Lecture 40: History & Summary: Theory of Computation

CSCI 81
Spring, 2012

Kim Bruce

Computability

- Leibniz, Hilbert
  - Formalist philosophy of mathematics
- Gödel, Church, Kleene, Rosser, Turing
- Post (like grammars), Labeled Markov Algorithms (rewriting with goto's), ...

Hilbert's Program

- Response to foundational crisis in math
  - Foundations of math threatened by paradoxes.
    - Set of all sets that do not contain themselves.
  - Reduce math to finite complete set of axioms & prove they are consistent

Goals

- Provide secure foundation for math:
  - Formalization of all math (axioms & rules)
  - Completeness: All true statements provable
  - Consistency: Proof no contradictions can be obtained
    - Use only finitistic reasoning.
  - Conservation: Proofs about "real" objects using ideal objects (infinite sets) can be obtained with ideal objects.
  - Decidability: Algorithm for deciding truth.

Impact of Gödel Incompleteness

- Can't formalize even number theory w/o leaving out some true statements.
- No complete consistent extension of Peano Arithmetic w/ semidecidable set of axioms
- No extension of PA can prove its own consistency
- No algorithm to determine truth of statements in consistent extension of PA (Church/Turing)

Impact

- Very little impact on practice of Math
  - Exception: Paris-Harrington Theorem in combinatorics is True but not provable in PA.
  - Implies consistency of PA
  - Math lives on "dangerous" foundations, but any problems likely easily repairable.
- Independence of Axiom of Choice, Continuum Hypothesis (Gödel plus others) from ZF set theory had more impact.

Undecidability

- Shocking at first
  - Pervasive
- Reduction proofs led to work on Complexity Hierarchy.
- NP-Completeness

NP-Completeness

- Set L is in NP if $w \in L$ can be solved in polynomial time on non-deterministic TM
- L is NP complete iff for all L' in NP, there is a polynomial time function p s.t.
  $w \in L'$ iff $p(w) \in L$, i.e. $L' \leq_{pM} L$
- Thus if there is a polynomial algorithm for L then there is one for L'
  - Example: Satisfiability in Prop. Logic
- Million-dollar problem: Does P = NP?

Finite Automata

- McCulloch and Pitts 1943: A Logical Calculus Immanent in Nervous Activity
  - In neural networks
- Mealy, Moore 1955-56 generalized
- Rabin & Scott (1959): Finite automata and their decision problems
  - Introduced non-deterministic FSM's
  - Students of Church at Princeton
  - Each got Turing award in mid-70's

Applications of FSM's

- State Design Pattern
  - Allows object to change behavior when internal state changes.
  - Instance variable representing state of object can be updated to reflect current state.
  - Messages depending on state are routed through state object which responds appropriately.
    - Vending machine
    - Very useful with GUI -- smart phone apps
- UML and other modeling languages

Applications of FSM's

- Cellular automata communicate with neighbors. (Game of life)
  - Wolfram: A new kind of science
  - Equivalent to TM's.
- Regular expressions handy for searches, built-in to UNIX:
  - Automatically compiled to FSM
- Probabilistic ndfsm - Markov chains

Context-free Languages

- Chomsky 1956: Modeling natural languages
- Backus-Naur Form: Describing Algol 60
- Oettinger 1961, Schuzenberger 1963: PDA's

Applications of CFL's

- Describing syntax of PL's
  - Can build parser (pda) directly from grammar
- Restrict to deterministic CFL's
  - LL(k), LR(k) limit amount of lookahead to k characters to decide which rule to apply.

Context-Sensitive Languages

- Like type 0, but require if $\alpha \rightarrow \beta$ is a rule then $|\beta| \geq |\alpha|$
- Example:
  - $S \rightarrow aSBC$
  - $S \rightarrow aBC$
  - $CB \rightarrow BC$
  - $aB \rightarrow ab$

  $bB \rightarrow bb$

  $bC \rightarrow bc$

  $cC \rightarrow cc$
- Generates $a^n b^n c^n$

Context-Sensitive Languages

- Alternatively, restrict to productions of form
  - $\alpha A \beta \rightarrow \alpha \gamma \beta$
  - Rewrite non-terminals in particular contexts
- Accepted by linear-bounded automata
  - Limit tape to amount of space to write input.
- Interesting from theoretical point of view, but not many applications

Logic of Programs

- Floyd 1967 "Assigning meaning to programs"
  - Flow charts. Turing award 1978 (parsing, semantics, verification)
- Hoare 1969 "An axiomatic basis for computer programming"
  - Hoare triples. Turing award 1980 (def & design programming languages)

Logic of Programs

- Remains hard!
  - Especially unsuccessful if write program first and proof later.
  - Systems (Coq) where write proof and extract program from proof.

Model Checking

- In early 1980's: Clarke and Emerson
- Remarkably successful, esp w/hardware
- Take finite state machine and find which nodes satisfy various assertions.
  - Languages include temporal and fixed-point logics
  - State-space explosion still problem
  - Also useful in concurrency: safety and liveness

Final Exam

- FSM, PDA, TM
- Decidability & Semidecidability
- Logic
- Program verification
- Model checking & temporal logic