A Short History of Computational Complexity

Lance Fortnow, Steve Homer

Georgia Kaouri
NTUAthens

Overview

- 1936: Turing machine
- early 60's: birth of computational complexity
- early 70's: NP-completeness, P?=?NP
- 70's: different models of computation
- 80's: finite models (eg. circuits)
- 90's: new models of computation (quantum computers, propositional proof system)

Early History

- Ancient Greek (?!?!?!), Chinese
- 1936: Turing
- 1960: Myhill (linear bounded automata)
- 1962: Yamada (real-time computable functions)

- specific time and space bounded machines, but no general approach to measuring complexity

Birth of CC

- 1965: Hartmanis, Stearns
  - definition of multitape TM, time, space
  - measured time/space as a function of the input
  - first results of form "given more time/space more things computed"
  - $s_1(n)=o(s_2(n))$ there are problems solvable in $s_2(n)$, but not in $s_1(n)$
- 1963: Rabin (two-taped TMs)
- 1966: Hennie, Stearns
  - 2-tape vs. single tape TM: log factor more time
  - Time Hierarchy Th: separation if $t_1(n)\log t_1(n)=o(t_2(n))$

Nondeterminism (Space)

- cannot use straightforward diagonalization
- 1970: Savitch's Theorem (problems solved in nondeterministic s(n) can be solved in deterministic $s^2(n)$)
- 1972: Ibarra (there exist problems computable in nondet space n^a, but not space n^b, $a>b\geq 1$)
- 1988: Immerman, Szelepcsenyi (nondet space is close under complement implying that NSPACE=co-NSPACE)

Nondeterminism (Time)

- 1973: Cook (problems computable in nondet space n^a, but not stime n^b, $a>b\geq 1$)
- 1978: Seiferas, Fischer, Meyer (ntime hierarchy th separation if $t_1(n+1)=o(t_2(n))$)
- 1967:Blum (Speed-up th: for any computable, unbounded $r(n)$ there exists a computable L s.t. for any TM accepting L in $t(n)$ there is another TM accepting L in $r(t(n))$)
  - note: $t(n)$ is not necessarily time constructible
- 1972: Borodin, also Trakhtenbrot 1964 (Gap Th: there is recursive f from nonnegatives to nonnegatives s.t. $TIME(f(n)=TIME(2f^{(n)})))$

P=NP

- 1965: Edmonds (MATCHING $\in$ P)
  - is poly time efficient?
  - informal desciption of nondeterministic poly time

NP-completeness

- captures the combinatorial difficulty of many efficient solution resisting problems
- method for proving that a combinatorial problem is as intractsbleas any NP problem
- TSP, Scheduling, LP: many possible solutions, brute-force search
- no evidence neither that there is no poly time solution nor that are difficult for the same reasons

NP-completeness

- 1956: Godel set the question (proofs in first-order logic)
- 1971: Cook (SAT is NP-complete)
- 1973: Levin (tiling problem is NP-complete)
- 1972: Karp (8 combinatorial problems are NP-complete)
  - introduced techniques

Completeness

- PSPACE: Garey, Johnson 1979
  - hex/checkers games (unbounded finite size)
- EXPTIME: Garey, Johnson 1979
  - small number of complete problems

early 70's

- relationship between complexity classes (mainly LOGSPACE and PSPACE)
- properties of problems in within the principal classes (mainly NP)

- Isomorphism Conjecture
- Polynomial Hierarchy
- Alternation
- Logspace
- Oracles

Isomorphism Conjecture

- 1977, 1978: Berman, Hartmanis
- all NP sets are P isomorphic (via poly time computable and invertible isomorphisms)
- proved that all known NP-complete sets are P isomorphic
- still open today
- what happens if the conjecture holds?

Polynomial Hierarchy

- 1976: Meyer, Stockmeyer
- classes between P and PSPACE
- 0 level: P
- 1^{st} level: NP, coNP
- 2^{nd} level: problems in NP related with NP oracle, etc
- if P=PSPACE PH collapses!!!

Alternation

- 1980: Kozen, Chandra, Stockmeyer
- classify combinatorial problems using an alternating TM
  - TM in which the computational tree has inner nodes ^ or v and the number of alternations in each path is bounded
- alternating log space=P
- alternating PSPACE=EXPTIME

Logspace (L, NL)

- off-line TM
- $L \leq NL \leq P$
- proving that a P-complete problem (eg. circuit value) is in L, implies that L=P

Oracles

- 1975: Baker, Gill, Solovay (there is an oracle reactive for which P=NP and another oracle relative to which P≠NP)

Counting Classes

- how many computational paths lead to acceptance?
- 1979: Valiant (#P: functions computing the number of accepting paths of a NTM)
- GapP: functions computing the difference between the number of accepting and rejecting paths of a NTM
- 1991: Toda's th (hard functions in #P lie above any problem in PH)
- 1994-5: Beigel, Reingold, Spielman (PP(unbounded two-sided error) is closed under union)

Probabilistic Complexity

- 1977: Solovay, Strassen (alg is n prime)
- 1977: Gill (BPP)
- 1977: Adleman, Manders (RP)
- Babai (ZPP)
- 1983: Sipser (BPP is contained in PH)
- Also probabilistic space classes
  - Aleliunas, Karp, Lipton, Lovasz, Rackoff (undirected graph connectivity is in RL)
  - BPL, ZPL

Interactive proof systems

- 1985: Babai (MA, AM)
- 1989: Goldwasser, Micali, Rackoff (IP: unbounded AM)
- 1989: Goldwasser, Sipser (equivalence)
- 1989: Furer, Goldreich, Mansour, Sipser, Zachos (for positive instances the prover can succed with no error)
- 1992: Shamir (IP=PSPACE)

Probabilistic Checkable Proofs

- 1994: Fortnow, Rompel, Sipser (the prover writes an exp long proof that the verifier spot checks in probabilistic time)
- 1996: Feige, Goldwasser, Lovasz, Safra, Szegedy (viewing possible proofs as nodes, the size of a clique cannot be approximated well without unexpected collapses in complexity classes)
- 1998: Arora, Lund, Motwani, Sudan, Szegedy (Arora, Safra 1992) every language in NP has a probabilistic checkable proof, where the verifier uses only log number of random coins and constant number of queries to the proof

Derandomization

- how can we reduce the number of truly random bits to simulate probabilistic algorithms?
- 1984: Blum, Micali (create randomness from cryptographically hard functions)
- 1999: (Hastad, Impagliazzo, Levin, Luby) pseudorandomness from one-way functions

Descriptive Complexity

- measures the computational complexity of a problem in terms of the complexity of the logical language needed to define it
- 1973-4: Jones, Selman, Fagin
- 1982: Immerman, Vardi (problems definable in FO logic with the fix-point operator is the P)

Finite Models

Circuit Complexity

- Circuit Complexity: bounds on the size and depth of circuits
- Boolean circuit (size:#gates, depth:|longest path|)
- A circuit recognises a set of strings on length n if it evaluates to 1.
- infinite set of strings <-> infinite collections of circuits

P?=?NP

- L in P is recognised by a circuit family of polynomial size
- Proving that some NP problem does not have polynomial size circuits => P≠NP
- 1949: Shannon (most Boolean functions require exp size circuits)
- AC^0: L recognised by uniform, constant depth, poly size circuits, unbounded fan-in

Communication Complexity

- models the efficiency and complexity of communication between computers
- bounds on the amount of communication and processors required
- distributed and parallel computations
- performance of VLSI circuits

Proof Complexity

- studies the length of proof in propositional logic and relationship
- NP: short, easily verified membership proof contrary to co-NP
- SAT vs. TAUT
- resolution proof systems: statement D is proved by assuming negation and reach a contadiction

Quantum Computing

- 1982: Richard Feynman
- 1985: David Deutch developed the theoretical computation model based on quantum mechanics, suggested that can compute efficiently problems that can not be computed by traditional computers
- 2 algorithms: Shor (1997) factoring integers, Grover (1996) searching a data base of n elements in $O(\sqrt{n})$ time
- 1997: Bernstein, Vazirani (formal definition of BQP: a language computable efficiently by quantum computers)

Future Directions

- P?=?NP remains the main challenge
  - possible connection with areas of mathematics, eg. algebraic geometry, higher cohomology(???)
  - new techniques to prove lower bounds on circuits, proof systems
  - new characterization of P and NP
  - clever twist on diagonalization
- basic questions in quantum computational complexity
- probabilistic, parallel, quantum complexity: new models of computation
- the other “complexity”: complex systems that occur in society and nature (eg. financial markets, internet, biological systems, the weather, physical systems)
- Big Surprise...

The End…

This was a good 40 years and complexity theory is only getting started.