Interactive verification of Markov chains: Two distributed protocol case studies

Johannes Hölzl and Tobias Nipkow

TU München

QFM 2012
28 August 2012
Introduction

- In the interactive theorem prover

Isabelle
Introduction

- In the interactive theorem prover

- Verified two case studies
Introduction

- In the interactive theorem prover

  Isabelle

- Verified two case studies
  - ZeroConf protocol (IPv4 address allocation)
Introduction

- In the interactive theorem prover

  Isabelle

- Verified two case studies
  - ZeroConf protocol (IPv4 address allocation)
  - Crowds protocol (anonymizing service)
Introduction

- In the interactive theorem prover

  ![Isabelle](image)

- Verified two case studies
  - ZeroConf protocol (IPv4 address allocation)
  - Crowds protocol (anonymizing service)

- Built on Isabelle’s probability theory and Markov chains
  Hölzl & Heller (ITP 2011), Hölzl & Nipkow (TACAS 2012)
Interactive theorem proving
Interactive theorem proving

- Mathematics, but checked by a computer
Interactive theorem proving

- Mathematics, but checked by a computer
- Powerful logics (e.g. ZF, CoC, HOL):
Interactive theorem proving

- Mathematics, but checked by a computer
- Powerful logics (e.g. ZF, CoC, HOL):
  - Can deal with infinite-state systems
Interactive theorem proving

- Mathematics, but checked by a computer
- Powerful logics (e.g. ZF, CoC, HOL):
  - Can deal with infinite-state systems
  - User-extensible
Interactive theorem proving

- Mathematics, but checked by a computer
- Powerful logics (e.g. ZF, CoC, HOL):
  - Can deal with infinite-state systems
  - User-extensible
- Too powerful to be fully automatic:
Interactive theorem proving

- Mathematics, but checked by a computer
- Powerful logics (e.g. ZF, CoC, HOL):
  - Can deal with infinite-state systems
  - User-extensible
- Too powerful to be fully automatic:
  *user needs to write proofs*
Interactive theorem proving

- Mathematics, but checked by a computer
- Powerful logics (e.g. ZF, CoC, HOL):
  - Can deal with infinite-state systems
  - User-extensible
- Too powerful to be fully automatic: *user needs to write proofs*
- Proof language and proof methods
Logic is HOL: functional programming + quantifiers
Logic is HOL: functional programming + quantifiers
Declarative proof language Isar
Isabelle/HOL

- Logic is HOL: functional programming + quantifiers
- Declarative proof language Isar
- Small kernel: each proof is reduced to primitive proof steps
Isabelle/HOL

- Logic is HOL: functional programming + quantifiers
- Declarative proof language Isar
- Small kernel: each proof is reduced to primitive proof steps
- Powerful proof methods (rewrite engine, Sledgehammer, ...)

5 / 24
Isabelle/HOL

- Logic is HOL: functional programming + quantifiers
- Declarative proof language Isar
- Small kernel: each proof is reduced to primitive proof steps
- Powerful proof methods (rewrite engine, Sledgehammer, ...)
- Important theories: datatypes, real analysis, measure theory, probability theory, Markov chains, ...
Case study: ZeroConf protocol
ZeroConf protocol

- Protocol to allocate an address in a link-local network, without central authority (RFC 3927)
ZeroConf protocol

- Protocol to allocate an address in a link-local network, without central authority (RFC 3927)
- We formalized the analysis of Bohnenkamp et al. (2003)
ZeroConf protocol

- Protocol to allocate an address in a link-local network, without central authority (RFC 3927)
- We formalized the analysis of Bohnenkamp et al. (2003)
  - Address allocation when only one computer is added
ZeroConf protocol

- Protocol to allocate an address in a link-local network, without central authority (RFC 3927)
- We formalized the analysis of Bohnenkamp et al. (2003)
  - Address allocation when only one computer is added
  - Probability that two hosts end up with the same address
ZeroConf protocol

- Protocol to allocate an address in a link-local network, without central authority (RFC 3927)
- We formalized the analysis of Bohnenkamp et al. (2003)
  - Address allocation when only one computer is added
  - Probability that two hosts end up with the same address
  - Expected time until an address is allocated
ZeroConf protocol

- Protocol to allocate an address in a link-local network, without central authority (RFC 3927)
- We formalized the analysis of Bohnenkamp et al. (2003)
  - Address allocation when only one computer is added
  - Probability that two hosts end up with the same address
  - Expected time until an address is allocated
- Model checking analysis of Kwiatkowska et al. (2006) and Andova et al. (2003)
\[ S \to 1 - q \]

\[ \text{Ok} \xrightarrow{1} \]

\[ E \]
\[ q \rightarrow S \rightarrow P0 \]

\[ 1 - q \rightarrow Ok \]

\[ 1 \rightarrow Ok \]

\[ p_1 \rightarrow 0 \]

\[ p_0 \rightarrow 0 \]

\[ (N + 1) \]

\[ 8/24 \]
\[ q \cdot (N + 1) \cdot \frac{1}{2} \]

\[ 1 - q \]

\[ 1 - p \]

\[ 1 \]
\[ S \xrightarrow{q} P_0 \xrightarrow{p} P_1 \xrightarrow{1-p} P_{N-1} \xrightarrow{1-p} \ldots \xrightarrow{1-p} P_{N} \xrightarrow{1-q} \text{Ok} \]

\[ 1 - p \]

\[ 1 - p \]
\[ S \xrightarrow{q} P_0 \xrightarrow{p} P_1 \xrightarrow{\cdots} P_N \xrightarrow{p} Err \]

\[ S \xrightarrow{1-p} Ok \xrightarrow{1} S \]

\[ Ok \xrightarrow{1-q} S \xrightarrow{1} Ok \]

\[ q \cdot (N+1) \]

\[ r \cdot (N+1) \]

\[ 1 \]
\[ 1 - q; r \cdot (N + 1) \]
Fix parameters:

\[
\text{fixes } N :: \mathbb{N} \text{ and } p \ q \ r \ E :: \mathbb{R}
\]
Fix parameters:

\[
\text{fixes } N :: \mathbb{N} \text{ and } p \ q \ r \ E :: \mathbb{R}
\]
Fix parameters:

fixes $N :: \mathbb{N}$ and $p \quad q \quad r \quad E :: \mathbb{R}$

assumes $0 < p$ and $p < 1$ and $0 < q$ and $q < 1$
Fix parameters:

- **fixes** $N :\mathbb{N}$ and $p$, $q$, $r$, $E :\mathbb{R}$
- **assumes** $0 < p$ and $p < 1$ and $0 < q$ and $q < 1$
- **assumes** $0 \leq E$ and $0 \leq r$
Fix parameters:

\[
\text{fixes } N :: \mathbb{N} \text{ and } p \ q \ r \ E :: \mathbb{R} \\
\text{assumes } 0 < p \text{ and } p < 1 \text{ and } 0 < q \text{ and } q < 1 \\
\text{assumes } 0 \leq E \text{ and } 0 \leq r
\]

Define state space:

```plaintext
datatype zc-state = S | P \ N | Ok | Err

\[ \Omega = \{ S, Ok, Err \} \cup \{ P \ n \mid n \leq N \} \]
```
Fix parameters:

```plaintext
fixes \( N \in \mathbb{N} \) and \( p, q, r, E \in \mathbb{R} \)
assumes \( 0 < p \) and \( p < 1 \) and \( 0 < q \) and \( q < 1 \)
assumes \( 0 \leq E \) and \( 0 \leq r \)
```

Define state space:

```plaintext
datatype \( zc\text{-}state = S | P \mathbb{N} | Ok | Err \)
\[
\Omega = \{ S, Ok, Err \} \cup \{ P \ n \mid n \leq N \}
\]
```

Define the transition function \( \tau \):

```plaintext
\begin{align*}
\tau \ S \ Ok &= 1 - q \\
\tau \ S \ (P \ 0) &= q \\
\tau \ (P \ n) \ (P \ (n + 1)) &= \text{if } n < N \text{ then } p \text{ else } 0 \\
\end{align*}
```
 Defines a Markov chain:

\[ \text{lemma} \quad \text{markov-chain} \ \Omega \ \tau \]
Defines a Markov chain:

```plaintext
lemma markov-chain Ω τ
```

Probability theory gives us:

\[ Pr_s(\omega. P \omega) \] – the probability that a trace \( \omega \) fulfills \( P \omega \)
Defines a Markov chain:

\[ \text{lemma } \text{markov-chain } \Omega \tau \]

Probability theory gives us:

\[ \Pr_s(\omega. P \omega) \] – the probability that a trace \( \omega \) fulfills \( P \omega \)

Define probability that an error is reached:

\[ P_{err} s = \Pr_s(\omega. \exists n. \omega n = Err) \]
- Defines a Markov chain:

```
lemma markov-chain Ω τ
```

- Probability theory gives us:

\[ \Pr_s(\omega. P \omega) \] – the probability that a trace \( \omega \) fulfills \( P \omega \)

- Define probability that an error is reached:

\[ P_{err} s = \Pr_s(\omega. \exists n. \omega n = Err) \]

- Analyse: \( P_{err} S = ? \)
lemma

\[ n \leq N \implies \text{P}_{\text{err}} (P(N - n)) = p^{n+1} + (1 - p^{n+1}) \cdot \text{P}_{\text{err}} S \]
lemma
\[ n \leq N \implies P_{\text{err}}(P(N - n)) = p^{n+1} + (1 - p^{n+1}) \cdot P_{\text{err}} \]

proof (induct \( n \))
lemma
\[ n \leq N \implies \text{P}_{\text{err}} (P(N - n)) = p^{n+1} + (1 - p^{n+1}) \cdot \text{P}_{\text{err}} \]

proof (induct n)

case (\( n + 1 \))
lemma
\[ n \leq N \implies \text{Perr} (P(N - n)) = p^{n+1} + (1 - p^{n+1}) \cdot \text{Perr} \ S \]

proof (induct \( n \))

case (\( n + 1 \))

have \( \text{Perr} (P(N - (n + 1))) \)
\[ = p \cdot (p^{n+1} + (1 - p^{n+1}) \cdot \text{Perr} \ S) + (1 - p) \cdot \text{Perr} \ S \]

by (simp\cdots)
lemma
\[ n \leq N \implies P_{\text{err}} (P(N - n)) = p^{n+1} + (1 - p^{n+1}) \cdot P_{\text{err}} S \]
proof (induct \( n \))
  case \( (n+1) \)
  have \( P_{\text{err}} (P(N - (n+1))) \)
  \[
  = p \cdot (p^{n+1} + (1 - p^{n+1}) \cdot P_{\text{err}} S) + (1 - p) \cdot P_{\text{err}} S 
  \]
  by \( \text{simp} \cdots \)
  also have \( \ldots = p^{(n+1)+1} + (1 - p^{(n+1)+1}) \cdot P_{\text{err}} S \)
  by \( \text{simp} \cdots \)
lemma
\[ n \leq N \implies P_{\text{err}} (P(N - n)) = p^{n+1} + (1 - p^{n+1}) \cdot P_{\text{err}} S \]

proof (induct n)

\begin{align*}
\text{case } (n + 1) & \\
\text{have } P_{\text{err}} (P(N - (n + 1))) & \\
& = p \cdot (p^{n+1} + (1 - p^{n+1}) \cdot P_{\text{err}} S) + (1 - p) \cdot P_{\text{err}} S \\
& \quad \text{by (simp\ldots)} \\
\text{also have } ... & = p^{(n+1)+1} + (1 - p^{(n+1)+1}) \cdot P_{\text{err}} S \\
& \quad \text{by (simp\ldots)} \\
\text{finally show } P_{\text{err}} (P(N - (n + 1))) & \\
& = p^{(n+1)+1} + (1 - p^{(n+1)+1}) \cdot P_{\text{err}} S . \\
\end{align*}

next
lemma
\[ n \leq N \implies P_{\text{err}}(P(N - n)) = p^{n+1} + (1 - p^{n+1}) \cdot P_{\text{err}} \]

proof (induct n)

\textbf{case } (n + 1)

have \[ P_{\text{err}}(P(N - (n + 1))) = p \cdot (p^{n+1} + (1 - p^{n+1}) \cdot P_{\text{err}} \cdot S) + (1 - p) \cdot P_{\text{err}} \]

\textbf{by } (simp\cdots)

\textbf{also have } ... = p^{(n+1)+1} + (1 - p^{(n+1)+1}) \cdot P_{\text{err}} \cdot S

\textbf{by } (simp\cdots)

\textbf{finally show } P_{\text{err}}(P(N - (n + 1)))

\[ = p^{(n+1)+1} + (1 - p^{(n+1)+1}) \cdot P_{\text{err}} \cdot S \]

\textbf{next}

\textbf{case } 0

\textbf{show } P_{\text{err}}(P(N - 0)) = p^{0+1} + (1 - p^{0+1}) \cdot P_{\text{err}} \cdot S

\textbf{by } simp
lemma

\[ n \leq N \implies P_{\text{err}}(P(N - n)) = p^{n+1} + (1 - p^{n+1}) \cdot P_{\text{err}} \]

proof (induct n)

case \((n + 1)\)

have \(P_{\text{err}}(P(N - (n + 1)))\)

\[ = p \cdot (p^{n+1} + (1 - p^{n+1}) \cdot P_{\text{err}} \cdot S) + (1 - p) \cdot P_{\text{err}} \cdot S \]

by (simp⋅⋅⋅)

also have \(\ldots = p^{(n+1)+1} + (1 - p^{(n+1)+1}) \cdot P_{\text{err}} \cdot S\)

by (simp⋅⋅⋅)

finally show \(P_{\text{err}}(P(N - (n + 1)))\)

\[ = p^{(n+1)+1} + (1 - p^{(n+1)+1}) \cdot P_{\text{err}} \cdot S \cdot .\]

next

case 0

show \(P_{\text{err}}(P(N - 0)) = p^{0+1} + (1 - p^{0+1}) \cdot P_{\text{err}} \cdot S\)

by simp

qed
General result:

\[ P_{\text{err}}(S) = \frac{q \cdot p^{N+1}}{1 - q \cdot (1 - p^{N+1})} \]
General result:

\[ P_{\text{err}} \ S = \frac{q \cdot p^{N+1}}{1 - q \cdot (1 - p^{N+1})} \]

16 hosts \( (q = 16/65024) \), 3 probe runs \( (N = 2) \), \( p = 0.01 \):

Corollary \( P_{\text{err}} \ S \leq 10^{-13} \)
How do we model the expected running time?

...
How do we model the expected running time?

Similar to $\tau$ define the cost function $\rho$:

- $\rho \ S \ Ok = r \cdot (N + 1)$
- $\rho \ S \ (P0) = r$
- $\rho \ (P\ n) \ (P(n+1)) = \begin{cases} r & \text{if } n < N \\ 0 & \text{otherwise} \end{cases}$
How do we model the expected running time?

Similar to $\tau$ define the cost function $\rho$:

\[
\begin{align*}
\rho \ S \ \text{Ok} &= r \cdot (N+1) \\
\rho \ S \ (P0) &= r \\
\rho \ (P \ n) \ (P(n+1)) &= \text{if } n < N \text{ then } r \text{ else } 0 \\
\end{align*}
\]

Define expected cost until \textit{Err} or \textit{Ok} is reached:

\[
C_{\text{fin}} \ s = \int_{\omega} \text{cost-until} \ \{\text{Err, Ok}\} \ (s \cdot \omega) \ dPr \ s
\]
How do we model the expected running time?

Similar to $\tau$ define the cost function $\rho$:

$$\rho \ S \ Ok = r \cdot (N+1)$$
$$\rho \ S \ (P0) = r$$
$$\rho \ (Pn) \ (P(n+1)) = \begin{cases} r & \text{if } n < N \text{ then } r \text{ else } 0 \\ \vdots & \end{cases}$$

Define expected cost until $Err$ or $Ok$ is reached:

$$C_{\text{fin}} \ S = \int_{\omega} \text{cost-until} \ \{Err, Ok\} \ (s \cdot \omega) \ d\Pr_s$$

16 hosts, 3 probe runs, $p = 0.01$, $r = 2ms$, $E = 3600s$:

**Theorem** $C_{\text{fin}} \ S \leq 0.007$
Case study: Crowds protocol
Crowds protocol

- Anonymizing protocol introduced and analysed by Reiter & Rubin (1998)
Crowds protocol

- Anonymizing protocol introduced and analysed by Reiter & Rubin (1998)
- Group of nodes establishes a connection by randomly choosing another node or the final server
Crowds protocol

- Anonymizing protocol introduced and analysed by Reiter & Rubin (1998)
- Group of nodes establishes a connection by randomly choosing another node or the final server
- Analysis:
Crowds protocol

- Anonymizing protocol introduced and analysed by Reiter & Rubin (1998)
- Group of \textit{nodes} establishes a connection by randomly choosing another node or the final server
- Analysis:
  - Probability that original sender contacts a collaborating node is small
Crowds protocol

- Anonymizing protocol introduced and analysed by Reiter & Rubin (1998)
- Group of *nodes* establishes a connection by randomly choosing another node or the final server
- Analysis:
  - Probability that original sender contacts a collaborating node is small
  - Information a contacted collaborating node gains is small
Probabilities:
Probabilities:

\[ \frac{1}{3}, \frac{2}{3} \]
Probabilities:

\[ \frac{1}{3} \]
Probabilities:

\[ P_1 = \frac{1}{3} \]

\[ P_2 = \frac{pf}{3} \]
Probabilities:

\[ \frac{1}{3} \]

\[ \frac{pf}{3} \]

\[ 1 - pf \]
Fix parameters

\textbf{Fixes}\ $N$, $C$: node set and $p_f \in \mathbb{R}$ and $p_i : \text{node} \rightarrow \mathbb{R}$
- **Fix parameters**

\[
\text{fixes } N, C :: \text{node set and } p_f :: \mathbb{R} \text{ and } p_i :: \text{node} \rightarrow \mathbb{R}
\]
Fix parameters

\( \text{fixes } N \ C \text{:: node set and } p_f :: \mathbb{R} \text{ and } p_i :: \text{node} \rightarrow \mathbb{R} \)

assumes \( 0 < p_f \text{ and } p_f < 1 \)
Fix parameters

**Fixes** $N, C :: \text{node set and } p_f :: \mathbb{R} \text{ and } p_i :: \text{node} \rightarrow \mathbb{R}$

**Assumes** $0 < p_f \text{ and } p_f < 1$

**Assumes** $N \neq \emptyset \text{ and finite } N$
Fix parameters

fixes $N, C :: \text{node set and } p_f :: \mathbb{R}$ and $p_i :: \text{node} \to \mathbb{R}$
assumes $0 < p_f$ and $p_f < 1$
assumes $N \neq \emptyset$ and finite $N$
assumes $\forall n \in N. 0 \leq p_i \ n$ and $\sum_{n \in N} p_i \ j = 1$
Fix parameters

\begin{itemize}
\item fixes $N \ C :: node \ set \ and \ p_f :: \mathbb{R} \ and \ p_i :: node \rightarrow \mathbb{R}$
\item assumes $0 < p_f \ and \ p_f < 1$
\item assumes $N \neq \emptyset \ and \ finite \ N$
\item assumes $\forall n \in N. \ 0 \leq p_i n \ and \ \sum_{n \in N} p_i j = 1$
\item assumes $C \neq \emptyset \ and \ C \subset N \ and \ \forall c \in C. \ p_i c = 0$
\end{itemize}
Fix parameters

\textbf{fixes} \ N \ C :: \ \text{node set and} \ \ p_f :: \ \mathbb{R} \ \text{and} \ \ p_i :: \ \text{node} \to \ \mathbb{R}

\textbf{assumes} \ 0 < p_f \ \text{and} \ p_f < 1

\textbf{assumes} \ N \neq \emptyset \ \text{and} \ \text{finite} \ N

\textbf{assumes} \ \forall n \in N. \ 0 \leq p_i \ n \ \text{and} \ \sum_{n \in N} p_i \ j = 1

\textbf{assumes} \ C \neq \emptyset \ \text{and} \ C \subset N \ \text{and} \ \forall c \in C. \ p_i \ c = 0

Define state space

\textbf{datatype} \ \alpha \ c\text{-state} = S | I \alpha | M \alpha | E

\Omega = \{S\} \cup \{I \ n \ | \ n \in N \ setminus C\} \cup \{M \ n \ | \ n \in N\} \cup \{E\}
Define transition function

\[ \tau_S (I_n) = p_i \ n \]
\[ \tau (I_n) (M_n') = 1/|N| \]
\[ \tau (M_n) (M_n') = p_f /|N| \]
\[ \tau (M_n) E = 1 - p_f \]
\[ \tau E E = 1 \]
\[ \tau _ _ = 0 \]
Define transition function

\[
\begin{align*}
\tau S (I \ n) &= p_i \ n \\
\tau (I \ n) (M \ n') &= 1/|N| \\
\tau (M \ n) (M \ n') &= p_f /|N| \\
\tau (M \ n) E &= 1 - p_f \\
\tau E E &= 1 \\
\tau \_ \_ &= 0
\end{align*}
\]

Prove Markov chain property

\textbf{theorem} \quad \textit{markov-chain} \ \Omega \ \tau
We introduce some random variables:

- the initiating node
- last-ncoll
- the first node contacting a collaborating node
- hit
  - true if a collaborating node is contacted

Probability that initiating node contacts a collaborating node:

\[ \Pr(S(\omega \mid init \omega = last-ncoll \omega \mid hit) = 1 - \frac{|N \setminus C| - 1}{|N|} \cdot p_f) \]

Information the collaborating nodes gain when contacted:

\[ I_{hit}(\text{init}; last-ncoll) \leq \left(1 - \frac{|N \setminus C| - 1}{|N|} \cdot p_f \right) \cdot \log_2 \frac{|N \setminus C|}{2^{1/2}} \]
We introduce some random variables:

- the initiating node
- last-ncoll
- hit

- Probability that initiating node contacts a collaborating node

\[ \Pr(S(\omega \mid \text{init} \omega = \text{last-ncoll} \mid \text{hit}) = 1 - |N \setminus C| - 1 \cdot p_f) \]

- Information the collaborating nodes gain when contacted

\[ I_{\text{hit}}(\text{init} \mid \text{last-ncoll}) \leq (1 - |N \setminus C| - 1 \cdot p_f) \cdot \log_2 |N \setminus C| \cdot 2^{1/24} \]
We introduce some random variables:

- $init$: the initiating node
We introduce some random variables:

- $init$ the initiating node
- $last-ncoll$ the first node contacting a collaborating node

Probability that initiating node contacts a collaborating node:

$$Pr(S(\omega) \mid init(\omega) = last-ncoll(\omega)) = 1 - \frac{|N | - |N \setminus C| - 1 \cdot |N|}{\cdot p_f}$$

Information the collaborating nodes gain when contacted:

$$I(hit(\omega); init(\omega), last-ncoll(\omega)) \leq (1 - \frac{|N | - |N \setminus C| - 1 \cdot |N|}{\cdot p_f}) \cdot \log_2 |N \setminus C|^{21/24}$$
We introduce some random variables:

- **init** the initiating node
- **last-ncoll** the first node contacting a collaborating node
- **hit** true if a collaborating node is contacted
We introduce some random variables:

- **init** the initiating node
- **last-ncoll** the first node contacting a collaborating node
- **hit** true if a collaborating node is contacted

**Theorem**

\[
\Pr_S(\omega. \text{init } \omega = \text{last-ncoll } \omega \mid \text{hit } \omega) = 1 - \frac{|N \setminus C| - 1}{|N|} \cdot p_f
\]
- We introduce some random variables:
  - \textit{init} the initiating node
  - \textit{last-ncoll} the first node contacting a collaborating node
  - \textit{hit} true if a collaborating node is contacted
- Probability that initiating node contacts a collaborating node

  \textbf{theorem} \quad \Pr_S(\omega. \text{init } \omega = \text{last-ncoll } \omega | \text{hit } \omega) = 1 - \frac{|M \setminus C| - 1}{|N|} \cdot p_f

- Information the collaborating nodes gain when contacted

  \textbf{theorem} \quad I_{hit}(\text{init}; \text{last-ncoll}) \leq \left(1 - \frac{|M \setminus C| - 1}{|N|} \cdot p_f \right) \cdot \log_2 |N \setminus C|
Related Work: probability theory in ITPs

- Probability space of boolean sequences: $\mathbb{N} \rightarrow \{0, 1\}$
  Hurd (2002), Hasan et al. (2009), Liu et al. (2011)
Related Work: probability theory in ITPs

- Probability space of boolean sequences: \( \mathbb{N} \rightarrow \{0,1\} \)
  Hurd (2002), Hasan et al. (2009), Liu et al. (2011)

- Expectation and information theory (discrete, finite spaces)
  Coble (2009)
Related Work: probability theory in ITPs

- Probability space of boolean sequences: $\mathbb{N} \rightarrow \{0, 1\}$
  Hurd (2002), Hasan et al. (2009), Liu et al. (2011)

- Expectation and information theory (discrete, finite spaces)
  Coble (2009)

- Formalization of pGCL (prob. & non-det. language)
  Hurd et al. (2005), Audebaud & Paulin-Mohring (2009)
Summary & Future Work

- Markov chains with probability, expectation, and information

Future Work:

- More Markov models (MDPs, CTMCs, CTMDPs, PTAs)
- Certification of probabilistic model checker results
- Specification language

Slides available at: http://www.in.tum.de/~hoelzl
Summary & Future Work

- Markov chains with probability, expectation, and information
- ZeroConf protocol: a few days; \( \approx 300 \) lines of theory
Summary & Future Work

- Markov chains with probability, expectation, and information
- ZeroConf protocol: a few days; ≈ 300 lines of theory
- Crowds anonymity: a few weeks; ≈ 1,100 lines of theory

Future Work:
- More Markov models (MDPs, CTMCs, CTMDPs, PTAs)
- Certification of probabilistic model checker results
- Specification language

Slides available at: http://www.in.tum.de/~hoelzl
Summary & Future Work

- Markov chains with probability, expectation, and information
- ZeroConf protocol: a few days; ≈ 300 lines of theory
- Crowds anonymity: a few weeks; ≈ 1,100 lines of theory
- Compare: ≈ 20,600 lines of theory for probability theory

Slides available at: http://www.in.tum.de/~hoelzl
Summary & Future Work

- Markov chains with probability, expectation, and information
- ZeroConf protocol: a few days; ≈ 300 lines of theory
- Crowds anonymity: a few weeks; ≈ 1,100 lines of theory
- Compare: ≈ 20,600 lines of theory for probability theory

Future Work:

- More Markov models (MDPs, CTMCs, CTMDPs, PTAs)
- Certification of probabilistic model checker results
- Specification language

Slides available at: http://www.in.tum.de/~hoelzl
Summary & Future Work

- Markov chains with probability, expectation, and information
- ZeroConf protocol: a few days; \( \approx 300 \) lines of theory
- Crowds anonymity: a few weeks; \( \approx 1,100 \) lines of theory
- Compare: \( \approx 20,600 \) lines of theory for probability theory

Future Work:

- More Markov models (MDPs, CTMCs, CTMDPs, PTAs)
- Certification of probabilistic model checker results
- Specification language

Slides available at: http://www.in.tum.de/~hoelzl