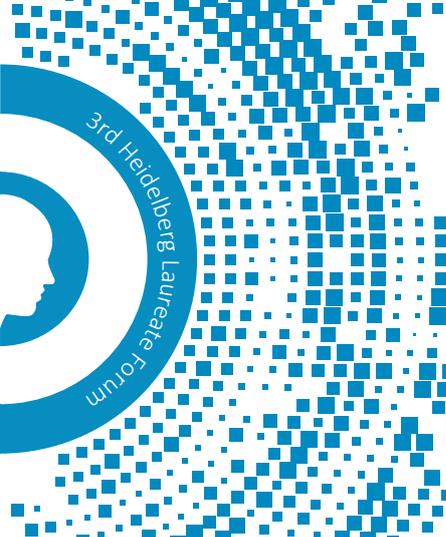
This talk of knowledge producing representations has been repeated in psychology ever since Kant’s time. Thus nowadays the representations of knowledge are supposed to be held in what is called memory. And so in Encyclopedia of the Neurological Sciences (2003) we find articles on Memory, Autobiographical by E. Svoboda and B. Levine, Memory, Episodic by J. H. Kramer, Memory, Explicit/Implicit by B. J. Cherry, Memory, Overview by J. H. Kramer, Memory, Semantic by E. K. Warrington, Memory, Spatial by L. F. Jacobs, Memory, Working by J. A. Waltz. However, what is said in these many articles is merely confusion. There is no clarity in what is supposedly held in what is called memory and how this relates to the experience of the person.

Meanwhile, already in 1890, William James in his Principles of Psychology stated a very different understanding of knowledge. James explains knowledge as a relation between minds and other things in the world. In this there is nothing that might be held in a memory container, no representations of knowledge.

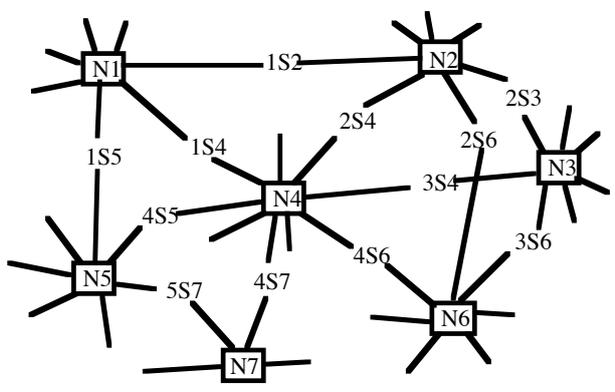
James further writes:

There are two kinds of knowledge broadly and practically distinguishable: we may call them respectively knowledge of acquaintance and knowledge-about. Most languages express the distinction; thus gnvnai, eidenai; noscere, scire; kennen, wissen; connaître, savoir. I am acquainted with many people and things, which I know very little about, except their presence in the places where I have met them. I know the color blue when I see it, and the flavor of a pear when I taste it; I know an inch when I move my finger through it; a second of time, when I feel it pass; an effort of attention when I notice it; but about the inner nature of these facts or what makes them what they are, I can say nothing at all.

This understanding of how knowing comes in two kinds is in perfect accord with the theory of how thinking happens in the neural system, in the brain, that I have developed since 2003 under the designation the synapse-state theory. According to this theory, the central part of the neural system in the brain, what I call the item layer, consists of a neural net in which a large number of nodes are connected pairwise through neurons and synapses. The figure shows a very small part of the neural net.



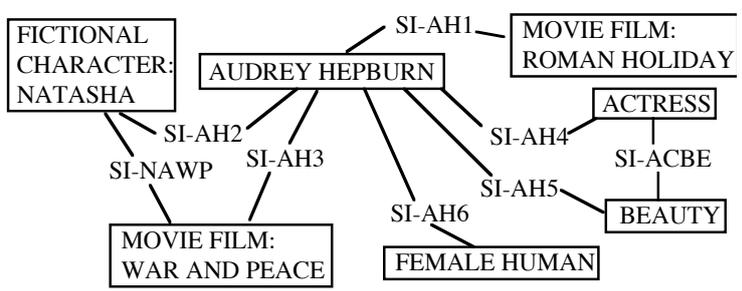
Abstracts Lectures



Nodes, N, and synapses, S, in small part of the item layer.

Each synapse at any time is in a particular state of conductivity to the neural excitations that are distributed throughout the net.

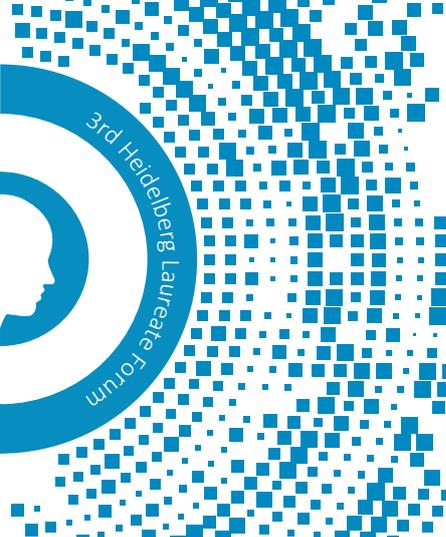
The excitation of the nodes of the item layer originates in the sense layer and in attention synapses, not shown in the figure. The excitation of each node at any time is formed as the sum of the excitations that it receives from the connected synapses. The overall neural activity at any moment is a pattern of excitations of the nodes in which a few nodes are excited strongly while a large number are excited weakly. This pattern of excitations is what is experienced by the person as the momentary stream of thought, of consciousness.



Subject aggregate: AUDREY HEPBURN

The neural net is where knowing by acquaintance and knowing about is held. Each thing known by acquaintance is held in one node. The figure shows the part of the net where the person's acquaintance with AUDREY HEPBURN, FICTIONAL CHARACTER: NATASHA, MOVIE FILM: WAR AND PEACE, MOVIE FILM: ROMAN HOLIDAY, ACTRESS, BEAUTY, and FEMALE HUMAN are held.

Knowing about is held in the net in the form of the conductive state of the synapses connecting any pair of nodes. For example, knowing about Audrey Hepburn that she is an actress is held as the conductive state of the synapse SI-AH4. The validity of this picture of the nervous system of the brain has been confirmed experimentally in 2005 by scientists at Caltech and UCLA. They found a single brain cell that became strongly activated whenever the person was shown a picture of Jenifer Aniston. This was reported in National Geographic Danmark 2014 No. 2.



Abstracts Lectures

Andrew C. Yao

"Quantum Computing: A Great Science in the Making"

(Friday, August 28, 9:45 a.m.)

In recent years, the scientific world has seen much excitement over the development of quantum computing, and the ever increasing possibility of building real quantum computers. What's the advantage of quantum computing? What are the secrets in the atoms that could potentially unleash such enormous power, to be used for computing and information processing? In this talk, we will take a look at quantum computing, and make the case that we are witnessing a great science in the making.

Srinivasa S. R. Varadhan

"Entropy and its many Avatars"

(Friday, August 28, 11:30 a.m.)

The concept of Entropy was first introduced by Clausius in his study of the relationship between mechanical energy and heat. It was given a mathematical definition by Boltzmann in his formulation of statistical mechanics. It has since appeared as an indispensable tool in dynamical systems, communication theory, probability theory and statistics. We will explore these concepts and their interrelationships.

Leslie G. Valiant

"Holographic Algorithms"

(Friday, August 28, 12:15 p.m.)

When are two mathematical functions the same? One might think that this can be generally answered immediately from their definitions. However, functions may have numerous dissimilar alternative definitions. Fortunately, sameness can be often demonstrated systematically by certain linear mappings internal to the function definitions. These mappings, called holographic transformations, offer a powerful tool for showing that a function class is equivalent to one known to be efficiently computable, or, alternatively, that it is equivalent to one known to be in a completeness class suspected of being computationally intractable. We shall survey these ideas and their applications in computational complexity.