INFO 4990: Information Technology Research Methods

Mathematical Methods
Lecture by A. Fekete

Mathematical Approaches to Engineering & IT Research

• The nature of mathematics
• How mathematical work is presented
• Using maths and/or doing maths?
• Levels of abstraction
• Examples

What is mathematics?

• A discipline with a very long history
  – Results accumulate since 500BC
  – Unlike other sciences, earlier work is not superceded by later studies
• Originally focused on studying number and shape
  – Now extended to many aspects of structure
• A characteristic style of reasoning: logical proof

Proof

• Logically valid reasoning
  – Which shows that conclusions are necessarily true
• This could be formalized
  – But rarely is!
• Proofs are checked through a social process
  – The community is sensitive to common mistakes
  – Experts have good insight into what fits with existing knowledge

The starting point

• Axioms
• Previous work
• Definitions

Precision

• Very careful use of language
  – What objects does the result apply to?
  – Which values are chosen, and which are determined?
  – Eg "for every x, there is y" vs "for some x, every y is"
  – Eg finite vs infinite
  – Many terms with special meanings
    • Normal, function, graph, group, component
• Make sure all cases are covered
  – Eg empty set? Non-continuous functions?
• Identify what properties are needed for the proof
Abstraction

- Define abstract structures as anything with operations that obey given properties
  - Eg abstract graph is a set of elements and a set of pairs of those elements
- Try to generalize results by seeing if the proof can be done with fewer properties

Mathematical Approaches to Engineering & IT Research

- The nature of mathematics
- How mathematical work is presented
- Using maths and/or doing maths?
- Levels of abstraction
- Examples

Mathematical Writing

- A long tradition of presenting results
- Intended for clarity and precision
  - Those in the intended audience can follow and check
- Very stylized writing
- Definition, Theorem, Proof

Theorem

- Result that might be used in building more research
  - Typically has hypothesis and conclusion
    - i.e. if ... then ...
  - Must be proved
    - Otherwise it is "conjecture"
- Often named after the prover (or someone else!)
- If the result seems less useful (outside this paper), call it a Lemma
  - Typically has very particular hypotheses, carefully chosen to make the proof work out

Common forms of proof

- Stylized English argument (therefore... therefore...)
- Case analysis
- Proof by contradiction
- Proof by induction

Mathematical Approaches to Engineering & IT Research

- The nature of mathematics
- How mathematical work is presented
- Using maths and/or doing maths?
- Levels of abstraction
- Examples
Mathematical Models

- Often key aspects of a computational or engineering problems can be captured in a mathematical form
  - Eg represent internet connections as a graph
  - Eg represent execution of a computation as a sequence of actions
  - Eg represent effect of earthquake in a building by a set of differential equations
- We can then use mathematics to investigate the model
  - And we try to interpret the conclusions as statements about the original situation
  - But we must remember that the model is not reality
    - Always some aspects are left out of the model

Doing Maths

- Sometimes we just use known results of maths to study a model
- But often, questions about the model have not been explored yet
  - So there is a motivation for doing new maths
    - Define appropriate abstractions
    - Prove theorems about them
- There are large communities of “theoretical computer scientists” who work this way
  - They write papers in the style of maths research

Mathematical Approaches to Engineering & IT Research

- The nature of mathematics
- How mathematical work is presented
- Using maths and/or doing maths?
- Levels of abstraction
- Examples

Generalization and meta-reasoning

- Result about a system
  - Eg here is an algorithm to do something
  - Eg here is a model for understanding performance of the system
- Results about all possible systems
  - Eg No algorithm can have better performance than …
  - Eg Any system can be described in this way
- Results about proof techniques
  - Eg we can find type declarations for any correct program
  - Eg we can automate the finding of proofs

Algorithms

- A large part of theoretical CS is inventing new algorithms
  - Sometimes, in new computational paradigms
    - Eg random, quantum, independent-agent
- It is expected that the inventor will analyse the performance (runtime and/or space required)
- It is expected that the inventor will prove the algorithm correct

Analysis of Algorithms

- Typically, time and space are analyzed as a function of the scale of the inputs
  - Usually, we only care about the asymptotic cost
    - Stated in big-Oh notation
    - Does not deal with constant factors or lower-order terms
- Many papers invent an algorithm and give an analysis
- Sometimes a paper just gives an analysis for a previously-known algorithm
  - Either if wasn’t analysed, or the new analysis gives closer (smaller) bounds
- Top conferences: SODA, also FOCS and STOC
Complexity Theory

• The study of the power of different computing approaches
• Eg what can be expressed with regular expressions
• Eg what can be computed in polynomial time
• Eg what can be computed with access to a polynomial number of random bits
• Top conferences: FOCS and STOC

Specialist communities

• Theory of Databases
• Theory of Programming Languages
• Theory of Distributed Computing
• Computational Geometry
• Theory of Cryptography
• Theory of Quantum Information
• etc

Mathematical Approaches to Engineering & IT Research

• The nature of mathematics
• How mathematical work is presented
• Using maths and/or doing maths?
• Levels of abstraction
• Examples

Two Example Papers

• A. Fekete, “Allocating Isolation levels to Transactions”, Proc ACM PODS’05