Interactive verification of Markov chains: Two distributed protocol case studies

Johannes Hölzl[*] Tobias Nipkow
Technische Universität München
http://www.in.tum.de/~hoelzl http://www.in.tum.de/~nipkow

Probabilistic model checkers like PRISM only check probabilistic systems of a fixed size. To guarantee the desired properties for an arbitrary size, mathematical analysis is necessary. We show for two case studies how this can be done in the interactive proof assistant Isabelle/HOL. The first case study is a detailed description of how we verified properties of the ZeroConf protocol, a decentral address allocation protocol. The second case study shows the more involved verification of anonymity properties of the Crowds protocol, an anonymizing protocol.

1 Introduction

The predominant approach to verification of probabilistic systems is model checking [4], and the most popular model checker is PRISM [17]. Model checking is automatic, but restricted to fixed finite models. In this paper we put forward interactive theorem proving as a realistic alternative approach that can deal with infinite-state systems on an abstract mathematical level of Markov chains. The specific contributions of this paper are two case studies that illustrate our approach: the ZeroConf protocol for decentralized address allocation and the anonymizing Crowds protocol. The verifications are carried out in the proof assistant Isabelle/HOL [21].

The characteristics of the theorem proving approach are:

- It can deal with infinite-state systems, although this paper considers only parameterized finite-state systems.
- It is not restricted to some fixed set of concepts but user-extensible.
- Logical soundness of the system depends only on the soundness of a small fixed and trustworthy kernel of the theorem prover.
- It requires familiarity with a theorem prover and a problem-dependent amount of work for each verification.

In a nutshell, it is mathematics, but checked by a computer. These characteristics indicate that the approach is more suitable for a research environment than a product development environment.

2 Formalization of probability in Isabelle/HOL

To reason about Markov chains, especially about the probability that a path is in a certain set, requires measure and probability theory. This section gives a short introduction into the formalization of the theories required by this paper. For a more detailed overview of the measure space formalization see Hölzl and Heller [11], and for the formalization of Markov chains see Hölzl and Nipkow [12].

[*]Supported by the DFG Graduiertenkolleg 1480 (PUMA) and DFG project NI 491/10-2.

U. Fahrenberg, A. Legay and C. Thrane:
Quantities in Formal Methods (QFM 2012)
EPTCS 103, 2012, pp. 17–31, doi:10.4204/EPTCS.103.2

2.1 Isabelle/HOL notation

Isabelle/HOL largely follows ordinary mathematical notation. With a few exceptions, we follow Isabelle/HOL notation in this paper, to give the reader a better impression of the look-and-feel of the work. HOL is based on λ-calculus. Hence functions are usually curried ($\tau_1 \to \tau_2 \to \tau_3$ rather than $\tau_1 \times \tau_2 \to \tau_3$) and function application is written $f\ a$ rather than $f(a)$. The letters α and β stand for type variables. Type $\mathbb{B}$ is the type of boolean values. Type τ *set* is the type of sets with elements of type τ. Notation $t :: \tau$ means that t is a term of type τ. We regard functions of type $\mathbb{N} \to \tau$ as infinite sequences of elements of type τ. Prepending an element $a :: \tau$ to a sequence $\omega :: \mathbb{N} \to \tau$ is written $a \cdot \omega$ and means $\lambda i.\ \mathsf{if}\ i = 0\ \mathsf{then}\ a\ \mathsf{else}\ \omega\ (i-1)$. The term $\mathsf{LEAST}\ n.\ P\ n$ is the least natural number n such that $P\ n$ holds. If there is no such n, then the term has some arbitrary (defined!) value, but we do not know which.

2.2 Probability space

In this paper we are only interested in probabilities, hence we write measures as $\Pr_s :: \alpha\ \mathit{set} \to \mathbb{R}$, where s indicates the particular probability measure under consideration. Similarly for the measurable sets we write $\mathcal{A}_s :: \alpha\ \mathit{set}\ \mathit{set}$ and for the entire space we write $\Omega_s :: \alpha\ \mathit{set}$. Here α is an arbitrary type where we cut out a space Ω_s. This is necessary as in many cases we are only interested in a subset of the entire type, e.g. α is the type of natural numbers $\mathbb{N}$ and we want to have a distribution on the finite subset $\Omega_s = \{0,\ldots,N\}$. We usually drop Ω_s and write $\{\omega \mid P\ \omega\}$ instead of $\{\omega \in \Omega_s \mid P\ \omega\}$ and $\Pr_s(\omega.\ P\ \omega)$ instead of $Pr_s\ \{\omega \in \Omega_s.\ P\ \omega\}$.

The measurable sets $\mathcal{A}_s$ form a σ-algebra, hence they are closed under conjunction, disjunction, negation and countably bounded universal and existential quantification. We have the defining properties on the probability measure $\Pr_s$, as $\Pr_s\ \emptyset = 0$, $\Pr_s\ \Omega_s = 1$, it is non-negative: $0 \le \Pr_s\ A$ and countably additive: For a measurable and disjoint family $P :: \mathbb{N} \to \alpha \to \mathbb{B}$

$$\Pr_s(\omega.\ \exists i.\ P\ i\ \omega) = \left(\textstyle\sum_i \Pr_s(\omega.\ P\ i\ \omega)\right) .$$

For a finite probability space measurable sets need only be closed under finite bounded quantifiers, and the probability needs only be finitely additive, instead of countably additive. Unfortunately, the path space on Markov chains is neither finite nor discrete, so we need σ-algebras and countably additive probability measures.

We also need conditional probability and define it as usual:

$$\Pr_s(\omega.\ P\ \omega \mid Q\ \omega) = \Pr_s(\omega.\ P\ \omega \wedge Q\ \omega) / \Pr_s(\omega.\ Q\ \omega) .$$

The AE-quantifier $\mathrm{AE}_s\,\omega.\ P\ \omega$ on a path measure $\Pr_s$ states that the property P holds with probability 1. Isabelle/HOL also has a formalization of the Lebesgue integral on probability spaces, as notation we use $\int_\omega f\ \omega\ d\Pr_s$.

2.3 Markov chains

We introduce Markov chains as probabilistic automata, i.e. as discrete-time time-homogeneous finite-space Markov processes. A Markov chain is defined by its state space $S :: \alpha\ \mathit{set}$ and an associated transition matrix $\tau :: \alpha \to \alpha \to \mathbb{R}$. We assume no initial distribution or starting state, however when measuring paths we always provide a starting state. A path on a Markov chain is a function $\mathbb{N} \to S$, i.e. an infinite sequence of states visited in the Markov chain.

$$\textit{markov-chain}\ S\ \tau = \textit{finite}\ S \wedge S \neq \emptyset \wedge \Big(\forall s, s' \in S.\ 0 \leq \tau\ s\ s'\Big) \wedge \Big(\forall s \in S.\ \big(\sum_{s' \in S} \tau\ s\ s'\big) = 1\Big)$$

For the rest of this section we assume a Markov chain with state space S and transition matrix τ. We write $E(s)$ for the set of all successor states, i.e. all $s' \in S$ with $\tau\ s\ s' \neq 0$. Note that a path ω does not require that $\omega\ (i+1)$ is a successor of $\omega\ i$.

We have defined a probability space $(\mathbb{N} \to S, \mathcal{A}, \Pr_s)$ on the space of all paths $\mathbb{N} \to S$ for a starting state $s \in S$. The measurable sets are the σ-algebra generated by all sets $\{\omega \in \mathbb{N} \to S \mid \omega\ i = t\}$ where $i \in \mathbb{N}$ and $t \in S$. The measure $\Pr_s$ (depending on the starting state $s \in S$ and τ) is defined via an infinite product and is shown to satisfy the following key property (where $s \cdot \omega$ prepends s to ω):

$$\forall \omega \in \mathbb{N} \to S, s \in S, n.\ \Pr_s \{\omega' \mid \forall i < n.\ \omega'\ i = \omega\ i\} = \textstyle\prod_{i<n} \tau\ ((s \cdot \omega)\ i)\ (\omega\ i)$$

Note that $\Pr_s$ explicitly carries the starting state and yields the transition probability for the steps $s \to_\tau \omega\ 0 \to_\tau \omega\ 1 \to_\tau \cdots \to_\tau \omega\ (n-1)$.

We also use Markov reward chains, where we assign a cost or reward to each transitions:

$$\textit{markov-reward-chain}\ S\ \tau\ \rho = \textit{markov-chain}\ S\ \tau \wedge (\forall s, s' \in S.\ 0 \leq \rho\ s\ s')$$

This approach allows a very easy definition of a Markov chain given as a transition system. Other formalizations of Markov chains [13, 19] use the probability space $\mathbb{N} \to \mathbb{B}$. This requires to provide a measurable function $X\ t\ \omega$, mapping a sequence of boolean choices $\omega :: \mathbb{N} \to \mathbb{B}$ into a state at time t. In our approach the set of states S and the transition matrix τ are enough.

Some models require an arbitrary set I of independent variables X_i with distribution P_i. For this case we provide the product $\prod_I P_i$. We use this product space to construct the path space for our Markov chains. Furthermore the probability space $\mathbb{N} \to \mathbb{B}$ is just a special instance of the generalized product space.

2.3.1 Iterative equations

The Markov chain induces *iterative equations* on the probability $\Pr_s$, the Lebesgue integral and the AE-quantifier, relating properties about s to properties of $E(s)$. These equations are often useful in inductive proofs and already give a hint how to prove concrete properties of probabilities and integrals. If A, P, and f are measurable and $s \in S$, then the following equations hold:

$$\begin{aligned}
\Pr_s\ A &= \sum_{s' \in E(s)} \tau\ s\ s' * \Pr_{s'}(\omega.\ s' \cdot \omega \in A) \\
\int_\omega f\ \omega\ d\Pr_s &= \sum_{s' \in E(s)} \tau\ s\ s' * \int_\omega f\ (s' \cdot \omega)\ d\Pr_{s'} \\
\mathrm{AE}_s\, \omega.\ P\ \omega &= \forall s' \in E(s).\ \mathrm{AE}_{s'}\, \omega.\ P\ (s' \cdot \omega)
\end{aligned}$$

2.3.2 Reachability

Let Φ be a subset of S. A state s' is *reachable* via Φ starting in s iff there is a non-zero probability to reach s' by only going through the specific set of states Φ. The starting state s and the final state s' need not be in Φ.

$$\begin{aligned}
\textit{reachable}\ \Phi\ s := \{s' \in S \mid \exists \omega \in \Omega, n.\ &(\forall i \leq n.\ \omega\ i \in E((s \cdot \omega)\ i)) \wedge \\
&(\forall i < n.\ \omega\ i \in \Phi) \wedge \omega\ n = s'\}
\end{aligned}$$

Reachability is a purely qualitative property, as it is defined on the graph of non-zero transitions.

The until-operator introduces a similar concept on paths. Its definition does not assume that a state is a successor state of the previous one, as this is already ensured by $\Pr_s$.

$$\mathit{until}\ \Phi\ \Psi = \{\omega \mid \exists n.\ (\forall i < n.\ \omega\ i \in \Phi) \wedge \omega\ n \in \Psi\}$$

Can we compute $\Pr_s(\mathit{until}\ \Phi\ \Psi)$ using only *reachable*? It is easy to show that $\Pr_s(\mathit{until}\ \Phi\ \Psi) = 0$ iff $(\mathit{reachable}\ \Phi\ s) \cap \Psi = \emptyset$. But is there also a way to characterize $\Pr_s(\mathit{until}\ \Phi\ \Psi) = 1$ in terms of *reachable*?

2.3.3 Fairness

To show that *reachable* can be used to guarantee that states are reached with probability 1, we need state fairness. A path ω is *state fair* w.r.t. s and t if s appears only finitely often provided that t also appears only finitely often as the successor of s in ω. The definition and proofs about state fairness are based on the thesis by Baier [3].

$$\mathit{fair}\ s\ t = \big\{\omega \mid \mathit{finite}\ \{n \mid \omega\ n = s \wedge \omega\ (n+1) = t\} \implies \mathit{finite}\ \{n \mid \omega\ n = s\}\big\}$$

We show that almost every path is state fair for each state and its successors.

$$\forall s, s' \in S, t' \in E(s').\ \mathrm{AE}_s\, \omega.\ s{\cdot}\omega \in \mathit{fair}\ s'\ t'$$

Using this we prove that starting in a state s almost every path fulfills $\mathit{until}\ \Phi\ \Psi$ if (1) all states reachable via Φ are in Φ or Ψ and (2) each state reachable from s has the possibility to reach Ψ. This theorem allows us to prove that $\mathit{until}\ \Phi\ \Psi$ holds almost everywhere by a reachability analysis on the graph:

$$\begin{aligned}
& s \in \Phi\ \wedge\ \Phi \subseteq S\ \wedge\ \mathit{reachable}\ (\Phi \setminus \Psi)\ s \subseteq \Phi \cup \Psi\ \wedge \\
& \forall t \in (\mathit{reachable}\ (\Phi \setminus \Psi)\ s \cup \{s\}) \setminus \Psi.\ \mathit{reachable}\ (\Phi \setminus \Psi)\ t \cap \Psi \neq \emptyset \\
& \implies \mathrm{AE}_s\, \omega.\ s{\cdot}\omega \in \mathit{until}\ \Phi\ \Psi
\end{aligned}$$

2.3.4 Hitting time

The *hitting time* on a path ω is the first index at which a state from a set Φ occurs:

$$\mathit{hitting\text{-}time}\ \Phi\ \omega = \mathsf{LEAST}\ i.\ \omega\ i \in \Phi$$

Note that if there is no i such that $\omega\ i \in \Phi$, then $\mathit{hitting\text{-}time}\ \Phi\ \omega$ is some arbitrary, underspecified natural number. For the computation of rewards it is important to know if the expected hitting time is finite. We show that the expected hitting time of Φ for paths starting in s is finite if almost every path starting in s reaches Φ. If s is in S and $\mathrm{AE}_s\, \omega.\ s{\cdot}\omega \in \mathit{until}\ S\ \Phi$ then

$$\int_\omega \mathit{hitting\text{-}time}\ \Phi\ (s{\cdot}\omega) d\Pr{}_s \neq \infty$$

For Markov reward chains we are interested in the transition costs until a set of states occurs:

$$\mathit{cost\text{-}until}\ \Phi\ \omega = \mathsf{if}\ \exists i.\ \omega\ i \in \Phi\ \mathsf{then}\ \textstyle\sum_{i < \mathit{hitting\text{-}time}\ \Phi\ \omega} \rho\ (\omega\ i)\ (\omega\ (i+1))\ \mathsf{else}\ \infty$$

3 Case study: The ZeroConf protocol

Ad-hoc networks usually do not have a central address authority assigning addresses to new nodes in the network. An example are consumer networks where users want to connect their laptops to exchange data or attach a network capable printer. When connecting with WiFi these devices use IPv4 and hence need IPv4 addresses to communicate with each other.

The *ZeroConf* protocol [6] is a distributed network protocol which allows new hosts in the network to allocate an unused link-local IPv4 address. A link-local address is only valid in the local network, e.g. a WiFi network. We assume point-to-point communication in our local network, and hence communicate directly with each host identified by a valid address. The problem with IPv4 addresses is that they are limited, i.e. they are represented by 32-bit numbers, and for the local network the addresses from 169.254.1.0 to 169.254.254.255 are available, hence we can chose from 65024 distinct addresses. ZeroConf works by randomly selecting an address from this pool and then probing if the address is already in use.

Bohnenkamp *et al.* [5] give a formal analysis of the probability that an address collision happens, i.e. two hosts end up with the same address. They also analyse the expected run time until a (not necessaryly valid) address is chosen. As our first case study we formalize their analysis in Isabelle/HOL.

Andova *et al.* [1] present a model-checking approach for discrete-time Markov reward chains and apply it to the ZeroConf protocol as a case study. They support multiple reward structures and can compute the probability based on multiple constraints on these reward structures. Kwiatkowska *et al.* [18] have modelled this protocol as a probabilistic timed automata in PRISM. Both models include more features of the actual protocol than the model by Bohnenkamp *et al.* [5] that we follow.

3.1 Description of address allocation

We give a short description of the model used in Bohnenkamp *et al.* [5]. The address allocation in ZeroConf uses ARP (address resolution protocol) to detect if an address is in use or not. An ARP request is sent to detect if a specific IPv4 address is already in use. When a host has the requested IPv4 address it answers with an ARP response. ZeroConf allocates a new address as follows:

1. Select uniformly a random address in the range 169.254.1.0 to 169.254.254.255.
2. Send an ARP request to detect if the address is already in use.
3. When a host responds to the ARP request, the address is already taken and we need to start again (go back to 1).
4. When no response arrives before a time limit r, we again send an ARP request. This is repeated N times.
5. When no response arrived for N requests we assume our address is not in use and are finished.

This probabilistic process depends on two parameters: (1) The probability q that the random chosen address is already taken; this probability depends on the number of hosts in the network and the number of available addresses. (2) The probability p that either the ARP request or response is lost.

The Markov chain shown in Fig. 1 describes the address allocation from a global viewpoint. At *Start* a new host is added to the network, it chooses an address and sends the first ARP request. There are two alternatives.

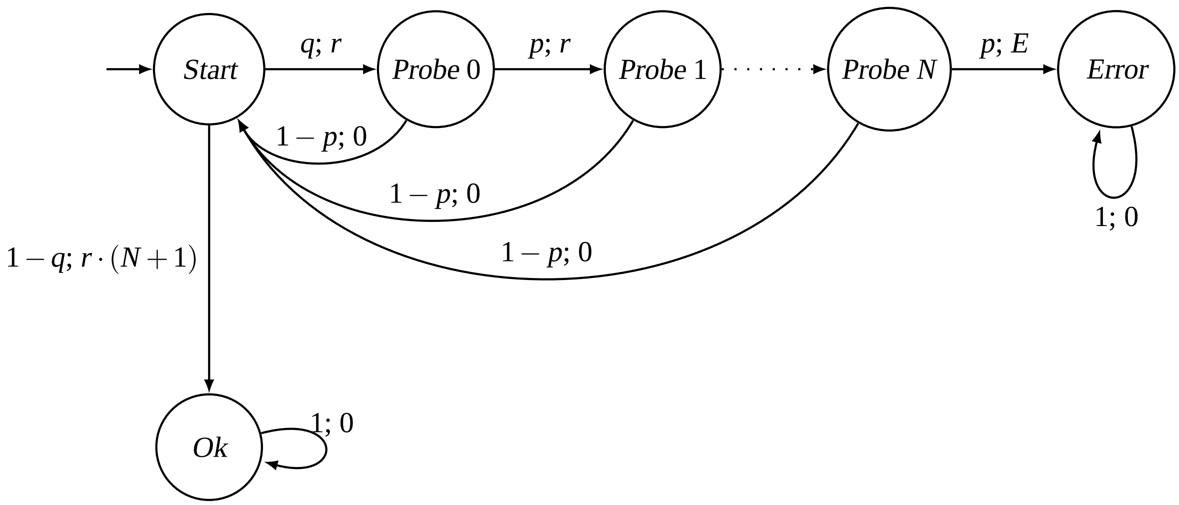

Figure 1: Markov chain of the ZeroConf protocol. The labels are annotated with $P;T$: the probability P to take this edge and the elapsed time T.

- With probability $1-q$ the host chooses an unused address, the allocation is finished, the Markov chain directly goes to Ok. Of course, the host does not know this, and still sends out $N+1$ ARP probes. Hence we associate the time cost $r \cdot (N+1)$ with this transition.
- With probability q the host chooses a used address and goes to the probing phase: In the *Probe n* state it sends an ARP request and waits until r time units have passed, or until it receives an ARP response from the address owner. With probability $1-p$ the host receives an ARP response and needs to choose a new address—we go back to $Start$. With probability p this exchange fails and we go to the next probe phase. After $N+1$ probes, the host assumes the chosen address is free. As two hosts in the network end up with the same address we reached the *Error* state. The time cost E models the cost to repair the double allocation. This might involve restarting a laptop.

3.2 Formal model of ZeroConf address allocation

The Isabelle/HOL model of the ZeroConf protocol describes the Markov chain in Fig. 1. We set up a context containing the probe numbers (starting with 0), the probabilities p and q, and the costs r and E:

$$\textsf{fixes}\ N :: \mathbb{N}\ \textsf{and}\ p\ q\ r\ E :: \mathbb{R}$$
$$\textsf{assumes}\ 0 < p\ \textsf{and}\ p < 1\ \textsf{and}\ 0 < q\ \textsf{and}\ q < 1$$
$$\textsf{assumes}\ 0 \le E\ \textsf{and}\ 0 \le r$$

In the following sections we assume that these fixed variables N, p, q, r, and E fulfill the above assumptions of the ZeroConf protocol.

To represent the states in the Markov chain we introduce a new datatype:

$$\textsf{datatype}\ \textit{zc-state} = \textit{Start} \mid \textit{Probe}\ \mathbb{N} \mid \textit{Ok} \mid \textit{Error}$$

We have the type *zc-state* with the distinct objects *Start*, *Ok*, *Error*, and *Probe n* for all $n :: \mathbb{N}$. The valid states $S ::$ *zc-state set* are a restriction of this to only valid probe numbers. This also gives us a finite number of states.

$$S = \{Start, Ok, Error\} \cup \{Probe\ n \mid n \leq N\}$$

The final modeling step is to define the transition matrix $\tau :: zc\text{-}state \to zc\text{-}state \to \mathbb{R}$ and the cost function $\rho :: zc\text{-}state \to zc\text{-}state \to \mathbb{R}$. Both are defined by a case distinction on the current state and return the zero function $\mathbf{0}$ which is updated at the states with non-zero transition probability or cost.

$$\begin{aligned}
\tau\ s = \mathsf{case}\ s\ \mathsf{of}\ & Start \Rightarrow \mathbf{0}(Probe\ 0 := q, Ok := 1 - q) \\
\mid\ & Probe\ n \Rightarrow \mathsf{if}\ n < N\ \mathsf{then}\ \mathbf{0}(Probe\ (n+1) := p, Start := 1 - p) \\
& \qquad\qquad\qquad\quad \mathsf{else}\ \ \mathbf{0}(Error := p, Start := 1 - p) \\
\mid\ & Ok \Rightarrow \mathbf{0}(Ok := 1) \\
\mid\ & Error \Rightarrow \mathbf{0}(Error := 1)
\end{aligned}$$

$$\begin{aligned}
\rho\ s = \mathsf{case}\ s\ \mathsf{of}\ & Start \Rightarrow \mathbf{0}(Probe\ 0 := r, Ok := r * (N+1)) \\
\mid\ & Probe\ n \Rightarrow \mathsf{if}\ n < N\ \mathsf{then}\ \mathbf{0}(Probe\ (n+1) := r)\ \mathsf{else}\ \mathbf{0}(Error := E) \\
\mid\ & Ok \Rightarrow \mathbf{0} \\
\mid\ & Error \Rightarrow \mathbf{0}
\end{aligned}$$

We need to prove that we actually defined a Markov chain: as a consequence, Isabelle/HOL is able to provide the probabilities $\Pr_s A$ for each state s and path set A. For this we show that τ is a valid transition matrix for a Markov chain on S, and ρ is a valid cost function:

theorem τ-DTMC: $\quad$ *markov-reward-chain* $S\ \tau\ \rho$

To prove this we need to show that τ and ρ are non-negative for all states in S. And finally we need to show that $\tau\ s$ is a distribution for all s in S, which is easy to show by using the helper lemma S-split:

lemma S-split: $\quad \sum_{s \in S} f\ s = f\ Start + f\ Ok + f\ Error + \sum_{n \leq N} f\ (Probe\ n)$

3.3 Probability of an erroneous allocation

The correctness property we want to verify is that no collision happens, i.e. we want to compute the probability that a protocol run ends in the *Error* state. The goal of this section is not only to show *what* we proved, but to show *how* we proved it. Most of the proofs are automatic by rewriting and we do not show the details. But we want to show the necessary lemmas and theorems needed to convince Isabelle/HOL.

We define $P_{err} :: zc\text{-}state \to \mathbb{R}$ to reason about the probability that a trace ω ends in the *Error* state when we started in a state s:

$$P_{err}\ s = \Pr_s\big(\omega.\ s{\cdot}\omega \in until\ S\ \{Error\}\big)$$

Our final theorem will be to characterize $P_{err}\ Start$ only in terms of the system parameters p, q and N.

The first obvious result is that when we are already in *Error*, we will stay in *Error*, and when we are in *Ok* we will never reach *Error*:

lemma P_{err}-error: $\quad P_{err}\ Error = 1$
lemma P_{err}-ok: $\quad P_{err}\ Ok = 0$

P_{err}-error is proved by rewriting: $\mathit{Error} \cdot \omega \in \mathit{until}\ S\ \{\mathit{Error}\}$ is always true. The *Ok* case is proved by $\mathit{reachable}\ (S \setminus \{\mathit{Error}\})\ \mathit{Ok} \subseteq \{\mathit{Ok}\}$. Together with lemma S-split and these two lemmas we provide an iterative lemma for P_{err}:

lemma P_{err}-iter:
$$s \in S \implies P_{err}\ s = \tau\ s\ \mathit{Start} * P_{err}\ \mathit{Start} + \tau\ s\ \mathit{Error} + \textstyle\sum_{n \leq N} \tau\ s\ (\mathit{Probe}\ n) * P_{err}\ (\mathit{Probe}\ n)$$

However this is a bad rewrite theorem, using it would result in non-termination of the rewrite engine. To avoid this we derive rules for specific states:

lemma P_{err}-last-probe: $P_{err}\ (\mathit{Probe}\ N) = p + (1 - p) * P_{err}\ \mathit{Start}$
lemma P_{err}-start-iter: $P_{err}\ \mathit{Start} = q * P_{err}\ (\mathit{Probe}\ 0)$

Our next step is to compute the probability to reach *Error* when we are in *Probe n*. This is the only proof which is not done by a simple rewrite step, but it requires induction and two separate rewrite steps. The induction is done over the number n of steps until we are in *Error*. To give the reader a better feeling for what these proofs look like, here is the skeleton of the Isabelle proof:

lemma P_{err}-probe-iter: $n \leq N \implies P_{err}\ (\mathit{Probe}\ (N - n)) = p^{n+1} + (1 - p^{n+1}) * P_{err}\ \mathit{Start}$
proof $(\mathit{induct}\ n)$
  case $(n+1)$
  have $P_{err}\ (\mathit{Probe}\ (N - (n+1))) = p * (p^{n+1} + (1 - p^{n+1}) * P_{err}\ \mathit{Start}) + (1 - p) * P_{err}\ \mathit{Start}$
    <proof>
  also have $\cdots = p^{(n+1)+1} + (1 - p^{(n+1)+1}) * P_{err}\ \mathit{Start}$
    <proof>
  finally show $P_{err}\ (\mathit{Probe}\ (N - (n+1))) = p^{(n+1)+1} + (1 - p^{(n+1)+1}) * P_{err}\ \mathit{Start}$.
qed *simp* – The 0-case is a simple rewriting step with P_{err}-last-probe.

Together with P_{err}-start-iter we prove our final theorem:

theorem P_{err}-start: $P_{err}\ \mathit{Start} = (q * p^{N+1}) / (1 - q * (1 - p^{N+1}))$

With typical parameters for the ZeroConf protocol (16 hosts ($q = 16/65024$), 3 probe runs ($N = 2$) and a probability of $p = 0.01$ to lose ARP packets) we compute (by rewriting) in Isabelle/HOL that the probability to reach *Error* is below $1/10^{13}$:

theorem $P_{err}\ \mathit{Start} \leq 1/10^{13}$

3.4 Expected running time of an allocation run

Users are not only interested in a very low error probability but also in fast allocation time for network address. Obviously there are runs which may take very long, but the probability for these runs are near zero. So we want to verify that the average running time of an allocation run is in the time range of milliseconds.

The running time of an allocation run $C_{fin} :: S \to \overline{\mathbb{R}}$ is modelled as the integral over the sum of all costs ρ for each step in each run. The sum of all steps until either *Ok* or *Error* is reached is simply *cost-until*:

$$C_{fin}\ s = \int_{\omega} \mathit{cost\text{-}until}\ \{\mathit{Error}, \mathit{Ok}\}\ (s \cdot \omega)\ \mathrm{dPr}_s$$

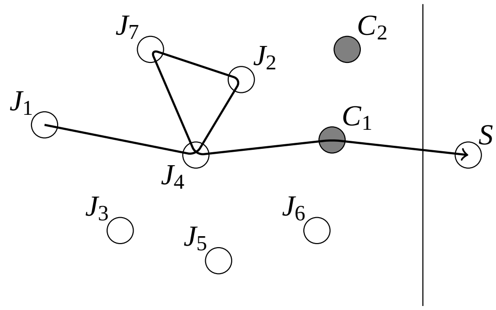

Figure 2: The established route $J_1 - J_4 - J_2 - J_7 - J_4 - C_1 - S$

In order to evaluate the integral we first show that it is finite. This is the case if *cost-until* $\{Error, Ok\}$ is finite almost everywhere. So we first show that almost every path reaches $\{Error, Ok\}$:

lemma AE-term: $s \in S \implies \mathrm{AE}_s\, \omega.\ s \cdot \omega \in \mathit{until}\ S\ \{Error, Ok\}$

Using this we show an elementary form of C_{fin} in a similar way to P_{err}:

lemma C_{fin}-start: $C_{fin}\ Start =$
$(q*(r+p^{N+1}*E+r*p*(1-p^N)/(1-p))+(1-q)*(r*N+1))/(1-q+q*p^{N+1})$

With typical values (16 hosts, 3 probe runs, a probability of $p = 0.01$ to lose ARP packets, 2 ms for an ARP round-trip ($r = 0.002$) and an error penalty of one hour ($E = 3600$)) we compute in Isabelle/HOL that the average time to terminate is less or equal 0.007 s:

theorem $C_{fin}\ Start \leq 0.007$

4 Case study: The Crowds protocol

The *Crowds* protocol described by Reiter and Rubin [22] is an anonymizing protocol. The goal is to allow users to connect to servers anonymously. Neither the final server should know which user connects to it, nor attackers collaborating in the network. The Crowds protocol establishes an anonymizing route through a so called mix network: Each user (Reiter and Rubin name them *jondo* pronounced "John Doe") is itself participating in the mix network. When a jondo establishes a route, it first connects to another random jondo which then decides based on a coin flip weighted with p_f if it should connect to the final server, or go through a further jondo, and so on. Figure 2 shows an established route through the jondos $J_1 - J_4 - J_2 - J_7 - J_4 - C_1 - S$. There is no global information about a route available to the participating jondos. For each connection a jondo only knows its immediate neighbours, but no other previous or following jondo, so it may happen that a route is going through a loop, as seen in Fig. 2.

First, Reiter and Rubin [22] show that the server has no chance to guess the original sender. In a second step they assume that some jondos collaborate to guess the jondo initiating the route. They analyse the probability that a collaborating node is the successor of the initiating jondo. This analysis is affected by the fact that the route may go through the initiating jondo multiple times. An analysis of the Crowds protocol in PRISM, for specific sizes, has been conducted by Shmatikov [23].

Similar to the ZeroConf case, we only analyse the Markov chain having a global view on the protocol. We could model the individual behaviour of jondos in Isabelle/HOL and show that this induces our Markov chain model, but this is not in the scope of this paper.

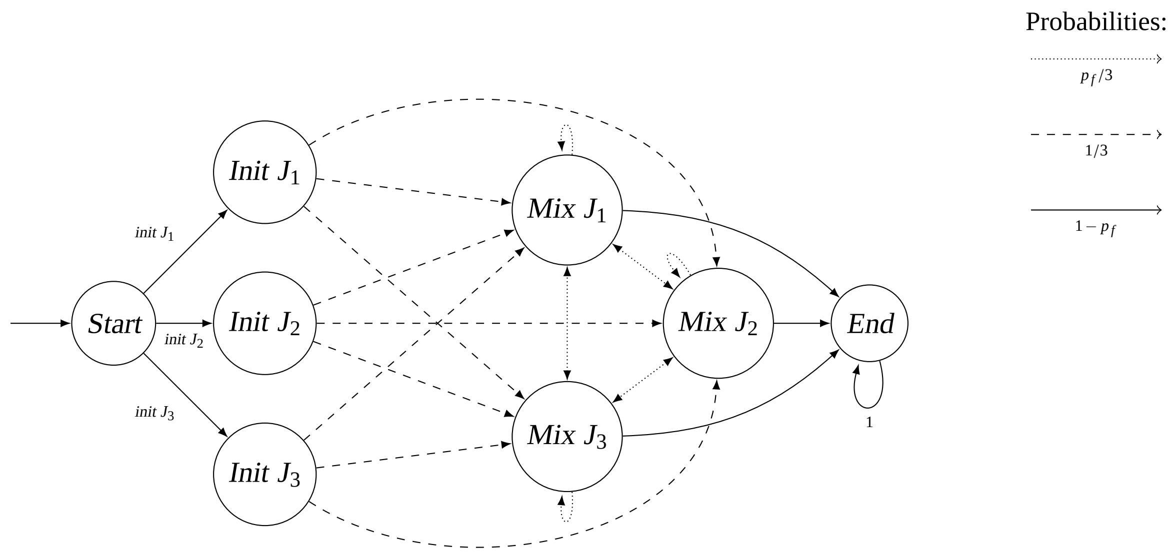

Figure 3: Example Markov chain of the small Crowds network $\{J_1, J_2, J_3\}$

4.1 Formalization of route establishment in the Crowds protocol

We concentrate on the probabilistic aspects of route establishment in the Crowds protocol. We assume a set *jondos* of an arbitrary type α (which is just used to uniquely identify jondos), and a strict subset *colls*, the collaborating attackers. A jondo decides with probability p_f if it chooses another jondo as next step, or if it connects directly to the server. The distribution of the initiating jondos is given by *init*. Naturally the initiating jondo is not a collaborating jondo. In Isabelle this is expressed as the following context:

fixes *jondos colls* $:: \alpha$ *set* **and** $p_f :: \mathbb{R}$ **and** *init* $:: \alpha \to \mathbb{R}$
assumes $0 < p_f$ **and** $p_f < 1$
assumes *jondos* $\neq \emptyset$ **and** *colls* $\neq \emptyset$ **and** *finite jondos* **and** *colls* $\subset$ *jondos*
assumes $\forall j \in \mathit{jondos}.\ 0 \leq \mathit{init}\ j$ **and** $\forall j \in \mathit{colls}.\ \mathit{init}\ j = 0$ **and** $\sum_{j \in \mathit{jondos}} \mathit{init}\ j = 1$

The Markov chain has four different phases: start, the initial node, and the mixing phase, and finally the end phase where the server is contacted. See Fig. 3 for a small example. Our formalization of Markov chains requires a single start node, otherwise we could choose *init* as initial distribution. The type of the state α *c-state* depends on the type of the jondos α.

datatype α *c-state* $=$ *Start* $|$ *Init* α $|$ *Mix* α $|$ *End*

Similar to the ZeroConf protocol not all possible values of *c-state* are necessary. We restrict them further by only allowing non-collaborating jondos as initial jondos, and only elements from *jondos* participate in the mixing phase. With this definition it is easy to show that the set of states $S :: \alpha$ *c-state set* is finite.

$$S = \{\mathit{Start}\} \cup \{\mathit{Init}\ j \mid j \in \mathit{jondos} \setminus \mathit{colls}\} \cup \{\mathit{Mix}\ j \mid j \in \mathit{jondos}\} \cup \{\mathit{End}\}$$

Often we are interested in the jondo referenced by the current state. We introduce *jondo-of* $::$ α *c-state* $\to \alpha$ returning the jondo if we are in an initial or mixing state:

jondo-of $s =$ case s of *Init* $j \Rightarrow j \mid$ *Mix* $j \Rightarrow j$

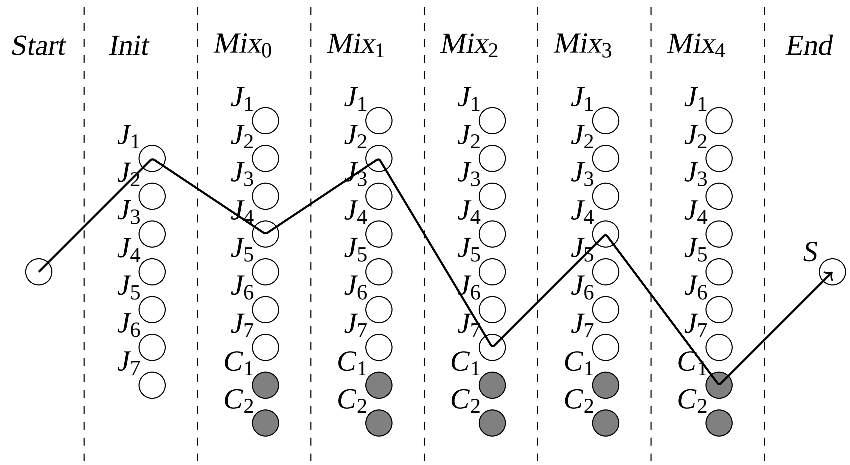

Figure 4: The established route $J_1 - J_4 - J_3 - J_5 - J_4 - C_1 - S$

The transition matrix $\tau :: \alpha$ *c-state* $\to \alpha$ *c-state* $\to \mathbb{R}$ is defined by a case distinction on all possible transitions. The probability for steps from *Start* are given by the distribution of the initiating jondos *init*. The first routing jondo is arbitrarily chosen, and the probability of going from a mixing state to a mixing state is the product of p_f to stay in the mixing phase and the probability $1/J$ for the next jondo. With probability $1 - p_f$ the mixing state is finished and than the Markov chain stays in *End*. Figure 4 shows an example path through the different phases.

$$J = |jondos|$$

$$H = |jondos \setminus colls|$$

$$\begin{array}{lll} \tau\ s\ t = \mathsf{case}\ (s,t)\ \mathsf{of} & (Start,\ \ Init\ j) & \Rightarrow init\ j \\ & \mid (Init\ j,\ \ Mix\ j') & \Rightarrow 1/J \\ & \mid (Mix\ j,\ \ Mix\ j') & \Rightarrow p_f/J \\ & \mid (Mix\ j,\ \ End) & \Rightarrow 1 - p_f \\ & \mid (End,\ \ End) & \Rightarrow 1 \\ & \mid \_ & \Rightarrow 0 \end{array}$$

This completes the definition of the Markov chain describing the route establishment in the Crowds protocol. Finally we show that S and τ describe a discrete-time Markov chain:

theorem *markov-chain* $S\ \tau$

4.2 The jondo contacting the server is independent from the initiating jondo

We define a number of path properties of our Markov chain. The functions *len* $:: (\mathbb{N} \to \alpha$ *c-state*$) \to \mathbb{N}$, *first-jondo* $:: (\mathbb{N} \to \alpha$ *c-state*$) \to \alpha$ and *last-jondo* $:: (\mathbb{N} \to \alpha$ *c-state*$) \to \alpha$ operate on paths not containing the *Start* element. *len* returns the length of the mixing phase, i.e. how many *Mix* states are in the path until *End* is reached, *first-jondo* is the initiating jondo, and *last-jondo* is the jondo contacting the server.

$$\mathit{len}\ \omega = (\mathsf{LEAST}\ n.\ \omega\ n = \mathit{End}) - 2$$

$$\mathit{first\text{-}jondo}\ \omega = \mathit{jondo\text{-}of}\ (\omega\ 0)$$

$$\mathit{last\text{-}jondo}\ \omega = \mathit{jondo\text{-}of}\ (\omega\ (\mathit{len}\ \omega + 1))$$

The path functions *len*, *first-jondo* and *last-jondo* are well defined on almost every path. The paths in our Markov chain do not contain the *Start* element, so the paths start with an *Init* state. Hence for almost every path we know that the first element is an initiating state, then for the next *len* elements we have mixing states, and finally a tail of *End* states:

lemma $AE_{\mathit{Start}}\ \omega.\ \omega \in \mathbb{N} \to S$

lemma $AE_{\mathit{Start}}\ \omega.\ \exists j \in \mathit{jondos} \setminus \mathit{colls}.\ \omega\ 0 = \mathit{Init}\ j$

lemma $AE_{\mathit{Start}}\ \omega.\ \forall i \leq \mathit{len}\ \omega.\ \exists j \in \mathit{jondos}.\ \omega\ (i+1) = \mathit{Mix}\ j$

lemma $AE_{\mathit{Start}}\ \omega.\ \forall i > \mathit{len}\ \omega.\ \omega\ (i+1) = \mathit{End}$

With this we can easily show that the jondo contacting the server is independent from the initiating jondo:

lemma
assumes $l \in \mathit{jondos}$ **and** $i \in \mathit{jondos} \setminus \mathit{colls}$
shows $\Pr(\omega.\ \mathit{first\text{-}jondo}\ \omega = i \wedge \mathit{last\text{-}ncoll}\ \omega = l) =$
$\Pr(\omega.\ \mathit{first\text{-}jondo}\ \omega = i) * \Pr(\omega.\ \mathit{last\text{-}ncoll}\ \omega = l)$

4.3 Probability that initiating jondo contacts a collaborator

The attacker model assumes that the collaborators want to detect the initiator of a route. This is obviously only possible if one of the collaborators is chosen as one of the mixing jondos. We have two goals: (1) If the numbers of collaborators is small, the probability to contact a collaborator should be near zero. (2) We want to analyse the probability that the initiating jondo directly contacts a collaborator. When we know the ratio of collaborators to jondos, how can we adjust p_f, so that this probability is less or equal to $1/2$?

The random variable *hit-colls* :: $(\mathbb{N} \to \alpha\ \mathit{c\text{-}state}) \to \mathbb{B}$ is true if a collaborator participates in the mixing phase, *first-coll* :: $(\mathbb{N} \to \alpha\ \mathit{c\text{-}state}) \to \mathbb{N}$ is the mixing phase in which the collaborator is hit, and *last-ncoll* :: $(\mathbb{N} \to \alpha\ \mathit{c\text{-}state}) \to \alpha$ is the last non-collaborating jondo, i.e. the jondo contacting a collaborator.

$$\mathit{hit\text{-}colls}\ \omega = \exists n, j \in \mathit{colls}.\ \omega\ n = \mathit{Mix}\ j$$

$$\mathit{first\text{-}coll}\ \omega = (\mathsf{LEAST}\ n.\ \exists j \in \mathit{colls}.\ \omega\ n = \mathit{Mix}\ j) - 1$$

$$\mathit{last\text{-}ncoll}\ \omega = \mathit{jondo\text{-}of}\ (\omega\ (\mathit{first\text{-}coll}\ \omega))$$

The property we want to check only makes sense if a collaborator participates in the mixing phase. So we first prove the probability to hit a collaborator:

lemma $\Pr_{\mathit{Start}}(\omega.\ \mathit{hit\text{-}colls}\ \omega) = (1 - H/J)/(1 - H/J * p_f)$

We already see that the probability to hit a collaborator goes to 0 if the number of collaborators and p_f stay constant and $J \longrightarrow \infty$. Then $H/J \longrightarrow 1$ and hence $\Pr_{\mathit{Start}}(\omega.\ \mathit{hit\text{-}colls}\ \omega) \longrightarrow 0$. Thus our first goal is satisfied.

Additionally, we want to control the probability that the initiating jondo hits a collaborator. For this, we compute the probability to have a fixed first and last non-collaborating jondo before we hit a collaborator:

$$\textbf{lemma}\quad \text{P-}\mathit{first\text{-}jondo\text{-}last\text{-}ncoll}: \\ \textbf{assumes } l \in \mathit{jondos} \setminus \mathit{colls} \textbf{ and } i \in \mathit{jondos} \setminus \mathit{colls} \\ \textbf{shows } \Pr(\omega.\ \mathit{first\text{-}jondo}\ \omega = i \wedge \mathit{last\text{-}ncoll}\ \omega = l \mid \mathit{hit\text{-}colls}\ \omega) = \\ \mathit{init}\ i * (p_f / J + (\mathsf{if}\ i = l\ \mathsf{then}\ 1 - H/J * p_f\ \mathsf{else}\ 0))$$

Note that the conditional probability does not divide by 0 because $\Pr_{\mathit{Start}}(\omega.\ \mathit{hit\text{-}colls}\ \omega) \neq 0$ by the previous lemma. By summing up over all possible non-collaborating jondos we show the probability that the last non-collaborating jondo is the initiating jondo:

$$\textbf{theorem}\ \Pr_{\mathit{Start}}(\omega.\ \mathit{first\text{-}jondo}\ \omega = \mathit{last\text{-}ncoll}\ \omega \mid \mathit{hit\text{-}colls}\ \omega) = 1 - (H-1)/J * p_f$$

With this we can now enforce that the probability that the initiating jondo hits a collaborator is less or equal to $\frac{1}{2}$:

$$\textbf{lemma}\quad H > 1\ \wedge\ J/(2*(H-1)) \leq p_f \implies \\ \Pr_{\mathit{Start}}(\omega.\ \mathit{first\text{-}jondo}\ \omega = \mathit{last\text{-}ncoll}\ \omega \mid \mathit{hit\text{-}colls}\ \omega) \leq \tfrac{1}{2}$$

Reiter and Rubin [22] call this probably innocent. Because $p_f < 1$ this is only possible if $1/2 < (H-1)/J$, i.e. more than half of the jondos are non-collaborating. This meets our second goal.

4.4 Information gained by the collaborators

Obviously, in Isabelle/HOL we are not only restricted to state probabilities or expectations. For example, for quantitative information flow analysis, similar to the analysis by Malacaria [20], we are interested in the mutual information $\mathcal{I}_s(X;Y)$ between two random variables X and Y. The mutual information is formalized in Isabelle/HOL using the Radon-Nikodým derivative. However, we know that if X and Y are simple functions, i.e. functions with a finite range, then $\mathcal{I}_s(X;Y)$ can be computed in the known discrete way:

$$\textbf{lemma}\quad \mathit{simple\text{-}function}_s\ X \implies \mathit{simple\text{-}function}_s\ Y \implies \\ \mathcal{I}_s(X;Y) = \textstyle\sum_{(x,y)\in\{(X x, Y x) \mid x.\, x\in\Omega\}}.\Pr_s(\omega.\ X\ \omega = x \wedge Y\ \omega = y) * \\ \log_2\left(\Pr_s(\omega.\ X\ \omega = x \wedge Y\ \omega = y)/(\Pr_s(\omega.\ X\ \omega = x) * \Pr_s(\omega.\ Y\ \omega = y))\right)$$

We are only interested in runs which hit a collaborator. To use mutual information with this restriction we introduce the conditional probability $\Pr_{\mathit{hit\text{-}colls}}$, with the condition that each run hits a collaborator. Its characteristic property (we omit the technical definition) is

$$\textbf{lemma}\quad \mathit{measurable}_s\ P \implies \Pr_{\mathit{hit\text{-}colls}}(\omega.\ P\ \omega) = \Pr_{\mathit{Start}}(\omega.\ P\ \omega \mid \mathit{hit\text{-}colls}\ \omega)$$

With this property and lemma P-*first-jondo-last-ncoll* we can now show an upper bound for the information flow:

$$\textbf{theorem}\quad \mathcal{I}_{\mathit{hit\text{-}colls}}(\mathit{first\text{-}jondo}; \mathit{last\text{-}ncoll}) \leq (1 - (H-1)/J * p_f) * \log_2 H$$

This supports the intuitive understanding that the information the attackers can gain is restricted by the probability that the initiating jondo is the jondo directly contacting a collaborator.

5 Related Work

There is already some work to verify parametric probabilistic models. Hermanns *et al.* [10] implement a probabilistic variant of counterexample-guided abstraction refinement (CEGAR). They handle infinite state spaces by breaking them up into finite partitions. Hahn *et al.* [8] allows parametric transition probabilities. The number of states is still fixed, but the transition probabilities are rational functions over parameter variables. Katoen *et al.* [16] present a method to generate and use quantitative invariants for linear probabilistic programs. Their motivation is to use these invariants to augment interactive proofs.

Now we survey other work that models probabilistic systems in an interactive theorem prover.

We build directly on the formalization of Markov chain theory developed for our verification of pCTL model checking [12], which builds on a formalization of measure theory [11]. Ultimately, all of the work cited in this section builds on the work of Hurd (see below). However, instead of Hurd's probability space $\mathbb{N} \to \mathbb{B}$ we have a probability space on arbitrary functions. This allows for a natural formalization of Markov chains over arbitrary state spaces and needs no encoding into booleans.

The formalization of probability theory in HOL starts with Hurd's thesis [13]. He introduces measure theory, proves Caratheodory's theorem about the existence of measure spaces and uses it to introduce a probability space on infinite boolean sequences. He defines concrete random variables with Bernoulli or uniform distribution. Using this work he also analyses a symmetric simple random walk. Hasan *et al.* [9] formalize the analysis of continuous random variables on Hurd's probability space. However, their work is quite different from ours in that they do not employ Markov chains. Based on Hurd's work, Liu *et al.* [19] define when a stochastic process is a Markov chain. Their theory does not provide everything we need: it is restricted to stochastic processes on Hurd's probability space $\mathbb{N} \to \mathbb{B}$ and does not construct the path space of Markov chains defined by transition probabilities. Coble [7] formalizes information theory on finite probability spaces. He applies it to a quantitative information flow analysis of the Dining Cryptographers protocol. Hurd *et al.* [14] in HOL4 and Audebaud and Paulin-Mohring [2] in Coq formalize semantics of probabilistic programs. Both reason about the probability of program termination and only allow discrete distributions for the result values.

6 Conclusion

The formalizations are available in the Archive of Formal Proofs [15]. For the ZeroConf protocol the formalization was done in a couple of days and required approx. 260 lines of Isabelle/HOL theory. The Crowds protocol requires approx. 1060 lines of Isabelle/HOL theory and it took one person a couple of weeks to verify. The time necessary for the verification includes finding an estimation for the information gained when a collaborator is hit. The probabilities we verified for the ZeroConf protocol and the Crowds protocol are expressible as PCTL formulas. However this is not a restriction of Isabelle/HOL. We can express ω-regular properties or multiple reward structures easily in higher-order logic.

Our future goals include more powerful models like Markov decision processes and continuous-time models but also the certification of probabilistic model checker runs in Isabelle/HOL.

Acknowledgment

We thank Sergio Giro for reading and commenting on a draft of this paper. We also thank the anonymous reviewers for the references on parametric probabilistic model checking.

References

[1] S. Andova, H. Hermanns & J.-P. Katoen (2003): *Discrete-time rewards model-checked*. In K. G. Larsen & P. Niebert, editors: *FORMATS, LNCS* 2791, pp. 88–104, doi:10.1007/978-3-540-40903-8_8.

[2] P. Audebaud & C. Paulin-Mohring (2009): *Proofs of randomized algorithms in Coq*. *Science of Computer Programming* 74(8), pp. 568–589, doi:10.1016/j.scico.2007.09.002.

[3] C. Baier (1998): *On the Algorithmic Verification of Probabilistic Systems*. Habilitation, U. Mannheim.

[4] C. Baier & J.-P. Katoen (2008): *Principles of Model Checking*. The MIT Press, Cambridge, Massachusetts.

[5] H. Bohnenkamp, P. van der Stok, H. Hermanns & F. Vaandrager (2003): *Cost-Optimisation of the IPv4 Zeroconf Protocol*. In: *DSN'03*, IEEE CS Press, pp. 531–540, doi:10.1109/DSN.2003.1209963.

[6] S. Cheshire, B. Aboba & E. Guttman (2005): *Dynamic Configuration of IPv4 Link-Local Addresses*. RFC 3927 (Proposed Standard). Available at `http://www.ietf.org/rfc/rfc3927.txt`.

[7] A. R. Coble (2009): *Anonymity, Information, and Machine-Assisted Proof*. Ph.D. thesis, U. of Cambridge.

[8] E. M. Hahn, H. Hermanns & L. Zhang (2011): *Probabilistic reachability for parametric Markov models*. *Int. J. on Software Tools for Technology Transfer (STTT)* 13(1), pp. 3–19, doi:10.1007/s10009-010-0146-x.

[9] O. Hasan, N. Abbasi, B. Akbarpour, S. Tahar & R. Akbarpour (2009): *Formal Reasoning about Expectation Properties for Continuous Random Variables*. In A. Cavalcanti & D. Dams, editors: *Formal Methods (FM 2009), LNCS* 5850, pp. 435–450, doi:10.1007/978-3-642-05089-3_28.

[10] H. Hermanns, B. Wachter & L. Zhang (2008): *Probabilistic CEGAR*. In A. Gupta & S. Malik, editors: *Computer Aided Verification (CAV 2008), LNCS* 5123, pp. 162–175, doi:10.1007/978-3-540-70545-1_16.

[11] J. Hölzl & A. Heller (2011): *Three Chapters of Measure Theory in Isabelle/HOL*. In M. C. J. D. van Eekelen, H. Geuvers, J. Schmaltz & F. Wiedijk, editors: *ITP 2011, LNCS* 6898, pp. 135–151, doi:10.1007/978-3-642-22863-6_12.

[12] J. Hölzl & T. Nipkow (2012): *Verifying pCTL Model Checking*. In C. Flanagan & B. König, editors: *TACAS 2012, LNCS* 7214, pp. 347–361, doi:10.1007/978-3-642-28756-5_24.

[13] J. Hurd (2002): *Formal Verification of Probabilistic Algorithms*. Ph.D. thesis, U. of Cambridge.

[14] J. Hurd, A. McIver & C. Morgan (2005): *Probabilistic Guarded Commands Mechanized in HOL*. *Theoretical Computer Science* 346(1), pp. 96–112, doi:10.1016/j.tcs.2005.08.005.

[15] J. Hölzl & T. Nipkow (2012): *Markov Models*. *The Archive of Formal Proofs*. `http://afp.sf.net/entries/Markov_Models.shtml`, Formal proof development.

[16] J.-P. Katoen, A. McIver, L. Meinicke & C. C. Morgan (2010): *Linear-Invariant Generation for Probabilistic Programs: - Automated Support for Proof-Based Methods*. In R. Cousot & M. Martel, editors: *Static Analysis (SAS 2010), LNCS* 6337, pp. 390–406, doi:10.1007/978-3-642-15769-1_24.

[17] M. Kwiatkowska, G. Norman & D. Parker (2011): *PRISM 4.0: Verification of Probabilistic Real-time Systems*. *LNCS* 6806, doi:10.1007/978-3-642-22110-1_47.

[18] M. Kwiatkowska, G. Norman, D. Parker & J. Sproston (2006): *Performance Analysis of Probabilistic Timed Automata using Digital Clocks*. *FM in System Design* 29, pp. 33–78, doi:10.1007/s10703-006-0005-2.

[19] L. Liu, O. Hasan & S. Tahar (2011): *Formalization of Finite-State Discrete-Time Markov Chains in HOL*. In T. Bultan & P.-A. Hsiung, editors: *ATVA 2011, LNCS* 6996, pp. 90–104, doi:10.1007/978-3-642-24372-1_8.

[20] P. Malacaria (2007): *Assessing Security Threats of Looping Constructs*. In: *ACM SIGPLAN Symposium on Principles of Programming Languages (POPL'07)*, pp. 225–235, doi:10.1145/1190215.1190251.

[21] T. Nipkow, L. C. Paulson & M. Wenzel (2002): *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*. *LNCS* 2283, Springer, doi:10.1007/3-540-45949-9.

[22] M. Reiter & A. Rubin (1998): *Crowds: Anonymity for web transactions*. *ACM Transactions on Information and System Security (TISSEC)* 1(1), pp. 66–92, doi:10.1145/290163.290168.

[23] V. Shmatikov (2004): *Probabilistic analysis of an anonymity system*. *J. of Comp. Sec.* 12, pp. 355–377.