Logic-Based Methods for Assurance of Complex System Performance (DRAFT)

NASA IV&V Workshop

11–13 September 2012

Morgantown, WV

Dr. Ralph L. Wojtowicz

Shepherd University
Shepherdstown, WV
rwojtowi@shepherd.edu

Baker Mountain Research Corporation
Yellow Spring, WV
ralphw@bakermountain.org

Outline

1. Introduction
   - Autonomous Systems
   - Model Checking
   - Categorical Logic
2. Model Checking
   - Probabilistic Model Checking
   - Example: Knuth-Yao Simulation
   - History
3. IPv4 Protocol
   - Concept
   - DTMC Model
   - Protocol Details
   - PTA Model
4. τN Theories
   - Syntax
   - Categorical Semantics
   - Models and Morphisms
5. Conclusions

Examples of Autonomous Platforms

Failures of Autonomous Systems

- Loss of the Mars Climate Orbiter in 1999
- Deaths of six cancer patients subjected to overdoses by the Therac-25 computerized radiation therapy machine in 1985–1987
- Airshow crash of Airbus A320 in 1988 in Mulhouse, France
- Airshow crash of China Airlines Airbus A-300 in 1994
- Temporary loss of the Dallas-Fort Worth air traffic system in 1990
- British destroyer H.M.S. Sheffield was sunk by exocet missile as a result of errors in the ship's missile defense software
- Araine 5 exploded forty seconds after liftoff on 4 June 1996 due to software error
- Gemini V capsule in 1965 missed its landing point in the Atlantic by 100 miles due to software error

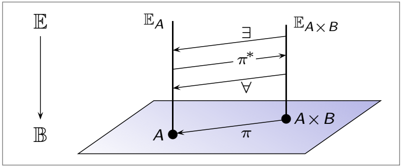
Formal Approaches to Software Verification

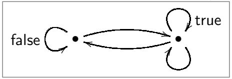
- Type theory
  - Type system gives a tractable syntactic method for proving the absence of certain program behaviors
  - Can be used to enforce highest level of system conformance to specification
  - Complete, formal system specifications are usually not available
  - Logical inference in rich type systems has high computational complexity
- Model checking
  - Finite-state model is exhaustively analyzed to test certain aspects of system behavior
  - State explosion problem resulting from aggregation of system components
- Research objective: develop syntactic inference systems that are applicable to model checking logics

Logics

- Logics are mathematical models of inference. Like models of physical phenomena, logics are developed with varying levels of fidelity in response to their intended applications.
- Mathematical logic plays fundamental roles in aspects of machine learning (Mitchell), AI (Russell & Norvig) and programming language theory (Pierce)
- Fundamental insight: Logics are interpreted in categories (Lawvere: 1963)

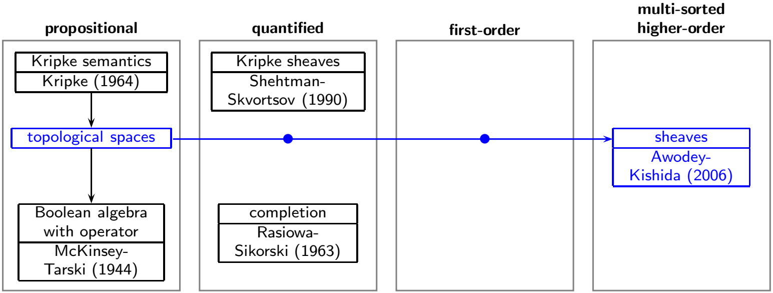
logic	semantic category	example
Horn	Cartesian	meet semi-lattice
first-order intuitionistic	Heyting	open sets
λ-calculus	Cartesian closed	group actions
first-order S4 modal	sheaf on topological space	infinite helix
higher-order intuitionistic	topos	directed graphs
linear	$*$-autonomous	relations





Categorical Logic Success Story: Modal Logic

- Modal logic
  - Modalities: logical operations that qualify assertions about the truth of statements
  - Necessity $\Box$ and possibility $\Diamond$
  - Knowledge of autonomous agents
  - Safety, security, and correctness of programs
- Semantics of S4 modal logic

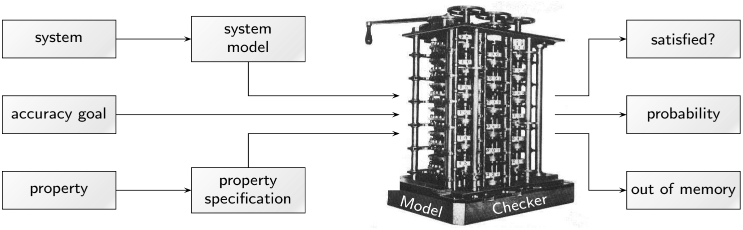


- Counterexamples to Barcan formulae: $\Box\exists \vdash \exists\Box$ and $\forall\Box \vdash \Box\forall$

Outline

1. Introduction
   - Autonomous Systems
   - Model Checking
   - Categorical Logic
2. Model Checking
   - Probabilistic Model Checking
   - Example: Knuth-Yao Simulation
   - History
3. IPv4 Protocol
   - Concept
   - DTMC Model
   - Protocol Details
   - PTA Model
4. τN Theories
   - Syntax
   - Categorical Semantics
   - Models and Morphisms
5. Conclusions

Probabilistic Model Checking Concept

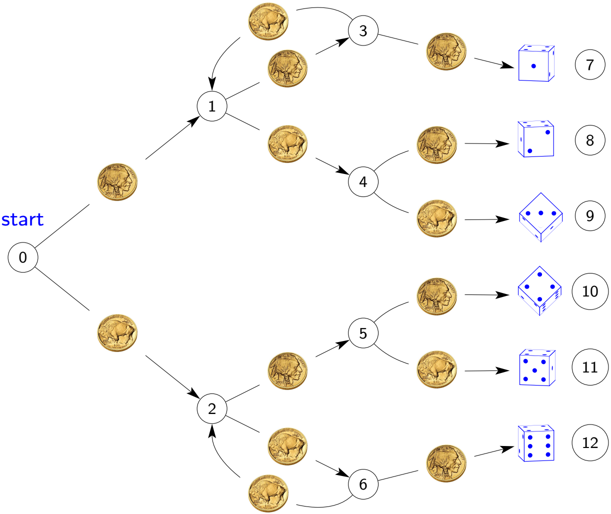


Model	Specification Language
Discrete Time Markov Chain	Probabilistic Computation Tree Logic
Markov Decision Process	Probabilistic Computation Tree Logic
Continuous Time Markov Chain	Continuous Stochastic Logic
Probabilistic Timed Automaton	Probabilistic Timed Computation Tree Logic

Research effort has focused on

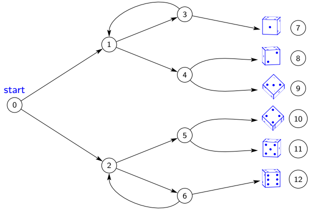
- Syntactic inference rules (sequent calculus)
- Applications: networking protocols, social network dynamics, etc.

Knuth-Yao 6-Sided Die Simulation

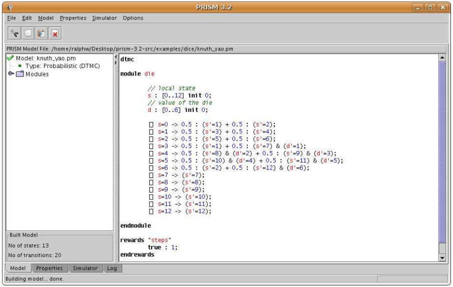


Properties of the Knuth-Yao Simulation

PCTL formula	type	satisfied by
start	state	$s = 0$
⚀	state	$s = 7$
$X[$ ⚀ $]$	path	$(s_0, s_1, \ldots)$ with $s_1 = 7$
$\Diamond$ ⚀	path	$s_n = 7$ for some n
$P_{>0}[\Diamond$ ⚀ $]$	state	states from which ⚀ can occur: 0, 1, 3, 7
start $\wedge P_{=1/6}[\Diamond$ ⚀ $]$	state	0 iff ⚀ has probability 1/6
start $\wedge P_{=1}[\Diamond$ ⚀ $\vee \cdots \vee \Diamond$ ⚅ $]$	state	0 iff termination with probability 1



PRISM: GPL Probabilistic Model Checker



PRISM Model File: /home/ralphw/Desktop/prism-3.2-src/examples/dice/knuth_yao.pm

Model: knuth_yao.pm
- Type: Probabilistic (DTMC)
- Modules

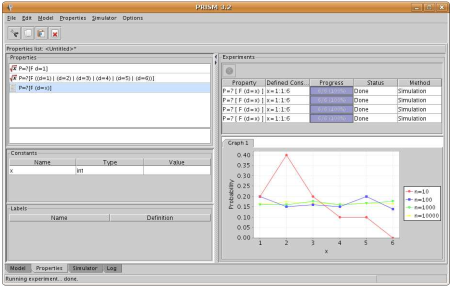
Built Model

No of states: 13

No of transitions: 20

```
dtmc

module die

        // local state
        s : [0..12] init 0;
        // value of the die
        d : [0..6] init 0;

        [] s=0 -> 0.5 : (s'=1) + 0.5 : (s'=2);
        [] s=1 -> 0.5 : (s'=3) + 0.5 : (s'=4);
        [] s=2 -> 0.5 : (s'=5) + 0.5 : (s'=6);
        [] s=3 -> 0.5 : (s'=1) + 0.5 : (s'=7) & (d'=1);
        [] s=4 -> 0.5 : (s'=8) & (d'=2) + 0.5 : (s'=9) & (d'=3);
        [] s=5 -> 0.5 : (s'=10) & (d'=4) + 0.5 : (s'=11) & (d'=5);
        [] s=6 -> 0.5 : (s'=2) + 0.5 : (s'=12) & (d'=6);
        [] s=7 -> (s'=7);
        [] s=8 -> (s'=8);
        [] s=9 -> (s'=9);
        [] s=10 -> (s'=10);
        [] s=11 -> (s'=11);
        [] s=12 -> (s'=12);

endmodule

rewards "steps"
        true : 1;
endrewards
```

Building model... done.



Properties list: <Untitled>*

Properties

- P=?[F d=1]
- P=?[F ((d=1) | (d=2) | (d=3) | (d=4) | (d=5) | (d=6))]
- P=?[F (d=x)]

Constants

Name	Type	Value
x	int	

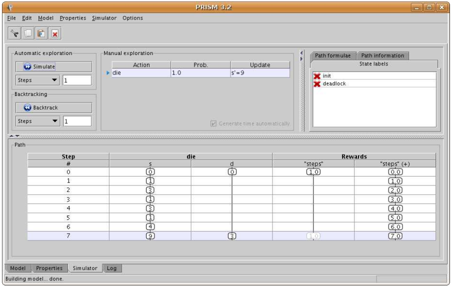
Labels

Name	Definition

Experiments

Property	Defined Cons...	Progress	Status	Method
P=? [F (d=x)]	x=1:1:6	6/6 (100%)	Done	Simulation
P=? [F (d=x)]	x=1:1:6	6/6 (100%)	Done	Simulation
P=? [F (d=x)]	x=1:1:6	6/6 (100%)	Done	Simulation
P=? [F (d=x)]	x=1:1:6	6/6 (100%)	Done	Simulation

Running experiment... done.



Steps 1

Steps 1

Manual exploration

Action	Prob.	Update
die	1.0	s'=9

☑ Generate time automatically

State labels

- ☒ init
- ☒ deadlock

Path

Step	die		Rewards	
#	s	d	"steps"	"steps" (+)
0	0	0	1.0	0.0
1	1			1.0
2	3			2.0
3	1			3.0
4	3			4.0
5	1			5.0
6	4			6.0
7	9	3	1.0	7.0

Building model... done.

www.prismmodelchecker.org

Model Checking Historical Sketch

1932 A. Church introduced untyped λ-calculus

1959 C. Lee introduced binary decision diagrams

1966 C. A. Petri wrote dissertation on Petri nets. D. Scott and P. Krauss wrote "Assigning Probabilities to Logical Formulas"

1968 Minsky introduced labeled transition systems

1969 D. Scott defined logic of computable functions of higher types

1974 D. E. Knuth received A.C.M. Turing Award

1976 D. Scott received Turing Award

1977 A. Pneuli proposed temporal logic model checking concept

1979 *Computer Aided Verification* colloquium started at Grenoble, FR

1980 R. Milner defined CSS (calculus of communicating systems)

1981 Clarke and Emerson and Sifakis independently published papers on temporal logic model checking

1982 CESAR Sifakis logic model checker developed at Grenoble

1984 P. Martin-Löf introduced intuitionistic type theory

1986 EMC CTL model checker developed at CMU

1986 R. Bryant popularized binary decision diagram in model checking

1987 Estelle model checker developed

1987 MEC Dicky calculus model checker developed at Bordeaux

1991 R. Milner received Turning Award

1992 Esterel real-time model checker developed

1993 Multi-terminal decision diagrams developed

1994 R. Alur and D. L. Dill defined timed automata

1994 J. Sifakis et al. introduced TCTL

1996 A. Pneuli received Turing Award

1996 $\mathrm{E} \vdash \mathrm{MC}^2$ DTMC/PCTL and CTMC/CSL probabilistic model checker developed

1996 KRONOS timed automata model checker developed

1989 Edinburgh Concurrency Workbench developed

1997 Katis, Sabadini, and Walters introduced bicategories of processes

2000 A. C-C. Yao received Turing Award

2002 RAPTURE MDP/PCTL probabilistic model checker developed

2002 PRISM probabilistic model checker developed

2007 E. M. Clarke (CMU), E. A. Emerson (UTA), and J. Sifakis (CNRS, FR) received Turing Award

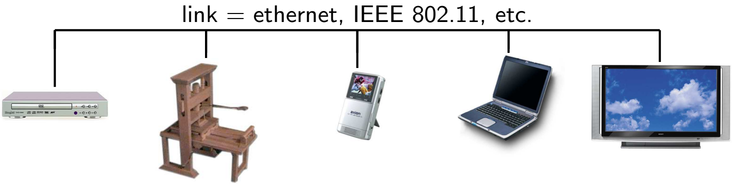
Outline

1. Introduction
   - Autonomous Systems
   - Model Checking
   - Categorical Logic
2. Model Checking
   - Probabilistic Model Checking
   - Example: Knuth-Yao Simulation
   - History
3. IPv4 Protocol
   - Concept
   - DTMC Model
   - Protocol Details
   - PTA Model
4. τN Theories
   - Syntax
   - Categorical Semantics
   - Models and Morphisms
5. Conclusions

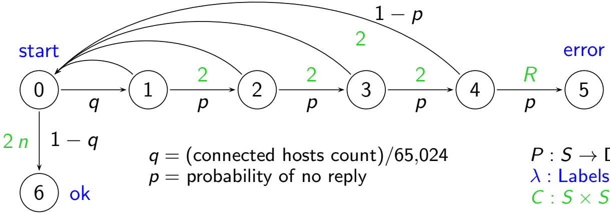
Dynamic Configuration of IPv4 Addresses

- Isolated network on a single link (e.g., no routers)
- No DHCP server or manual IP setup needed
- Upon connection, new host must:
  - Randomly select IP from a pool of 65,024

    169.254.1.0 – 169.254.254.255 (IANA assigned)
  - Probe for another host using that IP
  - Try again if IP is already in use
  - Claim IP if it is not in use

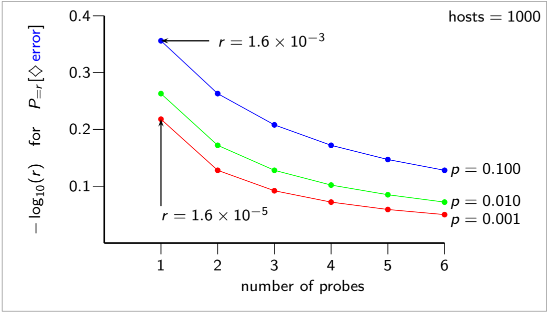


DTMC Model of the IPv4 Link-Local Protocol



$q =$ (connected hosts count)/65,024
$p =$ probability of no reply

$P : S \to \mathrm{Dist}(S)$
$\lambda : \mathrm{Labels} \to \mathcal{P}(S)$
$C : S \times S \to \mathbb{R}$



Dynamic Configuration of IPv4 Link-Local Addresses. www.ietf.org/rfc/rfc3927.txt. 2005.

Probabilistic Computation Tree Logic (PCTL)

- Presentation:

sorts	• S, Ω
types	• sorts, products, $\mathcal{P}S$ (states), $\mathcal{P}\Omega$ (paths)
function symbols	• $\epsilon : \Omega \to S$ • $\sigma : \Omega \to \Omega$ • $P_{\bowtie p} : \mathcal{P}\Omega \to \mathcal{P}S$ for each $p \in [0,1]$ and $\bowtie \in \{<, \leq, >, \geq\}$ • $E_{\bowtie c} : \mathcal{P}S \to \mathcal{P}S$ for each $c \in \mathbb{R}$
relation symbols	• $a \rightarrowtail S$

- State formulae:

$$a \qquad P_{\bowtie p}[\psi] \qquad P_{\bowtie p}[X[\varphi]] \qquad P_{\bowtie p}[U^{\leq k}[\varphi_1, \varphi_2]]$$

$$P_{\bowtie p}[U[\varphi_1, \varphi_2]] \qquad E_{\bowtie c}[\varphi]$$

$$\top \qquad \bot \qquad \varphi_1 \wedge \varphi_2 \qquad \varphi_1 \vee \varphi_2 \qquad \varphi_1 \Rightarrow \varphi_2$$

- Path formulae:

$$X[\varphi] \qquad U^{\leq k}[\varphi_1, \varphi_2] \qquad U[\varphi_1, \varphi_2] \qquad \Box[\varphi] \qquad \Diamond[\varphi]$$

DTMC Semantics of PCTL

- $S =$ set of states
- $\Omega =$ set of paths $\omega = (s_0, s_1, \ldots)$
- $\mathrm{Path}_s =$ set of paths ω with $s_0 = s$
- Probability measure p_s on Path_s
  - Cylinder $\Gamma(s_0, \ldots, s_n) =$ all paths with given prefix
  - Disjoint unions of cylinders form an algebra on Path_s
  - $p_s(\Gamma(s')) = \begin{cases} 1 & \text{if } s = s' \\ 0 & \text{otherwise} \end{cases}$
  - $p_s(\Gamma(s_0, \ldots, s_n)) = P(s_0, s_1) \cdot \cdots \cdot P(s_{n-1}, s_n)$
  - Extend p_s to a measure on the generated σ-algebra
- $s \models a$ iff s has label a
- $s \models P_{\bowtie p}[\psi]$ iff $p_s(\psi) \bowtie p$
- $s \models E_{\bowtie c}[\varphi]$ iff $\int_{\mathrm{Path}_s} \mathrm{cost}(\varphi)(\omega)\, dp_s \;\bowtie c$ where

$$\mathrm{cost}(\varphi)(\omega) = \begin{cases} \sum_{i=1}^{\min\{j \mid s_j \in \varphi\}} C(s_{i-1}, s_i) & \text{if } \exists j \in \mathbb{N}.\ s_j \in \varphi \\ \infty & \text{otherwise} \end{cases}$$

Protocol Details

- Parameters

PROBE_WAIT	1 sec	PROBE_NUM	3
PROBE_MIN	1 sec	PROBE_MAX	2 sec
ANNOUNCE_WAIT	2 sec	ANNOUNCE_NUM	2
ANNOUNCE_INTERVAL	2 sec	MAX_CONFLICTS	10
RATE_LIMIT_INTERVAL	60 sec	DEFEND_INTERVAL	10 sec

- Clocks and counters

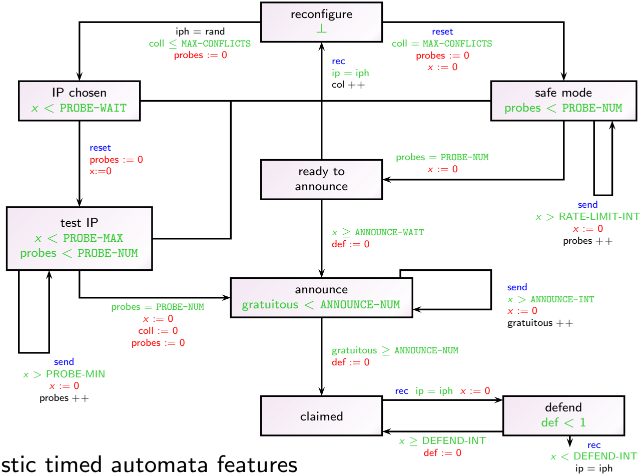
x = local clock	probes	gratuitous
coll	def	

- ARP Probe

<table>
<tr><td colspan="6">destination ethernet address</td><td colspan="6" rowspan="2">host ethernet address</td></tr>
<tr><td>ff</td><td>ff</td><td>ff</td><td>ff</td><td>ff</td><td>ff</td></tr>
<tr><td colspan="2">frame type</td><td colspan="2">hdw (eth)</td><td colspan="2">prot (IP)</td><td>(eth)</td><td>(IP)</td><td colspan="2">(ARP req)</td></tr>
<tr><td>08</td><td>06</td><td>00</td><td>01</td><td>08</td><td>00</td><td>06</td><td>04</td><td>00</td><td>01</td></tr>
<tr><td colspan="6" rowspan="2">host ethernet address</td><td colspan="4">host IP address</td></tr>
<tr><td>00</td><td>00</td><td>00</td><td>00</td></tr>
<tr><td colspan="6">target ethernet address</td><td colspan="4" rowspan="2">selected IP address</td></tr>
<tr><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td><td>00</td></tr>
</table>

- $P : \mathrm{Loc} \to \mathcal{P}\big(\mathrm{Zones}(\mathcal{X}) \times \Sigma \times \mathrm{Dist}(\mathcal{P}(X) \times \mathrm{Loc})\big)$

Probabilistic Timed Automaton Model



reconfigure
⊥

iph = rand
coll ≤ MAX-CONFLICTS
probes := 0

reset
coll = MAX-CONFLICTS
probes := 0
x := 0

rec
ip = iph
col ++

IP chosen
x < PROBE-WAIT

safe mode
probes < PROBE-NUM

reset
probes := 0
x:=0

ready to
announce

probes = PROBE-NUM
x := 0

send
x > RATE-LIMIT-INT
x := 0
probes ++

test IP
x < PROBE-MAX
probes < PROBE-NUM

x ≥ ANNOUNCE-WAIT
def := 0

announce
gratuitous < ANNOUNCE-NUM

send
x > ANNOUNCE-INT
x := 0
gratuitous ++

probes = PROBE-NUM
x := 0
coll := 0
probes := 0

send
x > PROBE-MIN
x := 0
probes ++

gratuitous ≥ ANNOUNCE-NUM
def := 0

rec ip = iph x := 0

claimed

defend
def < 1

x ≥ DEFEND-INT
def := 0

rec
x < DEFEND-INT
ip = iph

- Probabilistic timed automata features
  - Clocks and counters
  - Timing and counter constraints on states and transitions
  - Clock and timer resets
  - Digital clocks and region graph model checking algorithms

Outline

1. Introduction
   - Autonomous Systems
   - Model Checking
   - Categorical Logic
2. Model Checking
   - Probabilistic Model Checking
   - Example: Knuth-Yao Simulation
   - History
3. IPv4 Protocol
   - Concept
   - DTMC Model
   - Protocol Details
   - PTA Model
4. τN Theories
   - Syntax
   - Categorical Semantics
   - Models and Morphisms
5. Conclusions

τN-Theories — Syntax

- Signature
  - Types: sorts, 1, $A \times B$, N, PA
  - Function and relation symbols
- Terms
  - Variables $x : A$
  - Function application $f(t) : B$ if $f : A \to B$ and $t : A$
  - Products: $* : 1$, $\langle s, t \rangle : A \times B$ for $s : A$ and $t : B$ and
    $\mathrm{fst}(z) : A$ and $\mathrm{snd}(z) : B$ for $z : A \times B$
  - Natural number: $0 : N$, $\mathrm{succ}(t) : N$ if $t : N$ and $\mathrm{iter}_x(m, a, n) : A$ if $m : A$, $a : A$ and $n : N$ with x not free in a or n (or in $\mathrm{iter}_x(m, a, n)$)
  - Power: $\{x : A \,|\, \varphi\} : PA$ (with $\mathrm{FV}(\varphi)/\{x\}$ as set of free variables)
- Formulae
  - Atomic: $R(t)$, $(t =_A s)$ and $(s \in_A t)$ for $s : A$ and $t : PA$
  - Compound: $\varphi * \psi$ with $*$ one of $\wedge$, $\vee$, $\Rightarrow$
  - Negated: $\neg\varphi$
  - Quantified: $(\forall x)\varphi$ and $(\exists x)\varphi$

τN-Theories — Sequent Calculus

Structural Rules[1]	$(\varphi \vdash_{\vec{x}} \varphi)$ $\frac{(\varphi \vdash_{\vec{x}} \psi)}{(\varphi[\vec{s}/\vec{x}] \vdash_{\vec{y}} \psi[\vec{s}/\vec{x}])}$ $\frac{(\varphi \vdash_{\vec{x}} \psi)\ (\psi \vdash_{\vec{x}} \chi)}{(\varphi \vdash_{\vec{x}} \chi)}$
Implication	$\begin{array}{c}((\varphi \wedge \psi) \vdash_{\vec{x}} \chi) \\ \hline\hline (\varphi \vdash_{\vec{x}} (\psi \Rightarrow \chi))\end{array}$
Equality	$(\top \vdash_{x} (x = x))$ $((\vec{x} = \vec{y}) \wedge \varphi \vdash_{\vec{z}} \varphi[\vec{y}/\vec{x}])$
Quantification[2]	$\begin{array}{c}(\varphi \vdash_{\vec{x},y} \psi) \\ \hline\hline ((\exists y)\varphi \vdash_{\vec{x}} \psi)\end{array}$ $\begin{array}{c}(\varphi \vdash_{\vec{x},y} \psi) \\ \hline\hline (\varphi \vdash_{\vec{x}} (\forall y)\psi)\end{array}$
Conjunction	$(\varphi \vdash_{\vec{x}} \top)$ $((\varphi \wedge \psi) \vdash_{\vec{x}} \varphi)$ $((\varphi \wedge \psi) \vdash_{\vec{x}} \psi)$ $\begin{array}{c}(\varphi \vdash_{\vec{x}} \psi)\ (\varphi \vdash_{\vec{x}} \chi) \\ \hline\hline (\varphi \vdash_{\vec{x}} (\psi \wedge \chi))\end{array}$
Disjunction	$(\bot \vdash_{\vec{x}} \varphi)$ $(\varphi \vdash_{\vec{x}} (\varphi \vee \psi))$ $(\psi \vdash_{\vec{x}} (\varphi \vee \psi))$ $\begin{array}{c}(\varphi \vdash_{\vec{x}} \chi)\ \ (\psi \vdash_{\vec{x}} \chi) \\ \hline\hline ((\varphi \vee \psi) \vdash_{\vec{x}} \chi)\end{array}$
Product	$(\top \vdash_{x} (x =_{1} *))$ $(\top \vdash_{x,y} (\mathrm{fst}(\langle x, y\rangle) = x))$ $(\top \vdash_{z} (\langle \mathrm{fst}(z), \mathrm{snd}(z)\rangle = z))$ $(\top \vdash_{x,y} (\mathrm{snd}(\langle x, y\rangle) = y))$
Power[3]	$(\top \vdash_{w} (w =_{PA} \{x : A \mid x \in_{A} w\}))$ $((z \in_{A} \{y : A \mid \varphi\}) \dashv\vdash_{\vec{x},z} \varphi[z/y])$
Natural Numbers	$(\top \vdash_{\vec{y}} (\mathrm{iter}_{x}(m, a, 0) = a))$ $(\top \vdash_{\vec{y}} (\mathrm{iter}_{x}(m, a, \mathrm{succ}(n)) = m[\mathrm{iter}_{x}(m, a, n)/x]))$ $(((0 \in_{N} z) \wedge (\forall y)((y \in_{N} z) \Rightarrow (\mathrm{succ}(y) \in_{N} z))) \vdash_{z:PN} (\forall y)(y \in_{N} z))$

Contexts are suitable for the formulae that occur on both sides of $\vdash$.

[1] In the substitution rule, $\vec{y}$ contains all the variables of $\vec{x}$.

[2] Bound variables do not also occur free in any sequent.

[3] $w : PA$ is a variable.

τN-Theories — Models and Morphisms

- Any topos with natural number object is a suitable semantic category.
  - Soundness: If σ is provable in $\mathbb{T}$, then it is satisfied in all $\mathbb{T}$-models in such toposes. D4.3.17
  - Completeness: If σ is satisfied in all $\mathbb{T}$-models in such toposes, then it is provable. D4.3.19(b)
  - Peano Arithmetic: Any such topos has a model of PA. A2.5.4, A2.5.5
  - Recursive Partial Functions: $\mathbb{N}^k \rightharpoonup \mathbb{N}$ have interpretations.
- Logical Functors: cartesian and preserves exponentials, Ω and N
  - Preserve satisfaction of τN sequents
- Geometric morphisms: adjoint pairs

$$\begin{tikzcd}\mathcal{F} \arrow[d, shift left, "f_*"] \\ \mathcal{E} \arrow[u, shift left, "f^*"]\end{tikzcd}$$

with f^* cartesian
  - Preserve satisfaction of Horn sequents of $\mathcal{F}$ $(\top, \wedge)$
  - Preserve satisfaction of regular sequents of $\mathcal{E}$ $(\top, \wedge, \exists)$
  - Reflect natural number objects of $\mathcal{E}$

Citations in green are from Johnstone's *Sketches of an Elephant*. 2002.

Outline

1. Introduction
   - Autonomous Systems
   - Model Checking
   - Categorical Logic
2. Model Checking
   - Probabilistic Model Checking
   - Example: Knuth-Yao Simulation
   - History
3. IPv4 Protocol
   - Concept
   - DTMC Model
   - Protocol Details
   - PTA Model
4. τN Theories
   - Syntax
   - Categorical Semantics
   - Models and Morphisms
5. Conclusions

Conclusions

Forthcoming

References

- S. Awodey and K. Kishida. "Topology and Modality: The Topological Interpretation of First-Order Modal Logic". 2007. www.andrew.cmu.edu/user/awodey
- S. Eilenberg and C. C. Elgot. *Recursiveness*. Academic Press. 1970.
- B. Jacobs. *Categorical Logic and Type Theory*. Elsevier. 1999.
- P. E. Johnstone. *Sketches of an Elephant: A Topos Theory Compendium*. Oxford University Press. 2002.
- T. M. Mitchell. *Machine Learning*. 1997.
- B. C. Pierce. *Types and Programming Languages*. 2002.
- B. C. Pierce. *Advanced Types in Programming Languages*. 2004.
- PRISM web site: www.prismmodelchecker.org
- S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. 1995.
- J. J. M. M. Rutten, M. Kwiatkowska, G. Norman, and D. Parker. *Mathematical Techniques for Analyzing Concurrent and Probabilistic Systems*. American Mathematical Society. 2004.