

Computational complexity

In 1965, the year Juris Hartmanis became Chair of the new CS Department at Cornell, he and his colleague Richard Stearns published the paper *On the computational complexity of algorithms* in the *Transactions of the American Mathematical Society*. The paper introduced a new field and gave it its name. Immediately recognized as a fundamental advance, it attracted the best talent to the field. Theoretical computer science was immediately broadened from automata theory, formal languages, and algorithms to include computational complexity.

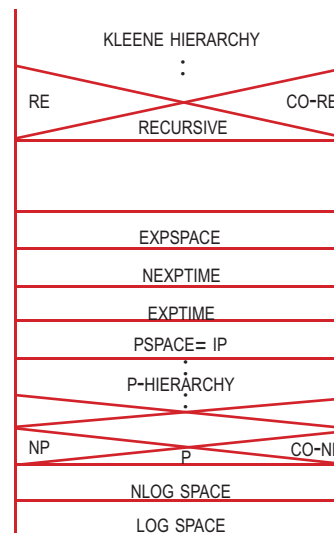
As Richard Karp said in his Turing Award lecture, “All of us who read their paper could not fail to realize that we now had a satisfactory formal framework for pursuing the questions that Edmonds had raised earlier in an intuitive fashion — questions about whether, for instance, the traveling salesman problem is solvable in polynomial time.”

Hartmanis and Stearns showed that computational problems have an inherent complexity, which can be quantified in terms of the number of steps needed on a simple model of a computer, the multi-tape Turing machine. In a subsequent paper with Philip Lewis, they proved analogous results for the number of tape cells used. They showed that, given sufficiently more time, Turing machines can always compute more functions. This theorem revealed the existence of a rich hierarchy of complexity classes and provided a framework on which modern complexity theory is built.

Hartmanis and Stearns had the insight to consider both deterministic and nondeterministic models of computation. Their exploration of this relationship laid the groundwork for the celebrated P vs. NP question: whether the class of problems solvable by nondeterministic Turing machines running in polynomial time is strictly larger than the class of problems solvable by their deterministic counterparts. The subsequent work of Stephen

Cook, Leonid Levin, and Karp revealed how this question lies at the heart of computationally hard problems throughout computer science and other fields. Nondeterministic Turing machine computation has turned out to be very effective at exposing the subtle interactions between the notions of computing a function and simply verifying its result. Due to its fundamental nature, the P vs. NP question is today widely viewed as one of the most important open questions in all of mathematics.

Studying the structure of complexity classes defined by bounding the time or space allowed for computation has led to a surprisingly rich theory, with striking



This diagram shows how the field believes complexity classes look. It is known that P is different from ExpTime, but there is no proof that $NP \neq P$ and $PSPACE \neq P$.

equivalences and separations among complexity classes, fundamental and hard open problems, and unexpected connections to distant fields of study. An early example of the surprises that lurk in the structure of complexity classes is the Gap Theorem, proved by Hartmanis’s student Allan Borodin and by Boris Trakhtenbrot; essentially, it says that surprisingly large “gaps” appear in the hierarchy. Sometimes, the surprises come from the equivalence of two complexity classes that had been long imagined to be different: in 1988, Hartmanis’s student Neil Immerman and Róbert Szelepcsényi showed that nondeterministic space is closed under complementation, resolving a question that had withstood 25 years of research. They were awarded the 1995 Gödel prize for this work.

Many fundamental questions in complexity theory remain open, despite efforts of researchers in the past 40 years. One of the characteristics of Hartmanis has been his ability to come up with questions that are astoundingly hard to answer. For example, he and his student Ted Baker conjectured that all NP complete sets are isomorphic, under isomorphism computable in polynomial time — a conjecture that is still open. If this conjecture is true, then there is essentially only one NP-complete set, which appears in many guises.

The concepts and methods of complexity theory apply widely and unexpectedly in many parts of math and computer science, from conjectures about the complexity of computable real numbers to reformulations of such basic notions as inductively defined sets. For instance, CS professor Dexter Kozen, a former student of Hartmanis, who introduced alternating machines combining existential and universal quantifiers, recently generalized this award winning work to provide a computational characterization of inductively defined sets that captures the hyperelementary relations over arbitrary structures — the runtime of his machines is measured by ordinals. Even more far afield, Bob Constable and Kurt Mehlhorn defined the computational complexity of higher-order functions and complexity classes

Hartmanis (pictured) and colleague Richard Stearns showed that computational problems have an inherent complexity.



of such functions; these concepts have been useful in studying the runtime of programs in functional languages such as ML and Haskell.

In this short article, we have barely touched the surface of a field that is mathematically deep, rich, and beautiful. Over the years, Hartmanis and his students have been in the thick of the research in this field and, together with other Cornell faculty, have helped it reach out to influence other disciplines. For example, the connections between phase transitions

and computational complexity, discussed below in the “ice cube” highlight, is drawing physicists into the study of computational complexity.

In 1993, Hartmanis and Stearns were awarded the ACM Alan M. Turing Award, the highest prize given in computer science, “In recognition of their seminal paper, which established the foundations for the field of computational complexity theory.” That paper was the start of 40 years of research by some of the brightest and most curious minds in computer science.



Bart Selman uses SAT to study phase transitions.

ticians, studying random combinatorial structures; and computer scientists, interested in complexity and algorithms. The collaboration has led to novel algorithms and to a detailed understanding of phase transition phenomena in computational problems and the underlying combinatorial search spaces.

CS professor Bart Selman, with physics colleagues Remi Monasson in France and Riccardo Zecchina in Italy and Scott Kirkpatrick at the IBM T. J. Watson Research Center, has been in the thick of this research, dealing mainly with the SAT (satisfiability) problem. Consider a boolean formula that is the **and** of a set of clauses, each of which is the **or** of variables or their negation. Here is a formula with two clauses: (x **or** y) **and** (**not** x **or not** y). This formula can be satisfied (made true) by making x **true** and y **false**. The SAT problem is to determine whether such a formula is satisfiable.

Consider a collection of formulas, and group them by the ratio r of the number of clauses to the number of variables. For small enough r , most formulas in the group will be satisfiable. This makes sense; each of the clauses in a formula restricts possible satisfying assignments, and the fewer restrictions, the more likelihood of being able to find an assignment. But as r get large, more and more formulas become unsatisfiable. A *phase transition* takes place at the point r where suddenly most formulas become unsatisfiable. For the collection of formulas with three variable clauses (3-SAT), r is about 4.25.

Selman and his colleagues found the following surprising result for mixtures of “random” 2-SAT and 3-SAT. With up to a certain percentage of 3-SAT in the mixture, the formulas can be solved in average polynomial time, and the phase transition is smooth. But with a higher fraction of 3-SAT, search procedures for satisfying assignments scale exponentially at the phase transition, and this transition is abrupt, as when water freezes.

They also discuss the *spin-glass* model as a way of explaining why phase transitions work the way they do —perhaps not only for computational problems but for problems in physics as well. The spin glass, a basic model of a magnetic system, starts with an array of magnetic particles, each oriented either up or down. The orientation of each particle affects the orientation of its neighbors, and a particle can flip from one state to the other with certain probability. Getting the model to settle into a lowest-energy state, in which there is no more flipping, is equivalent to solving a satisfiability problem.

In the past, the flow of information has gone from physics to computing. Computing may now provide insights that deepen the understanding of physics and the physical world.

Computational complexity and the ice cube

Water goes through a phase transition when it freezes. So does a metal when it melts. Some of these transitions are smooth; others exhibit anomalies when the critical point of transition is neared, as in water freezing. In 1982, Cornell physics professor Ken Wilson was awarded the Nobel Prize for his 1971 theory that helped explain how these phase transitions worked.

Now, computer scientists are finding similar phase transitions in computational problems. This connection has stimulated collaboration between statistical physicists, studying disordered systems; mathematicians,



Johannes Gehrke receives a Cornell University Provost's Award for Distinguished Scholarship.

David Gries publishes *Multimedia Introduction to Programming Using Java* (Springer-Verlag), with his son, Paul.

Once again, PhD student Ioannis Vetsikas and his software “whitebear” wins first place in the Trading Agent Competition. From 2001 to 2005, his worst finish is third.

Bobby Kleinberg joins.

2005

John Hopcroft receives the 2005 IEEE Harry Goode Memorial Award for “fundamental contributions to the study of algorithms and their applications in information processing”.

Technology Research News magazine, in its “Top Picks: Technology Research Advances of 2004”, includes work by two CS groups: Jon Aizen, Dan Huttenlocher, Jon Kleinberg, and Tony Novak devised a way to measure users’ reactions to an item description; and Lillian Lee and Regina Barzilay developed software that picks up the topic structure of whole documents to generate more accurate automatic summaries.

Fred Schneider is named chief scientist of TRUST (Team for Research in Ubiquitous Secure Technologies) a new five-university NSF Science and Technology Center.

Student Filip Radlinski receives the Best Student Paper Award at the ACM SIGKDD Conference.

Student Alexandru Niculescu-Mizil receives a Distinguished Student Paper Award at ICML.

Student Thomas Finley receives a Distinguished Student Paper Award at ICML.

Thorsten Joachims receives the Best Paper Award at ICML.

Jon Kleinberg, Jure Leskovec, and Christos Faloutsos receive the Best Research Paper Award at the 11th Conf. on Knowledge Discovery and Data Mining.

Jon Kleinberg and Eva Tardos publish *Algorithm Design*. (Addison-Wesley).

Rafael Vinoly architects begin a feasibility study for a CIS information campus signature building; CS is included as a core unit of CIS.