Formal Methods (Tools)

Jeannette M. Wing

15-827 Security and Cryptography
29 October 1997

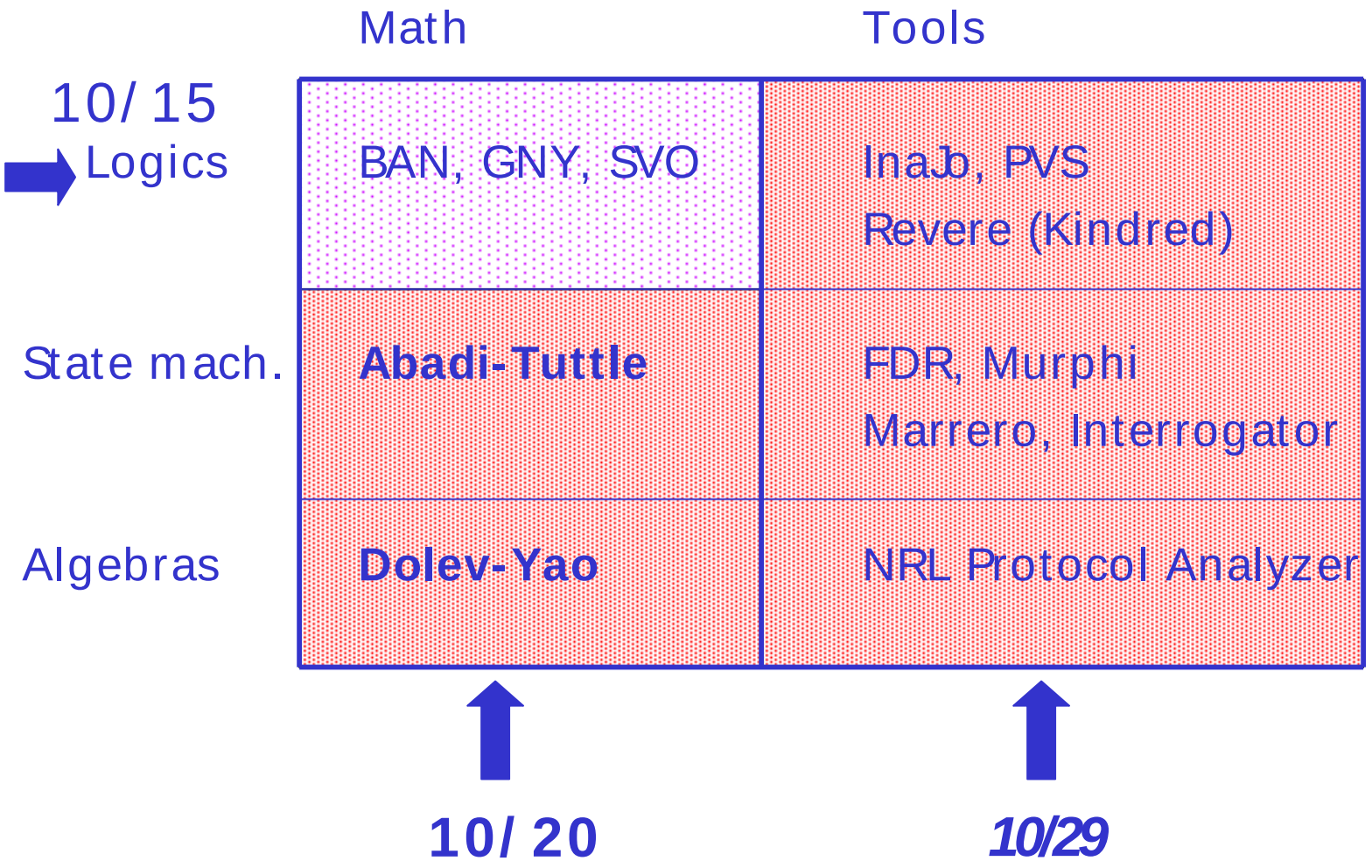
Plan for Today

- Historical perspective
- Current approaches
  - Theorem proving (proof checking, really)
  - Model checking

Goals for Today

- How can you use X to reason about security protocols? What are their strengths and limitations?
  - X = theorem prover
  - X = model checker

Taxonomy of Approaches



NSA and Formal Verification

- NSA (NCSC) was a major source of funding for formal verification systems in the US in the mid-70's to mid-80's.

- The focus was on building **security kernels** relying on use of a *reference monitor* [Anderson72], which is supposed to satisfy three properties:
  - It must mediate all accesses to objects.
  - It must be tamperproof.
  - It must be small enough to be subjected to *analysis* and test.

- One form of "*analysis*" is formal verification.

Bell-LaPadula Model

- Bell-LaPadula model
  - Subjects (e.g., users), objects (e.g., files), access modes (e.g., read, write)
  - State machine semantics
- *-property
  - Write Up and Read Down
    - Jim (confidential) needs to be able to read data from my (unclassified) files to make a salary decision (confidential).
    - Why no Read Up?
      - Then I (unclassified) can see Doug's salary (confidential).
    - Why no Write Down?
      - Don't want Jim to leak Doug's salary (confidential) to me (unclassified).
    - Prevents nontrusted subjects from simultaneously having read access to information at one level and write access to information at a lower level

The Orange Book

- Hierarchy of security levels
  - D: Minimal
    - Reserved for systems that have been evaluated but failed to meet requirements of higher evaluation classes
  - C: Discretionary (3 classes)
    - Provide "need to know" protection and accountability of subjects and their actions.
    - Like UNIX access control. I.e., users have control.
  - B: Mandatory (2 classes)
    - System, not a user, labels objects according to a fixed policy.
    - Requires system developer to state policy model, provide a spec of TCB, and *evidence to demonstrate the reference monitor has been implemented correctly*.
  - A: Verified
    - Like B except requires use of **"formal security verification methods"** for the *evidence*.
    - MITRE SCOMP: 8-bit Honeywell XTS-200 proc.
    - Ford Aerospace: Multinet Gateway System

Four Theorem Proving Systems (1984)

- FDM (SDC)
  - InaJo: proof by contradiction (tableaux method)
  - nondeterministic state machines
- HDM (SRI)
  - Special: lambda-calculus, second-order pred. calculus, Hoare sentences, Pascal code
  - TP: decision procs. for 1st-order logic and subset of int. arith.
- Gypsy (Texas)
  - Pascal-like program verification system
  - vc generator
- Affirm (USC/ISI)
  - algebraic specifications
  - rewrite rule engine (Knuth-Bendix)

What Does it Mean for a System to be Secure?

- Basic Security Theorem
  - A state is *secure* if it satisfies the *- property (and two other properties).
  - A system is *secure* if and only if the initial state is secure and the states before and after each action are secure.

What Does it Mean to *Prove* that a System is Secure?

- McLean's blasting of BST
  - In IPL '85 McLean showed that you can prove an obviously insecure system (which satisfies the reverse of the *- property) secure.
    - This dagger- property allows subjects to transfer information from higher security levels to lower security levels.
  - In a draft paper McLean constructed an obviously insecure model called System Z that satisfies BST.
    - System Z had only one action that downgrades all subjects and objects to the lowest possible level.
  - Conclusion: "We have shown that the BST does nothing to establish that a system is really secure."