SAND2012-10574P

Applications of Complexity Science to Digital Systems

Jackson Mayo

Computer Sciences and Information Systems
Sandia National Laboratories, Livermore, California

December 13, 2012

Sandia National Laboratories is a multiprogram laboratory operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

The problem

- **Science today confronts "complex" systems that behave as large-scale information networks and do not yield to traditional analysis**
  - Complex systems can be engineered or evolved

Infrastructure

Computers

Societies

  - Basis for their intractability: Turing's halting problem
- **How can we design/analyze these systems?**
  - In particular, how can we deal with widespread digital systems and consequent cybersecurity problems?

Characteristics of complexity

- **Complex systems are characterized by large numbers of interacting entities where even a few entities can strongly affect system behavior**
- **Complex systems are irreducible; their behavior is emergent and not evident a priori, but is accessible via observation and simulation**
- **Examples are ubiquitous**
  - Living things and ecosystems
  - Human societies, economies, and institutions
  - Highly engineered artifacts – e.g., airplanes, nuclear weapons
  - Large-scale infrastructure – e.g., power grids
  - Computer software, hardware, and networks

Sandia National Laboratories

The complexity problem has its roots in theoretical computer science

- **Theorem (Turing 1936, Rice 1953): No algorithm exists to predict a priori the behavior of a generic information processing system**
  - i.e., such a system is undecidable even if deterministic
  - Abstract significance: A generic system with an unbounded number of states is undecidable
  - Practical significance: A real-world system, with a finite exponentially large number of states but otherwise generic, is *effectively* undecidable

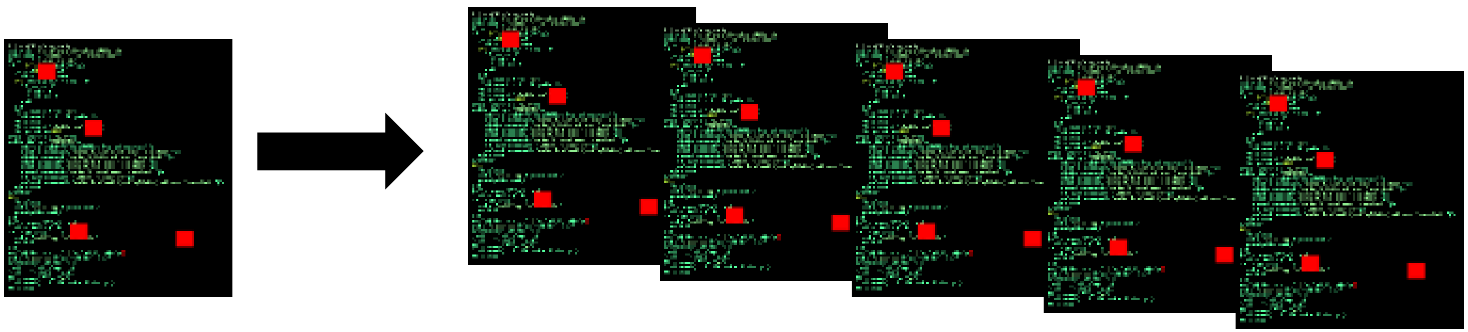
$\overset{?}{>}$

What solutions are possible?

- **We are researching improved analysis and design approaches for complex systems**
  - Because complex systems are intractable in general…
  - These approaches must rely on non-generic features resulting from how the system is engineered or evolved
  - That is, complex systems must be specially constrained to be analyzable
- **Two vital strategies:**
  - Reduce the complexity to enable exhaustive analysis by *formal methods* (widely used in industry)
  - Structure the complexity to enable probabilistic analysis when exploring the entire state space is infeasible

Economies of scale in computing: Friend and enemy

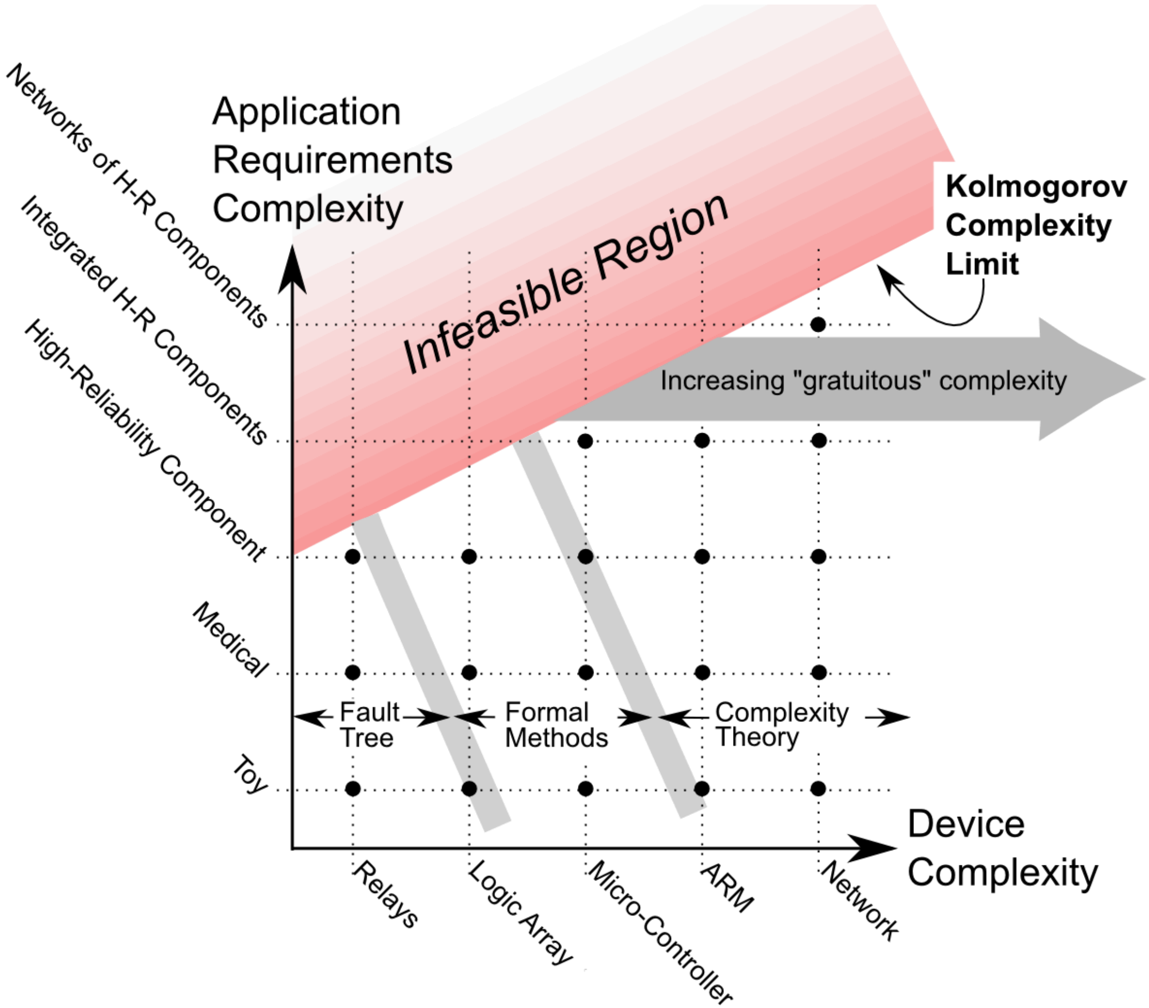
- **Enormously complex hardware and software is created at enormous cost**
  - Cost is recouped by stamping out millions of identical copies



- **A kid in his basement can make it do something interesting but unknown (unpredictable). He can be certain he can do the same thing to your desktop PC (deterministic)**
- **In the general case, all digital designs share these problems**

Solution: Make the design less general, more analyzable

Complexity space illustrates tradeoffs in device engineering and analysis



- **Formal methods research directions:**
  parallel scalability of algorithms,
  mixed analog-digital system verification
- **Complexity theory research directions:**
  diverse redundancy as a vulnerability-tolerant design,
  more general criteria for resilient designs

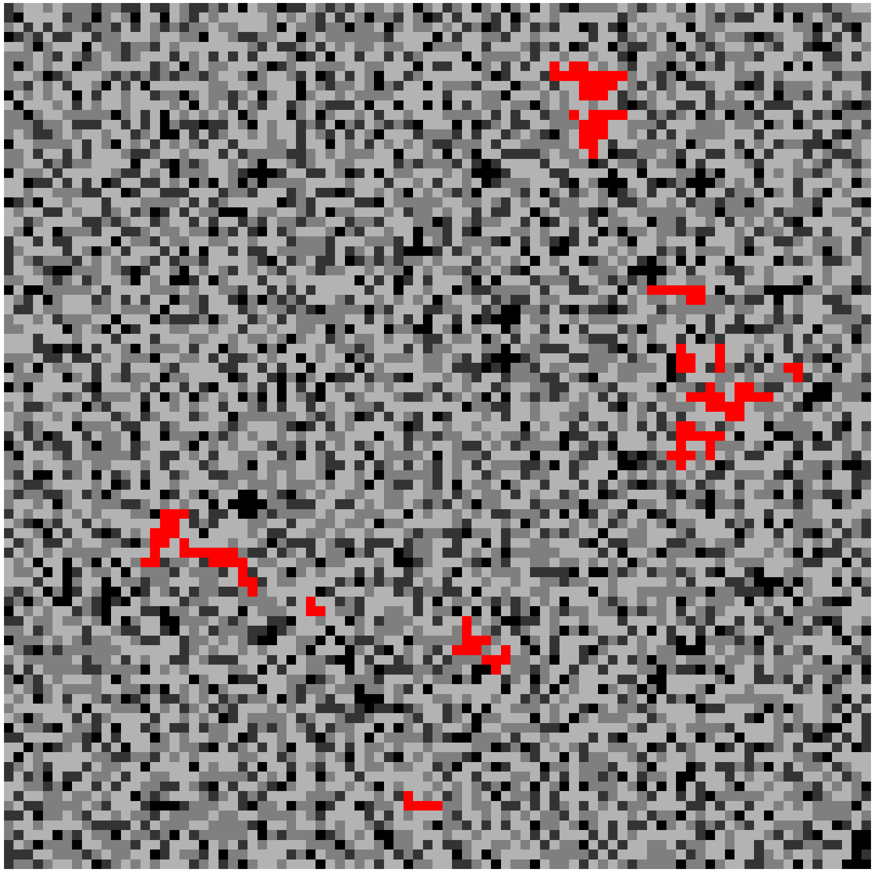
Sandia National Laboratories

Formal methods are a bridge to complexity, filling an important gap

- **Formal methods use computer analysis to verify digital systems rigorously and exhaustively**
  - Applicable to less complex systems that are still beyond the reach of manual analysis
  - Widely used in high-consequence industrial applications such as aviation and medical devices
- **Verification of components does not generally translate to verification of whole system**
- **Irreducible complexity enters when exploring entire state space is infeasible**
  - Reliability and security assertions become probabilistic
- **Both formal verification and complexity science are vital for gaining confidence in digital systems**

Self-organized criticality is a simple example of emergent behavior

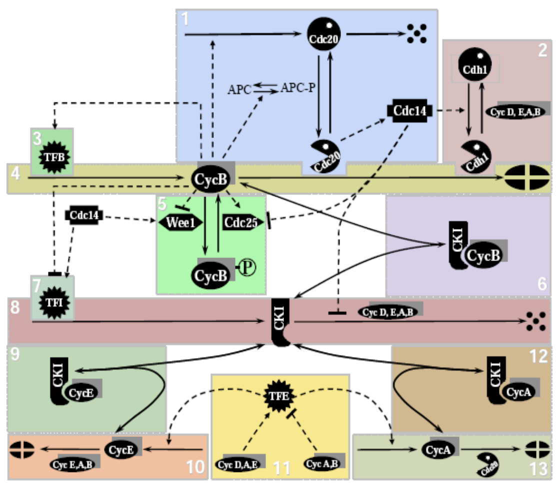
- **"Sandbot": cyber model of coordinated malware**
- **SOC (Bak et al. 1987) is *spontaneous* development of fractal phenomena with power-law distributions**
  - Similar to thermodynamic criticality but without tuning
- **Illustrated by sandpile model: physics-like cellular automaton**
  - Sand is sprinkled randomly
  - Avalanches occur at all scales



Sandia National Laboratories

Complexity is a fact of "life"

- **Biological phenomena are a prototype and inspiration for many complex domains**
  - Life involves a large chemical regulatory network



Eukaryotic cell-cycle regulation

  - "Game of Life" model is based on population dynamics
  - Bio concepts pervade computing (viruses, mutations)
- **Biology typifies complex couplings of manmade systems – economy, energy, cybersecurity**

Robustness is key to understanding real-world systems with "organic" behavior

- **Highly optimized tolerance (HOT, Carlson & Doyle 1999): Systems *designed* or *selected* to perform well despite perturbations**
- **HOT systems exhibit power-law distributions but have organic structure (not self-similar or fractal)**

HOT

≠

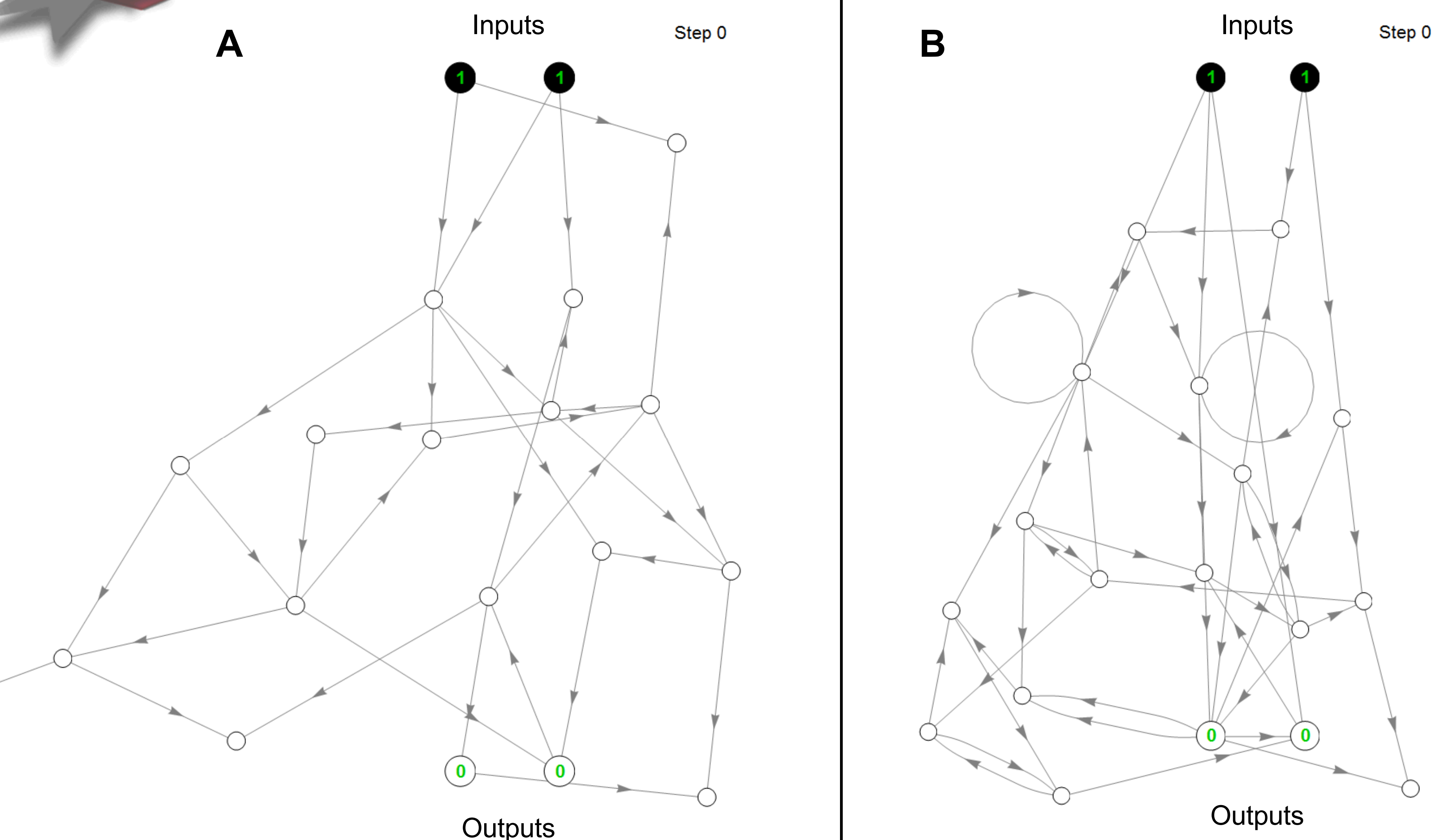
Fractal

- **Adapted robustness to one set of perturbations induces extra fragility to different perturbations**
- **Indeed, rare but catastrophic failures are seen in highly engineered/evolved systems**
  - Electrical blackouts, financial panics, epidemics, cyber shutdown of Estonia, etc.

Current work shows ways to address "whole system" robustness and stability

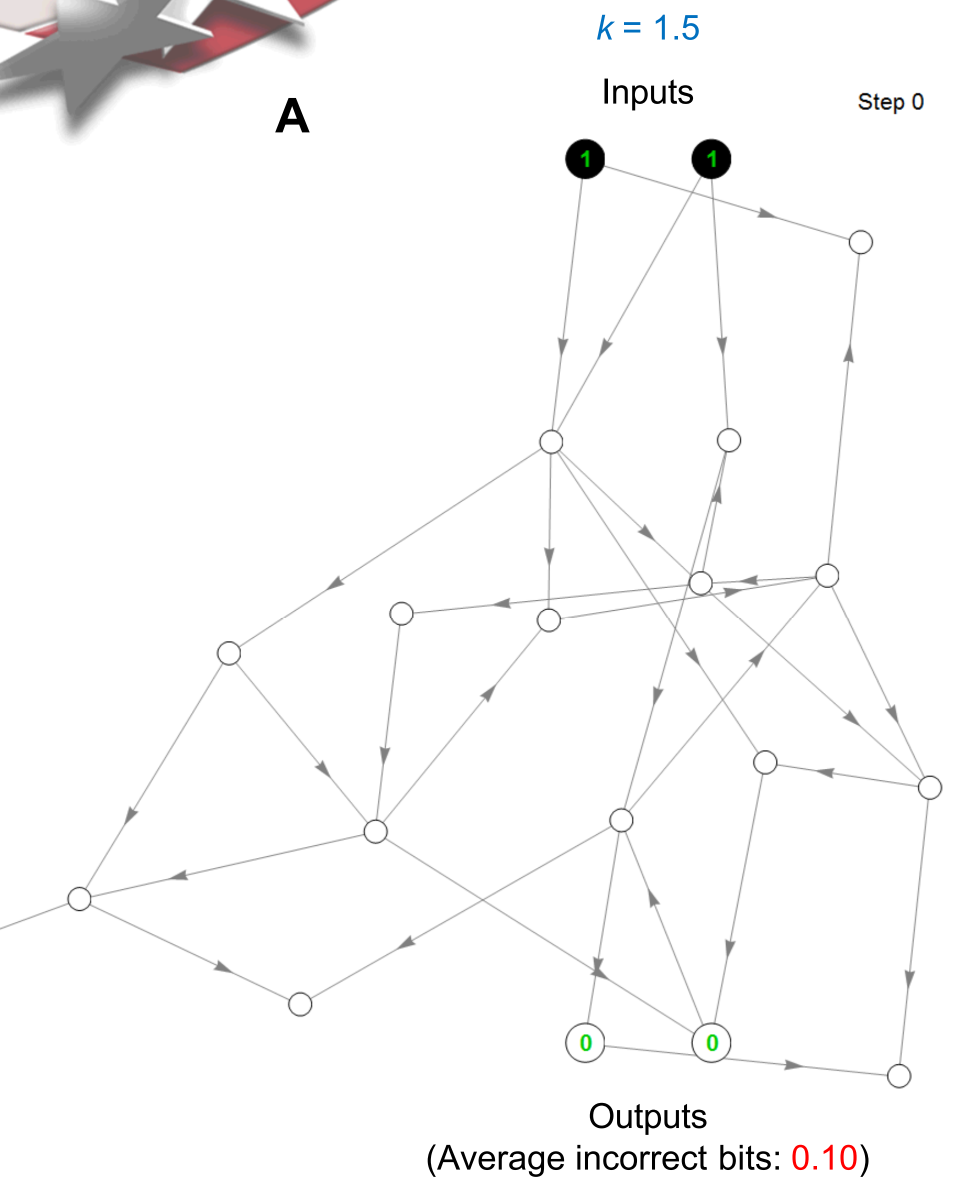
- **Cybersecurity vision: Create high-consequence digital systems (e.g., smart-meter networks) in new ways, so that they are analyzable**
  - Seek to understand computers as dynamical systems
- **Toy example: "Growing" a digital circuit to add two 1-bit numbers – a half adder**
- **There are many ways of composing logic gates to implement this functionality**
- **Next slide shows two such "grown" circuits; each performs as a half adder when run for 20 steps**
  - Shown correctly adding 1 + 1 to get the binary result 10
  - They also respond correctly to the other possible inputs

Sandia National Laboratories

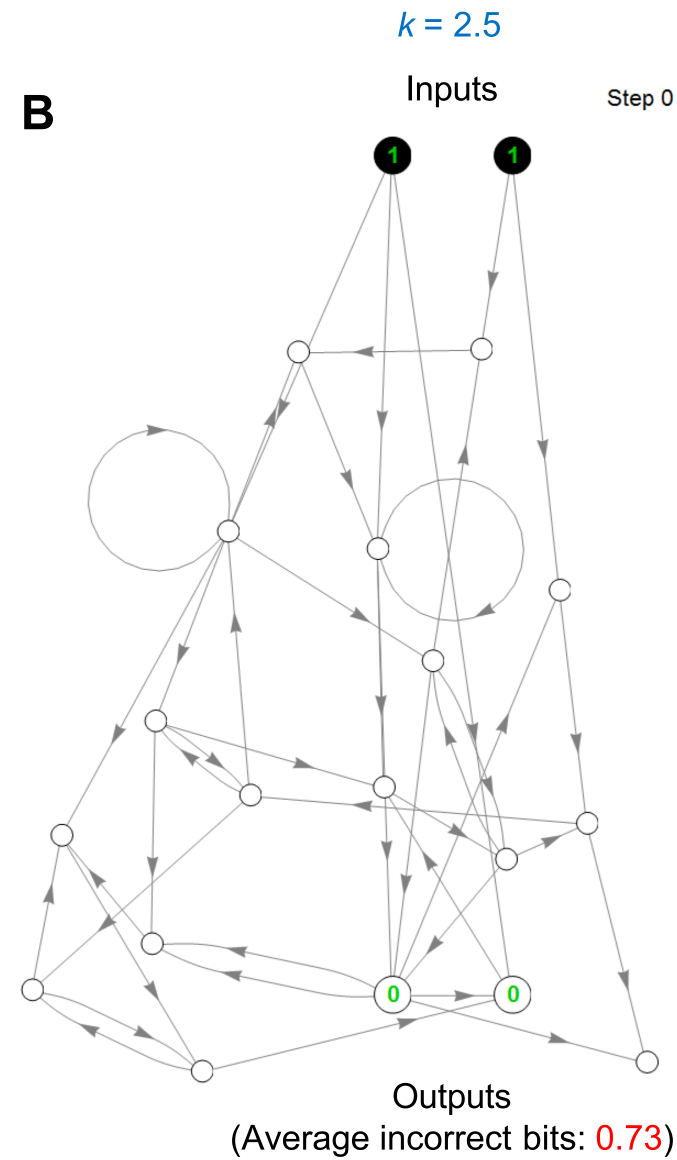

A
Inputs
Step 0
1
1
0
0
Outputs
B
Inputs
Step 0
1
1
0
0
Outputs

What distinguishes the two implementations? *Resilience*

- **Resilience of a digital model to bit errors can be assessed via growth or damping of perturbations**
  - Bit errors can represent breakdown of digital model, or effect of untested states within the digital space
  - Networks transition from stable to unstable based on connectivity and logic (generalizing Kauffman 1969)
- **Next slide: runs with 1% error rate per update**
  - States that deviate from the ideal run are outlined in red
- **Circuit A has much less error in final output (greater resilience) than circuit B – why?**
  - Here, average inputs per node (k) makes the difference
  - More of our circuit analysis: Seshadhri et al. PRL 2011

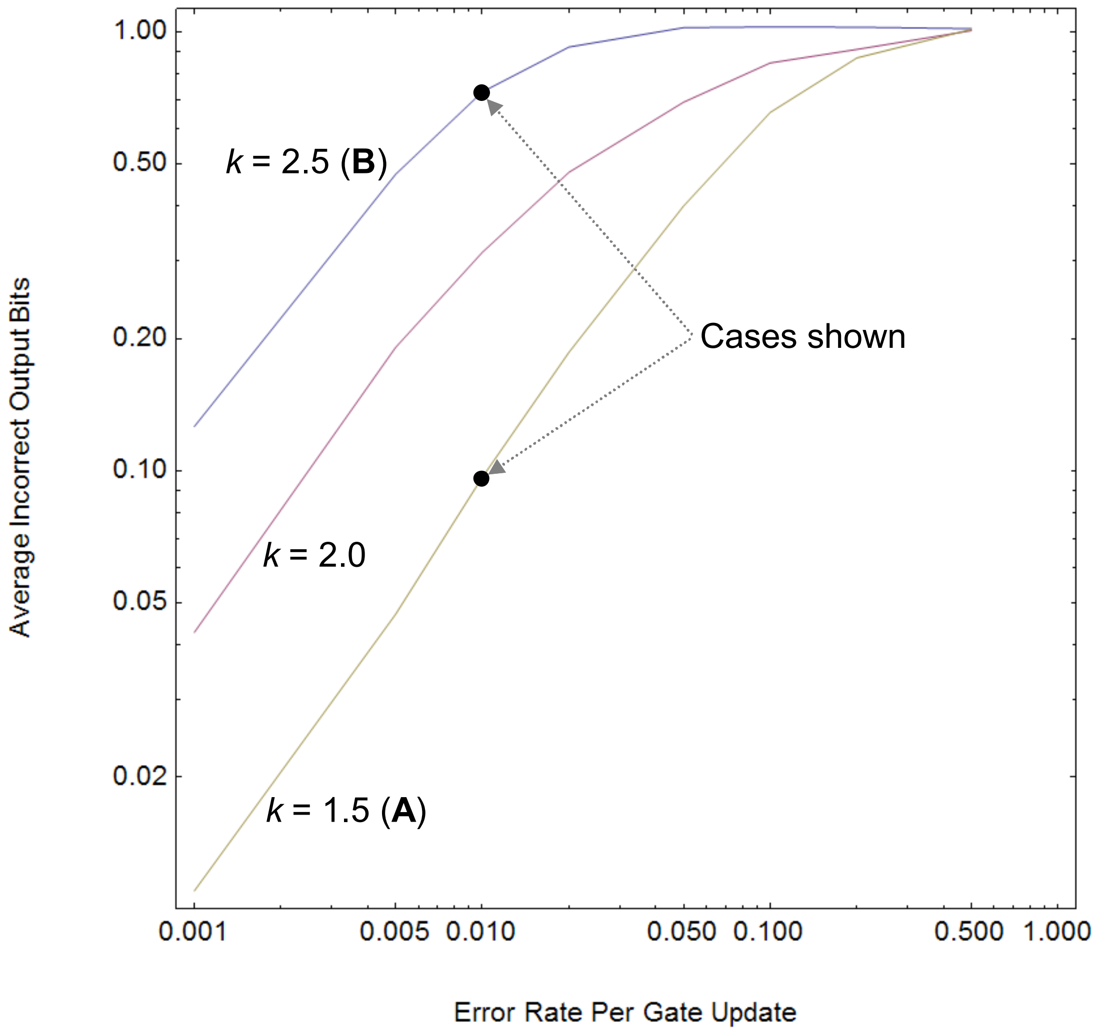


k = 1.5
A
Inputs
Step 0
1
1
0
0
Outputs
(Average incorrect bits: 0.10)



k = 2.5
B
Inputs
Step 0
1
1
0
0
Outputs
(Average incorrect bits: 0.73)

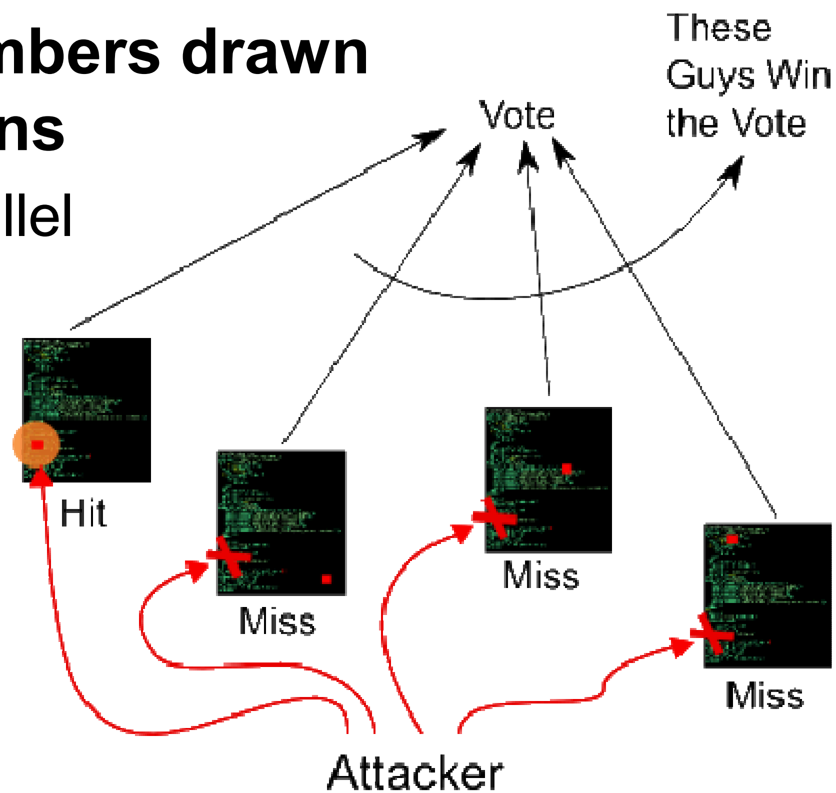
Example illustrates potential to quantify resilience implications of designs



- **Results for these half-adder circuits can be obtained by brute testing**
- **Systematic relations to real-world design parameters enable assessing potential catastrophic failures too rare to be found reliably through testing**

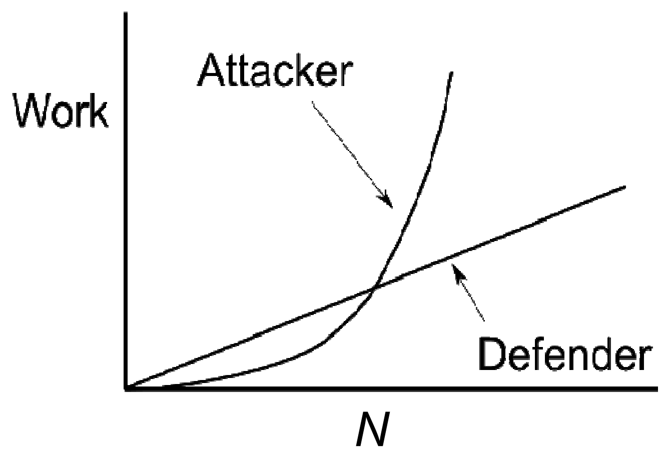
Bio-inspired "diverse redundancy" can be leveraged for cybersecurity

- **Use a voting system with members drawn from the set of implementations**
  - Input processed by each in parallel
  - Outputs compared to determine response
- **Keep intended functionality while varying vulnerabilities over space and time**
- **Similar to redundancy for physical fault tolerance**
- **Diversity leverages a simple trust anchor (the voting unit) for benefits at the *complex system level***



Analyzable statistics arise from an ensemble of undecidable programs

- **For a specific feature set, there is a probability P_v that a particular member of the set of implementations will be susceptible to vulnerability *v*. For a voting system of size *N*:**
  - The probability of success for the attacker is $(P_v)^{N/2}$
  - The attacker "work" is the expected number of tries: $(P_v)^{-N/2}$
  - The work for defender is the cost of producing *N* implementations: $\propto N$

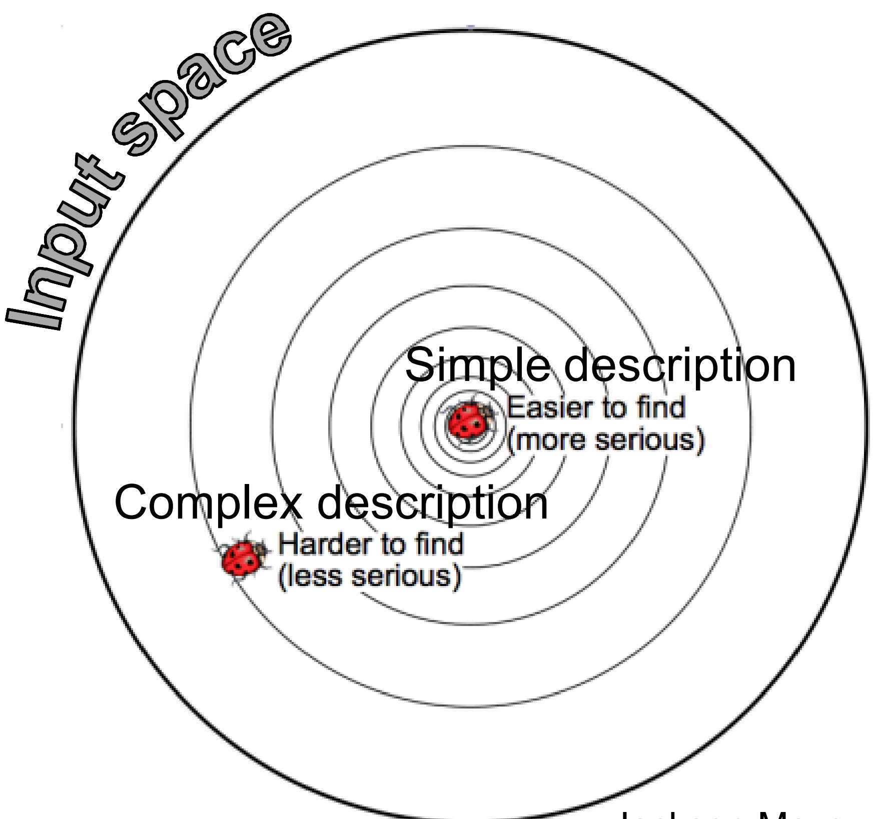


How diversity's benefits can be assessed

- **Fuzzing approaches**
  - Fuzzing (automated randomized testing) can discover faults in individual implementations and in voting systems, and guide selection of the implementations
  - Using the complexity perspective, we developed a systematic way to generate test inputs for fuzzing, published in 2011 Oak Ridge cybersecurity workshop
- **Formal approaches**
  - Model checkers (e.g., NuSMV) can exhaustively evaluate simple programs and thus can tell us how often the voting system we create is *provably* fault-free
  - We have implemented this technique for "string recognizer" circuits, with promising results

Complexity measure leads to targeted fuzzing strategies

- **Evolved and designed systems have coherence that makes it useful to fuzz in “simpler” spaces**
- **Example: Fuzzing a program with patterns *close to the nominal input* is more likely to find faults**

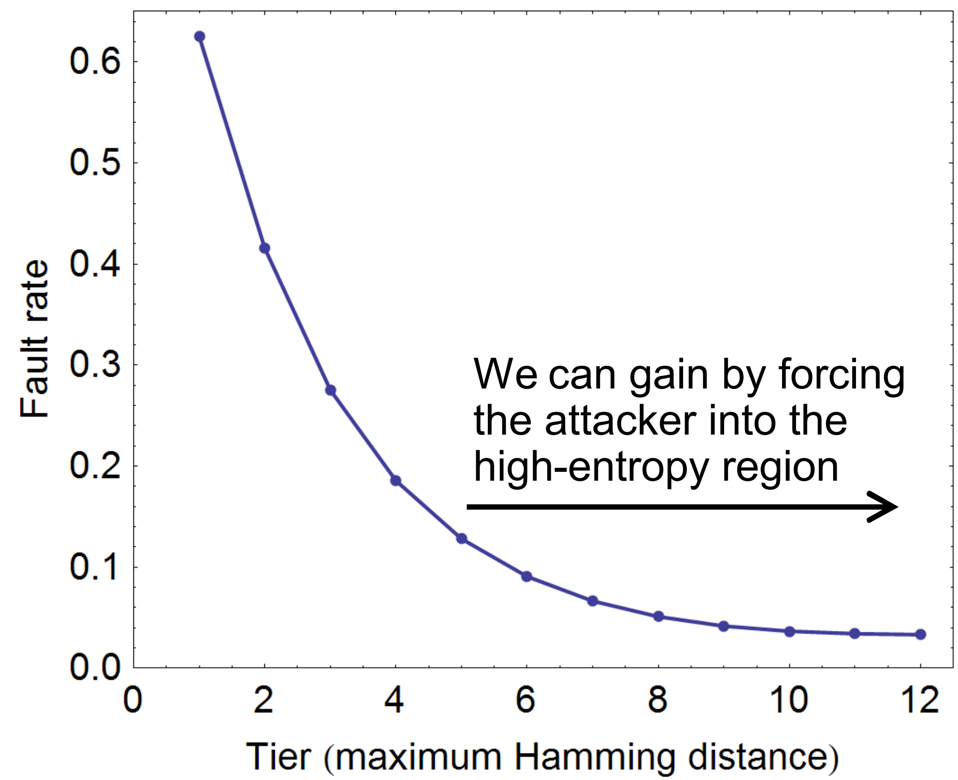


- **More generally: Inputs that have a *simple description* (relative to available information) should be targeted for coverage because they form a smaller “corner” space (also more attractive to attacker)**

Sandia National Laboratories

Fault statistics of simple "grown" programs seem to corroborate

- **16-bit "string recognizer" circuit (password checker) has small enough input space for *exhaustive* fuzzing**
- **We measure complexity ("entropy") by an edit function from the gold string, initially bitwise (approximate entropy by Hamming distance)**
- **As expected, faults are most common close to the gold string**



Sandia National Laboratories

NuSMV formal analysis of diverse string recognizers exposes voting benefit