

# Probabilistic Reasoning in Networks

Dexter Kozen  
Cornell

Cornell–Princeton  
Center for Network Programming  
Kickoff Retreat  
20 June 2016

(thanks to Nate Foster, Praveen Kumar, Konstantinos Mamouras,  
Mark Reitblatt, Alexandra Silva, Steffen Smolka)

# Challenges when Designing a Programming Language/Specification Language/Logic

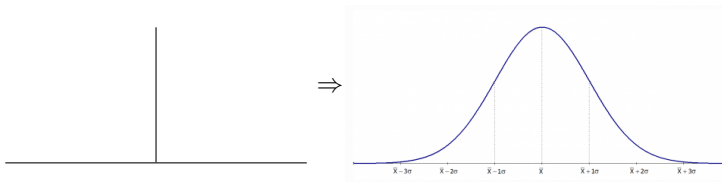
- ▶ Understand the typical problems
- ▶ Find good abstractions/primitives/operators
- ▶ Expressiveness vs simplicity & ease of use
- ▶ Scalability via compositionality
- ▶ Correct & coherent - provide strong guarantees
- ▶ Automation
- ▶ Robust

# Ordinary Reasoning in Networks

Ordinary reasoning is **truth-functional**

- ▶ Will this packet reach its destination?
- ▶ If a link fails, are there alternative routes?
- ▶ Can these two hosts communicate?
- ▶ Is there a forwarding cycle?

# Probabilistic Reasoning



Probabilistic reasoning is more **calculational**

- ▶ What is the **probability** that this packet will reach its destination?
- ▶ What is the **likelihood** of link failure in a given time interval under a certain load?
- ▶ Given a routing scheme, what is the **expected time** to packet delivery?
- ▶ For this randomized broadcast protocol, what is the **expected time** before saturation?

# Modeling Probabilistic Computation

## Sources of Randomness

- ▶ External (assume stochastic traffic, deterministic program)
- ▶ Internal (use of random number generator)

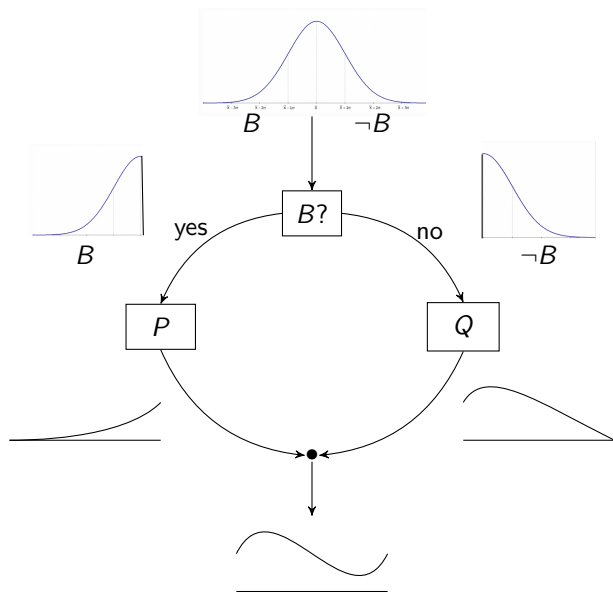
## Basic Tools

- ▶ Simulation by forward propagation
- ▶ Integration

## Additional Difficulties for Networks

- ▶ Packet filtering vs state modification

# Example – Conditional Test



# Probabilistic NetKAT (ProbNetKAT)

Extend NetKAT with constructs for reasoning about probabilistic behavior

Goals:

- ▶ Programming/Specification Language/Logic ✓
- ▶ Formal semantics extending NetKAT conservatively ✓
- ▶ Deduction rules ?
- ▶ Automation ?

<http://frenetic-lang.org/>

Thanks!

