Preserving Opacity on Interval Markov Chains under Simulation

Béatrice Bérard[*], Olga Kouchnarenko[†], John Mullins[‡], Mathieu Sassolas[§]
[*]Sorbonne Université, UPMC Univ. Paris 06, CNRS UMR 7606, LIP6, Paris, France
[†]Univ. Franche-Comté, FEMTO-ST, CNRS UMR 6174, Besançon, France
[‡]Dept. of Computer & Software Eng., École Polytechnique de Montréal, Montreal (Quebec), Canada
[§]Université Paris-Est, LACL, Créteil, France

***Abstract*—Given a probabilistic transition system (PTS) $\mathcal{A}$ partially observed by an attacker, and an ω-regular predicate φ over the traces of $\mathcal{A}$, measuring the disclosure of the secret φ in $\mathcal{A}$ means computing the probability that an attacker who observes a run of $\mathcal{A}$ can ascertain that its trace belongs to φ. We consider specifications given as Interval Markov Chains (IMCs), which are underspecified Markov chains where probabilities on edges are only required to belong to intervals. Scheduling an IMC $\mathcal{S}$ produces a concrete implementation as a PTS and we define the worst case disclosure of secret φ in $\mathcal{S}$ as the maximal disclosure of φ over all PTSs thus produced. We compute this value for a subclass of IMCs and we prove that simulation between specifications can only improve the opacity of implementations.**

I. Introduction

a) Context and motivation: When modeling complex systems, a top-down approach allows gradually specifying various system requirements, while preserving some behavioral properties, like safety, reachability, and liveness under some conditions.

Security requirements, which are not behavioral ones [1], may not fare well under refinement, unless tailored specially to do so, as in [2]. Several known security properties such as noninference or anonymity can be encoded in the framework of *opacity* [3], [4], [2]. In this context, an external observer tries to discover whether a predicate (given as an ω-regular set) holds by partially observing the system through a projection of its actions. A system is opaque if the attacker fails to discover this information. In the possibilistic setting, a violation of opacity captures the existence of at least one perfect leak.

In probabilistic models like Discrete Time Markov Chains (DTMCs), naturally random events such as faults or message transmission failure, can be taken into account. Opacity was extended in this setting [5], [6], [7] to provide various measures of what is disclosed by observation.

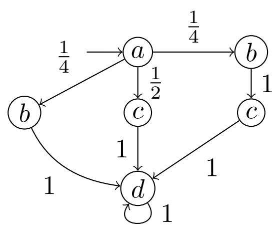
Consider for instance the two systems in Figure 1(a)-(b), which are DTMCs with the addition of labels on states (indicated inside). We assume that the occurrence of b must be kept secret and that all labels except b are observable. In this case, the only runs *disclosing* the secret are those observed by ad^ω, since every such run betrays the occurrence of b. The probability of disclosure is $1/4$ in $\mathcal{A}_1$ while it is $3/4$ in $\mathcal{A}_2$, hence $\mathcal{A}_1$ is more secure than $\mathcal{A}_2$. Our aim is to establish sufficient conditions on systems like $\mathcal{A}_1$ and $\mathcal{A}_2$, that can be compared, for one of them to be more secure than the other.

In the process of system modeling, it is common practice to use *underspecified* models as first steps of specification. A first approach is to consider *sub-stochastic* models where transition probabilities need not sum up to 1. In this framework, the notions of satisfaction and simulation were extensively studied in [8]. The second approach is to introduce non-determinism in the model to describe environment choices [9], [10], [11], [12], [13], [7], [14]. These models have also been studied in relation to the refinement process [9]. For example, both systems of Figure 1(a)-(b) could have been derived from a single underspecified system $\mathcal{S}$ with the same structure but imprecise probabilities, like the one in Figure 1(c). A particular case of such models is the Interval Markov Chains (IMCs) where the transitions are equipped with probability bounds in the form of intervals, as done in Figure 1(c).

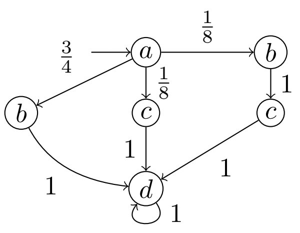
Scheduling is an effective way to obtain implementations of IMCs: at each step, the scheduler provides a distribution satisfying the bounds, producing a (possibly infinite) DTMC on-the-fly. In the case of opacity, a scheduler represents a strategy of an agent inside the system, trying to disclose as much information as possible to a passive observer.

b) Contribution: We investigate opacity for IMCs, defining disclosure in the worst case scenario, as the supremum of the disclosure for all scheduled implementations. This measures the information obtained by the passive observer when the system is controlled by the smartest scheduler in coalition with the observer. We first show how to compute this value for a subclass of IMCs, where no transition can be completely blocked by the scheduler. Note that schedulers were already used in [7] to evaluate disclosure, although in the context of (fully specified) Markov Decision Processes. We then prove that simulations between IMCs can only improve the opacity of all implementations obtained by scheduling. This can be viewed as an extension of the work in [2] to the probabilistic setting. The main difficulty of this result comes from the restriction of the implementations to those obtained by scheduling.

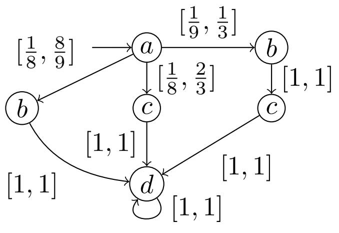
c) Organization of the paper: In Section II we present the underlying models for specification and implementation, with the simulation relation and related semantics. We define probabilistic disclosure in this context and show how to compute it (for a restricted case) in Section III. Finally, we prove monotonicity of opacity under simulation in Section IV.



(a) $\mathcal{A}_1$



(b) $\mathcal{A}_2$



(c) $\mathcal{S}$

Figure 1. Probabilistic systems $\mathcal{A}_1$ or $\mathcal{A}_2$ implementing underspecified system $\mathcal{S}$.

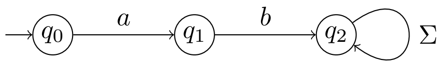


Figure 2. A DPA for $\varphi_b = ab\Sigma^\omega$ with $F(q_0) = F(q_2) = 1$ and $F(q_2) = 2$.

Due to lack of space, some proofs can be found in [15].

II. Models and simulation

The set of natural numbers is denoted by $\mathbb{N}$. The composition of relations $\mathcal{R}_2$ and $\mathcal{R}_1$ is defined by $\mathcal{R}_2 \circ \mathcal{R}_1 = \{(x,z) \mid \exists y, (x,y) \in \mathcal{R}_1 \wedge (y,z) \in \mathcal{R}_2\}$. Given a finite alphabet Σ, we denote by Σ^* (resp. Σ^ω) the set of finite (resp. infinite) words over Σ, with $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$ and ε the empty word.

Given a countable set Z, a discrete distribution is a mapping $\mu : Z \to [0,1]$ such that $\sum_{z\in Z} \mu(z) = 1$. The support of μ is $supp(\mu) = \{z \in Z \mid \mu(z) > 0\}$. The set of all discrete distributions on Z is denoted by $\mathcal{D}ist(Z)$. When dealing with a *joint* distribution on domain $Z_1 \times Z_2$, we write $\mu(Y_1, Y_2) = \sum_{y_1\in Y_1, y_2\in Y_2} \mu(y_1, y_2)$ for $Y_1 \subseteq Z_1$ and $Y_2 \subseteq Z_2$, and we use as shorthands $\mu(y_1, Y_2) = \mu(\{y_1\}, Y_2)$ and $\mu(Y_1, y_2) = \mu(Y_1, \{y_2\})$.

A. Models

The secret to be protected from disclosure is described by a Deterministic Parity Automaton (DPA, see example in Figure 2). Implementations are given by Probabilistic Transition Systems (PTSs). They are classical Discrete Time Markov Chains, with the addition of state labeling [8], restricting the processes of [9] with a countable set of states. Specifications are described by Interval Markov chains (IMCs).

Definition 1 (Deterministic Parity Automaton). *A deterministic parity automaton (DPA) is a tuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of states, Σ is an input alphabet, $\delta : Q \times \Sigma \to Q$ is a transition function, $q_0 \in Q$ is an initial state, and F is a mapping from Q to a finite set of colors $\{1, \ldots, k\}$.*

A run of $\mathcal{A}$ on a word $w = a_1a_2 \cdots \in \Sigma^\omega$ is an infinite sequence $\rho = q_0q_1 \cdots \in Q^\omega$ such that for all $i \geq 0$, $q_{i+1} = \delta(q_i, a_{i+1})$. For such a run ρ, we define *Inf*(ρ) as the set of states appearing infinitely often in the sequence. The run is accepting if $min\{F(q) \mid q \in Inf(\rho)\}$ is even. In this case, the corresponding word is accepted by $\mathcal{A}$ and $\mathcal{L}(\mathcal{A})$ is the subset of Σ^ω of words accepted by $\mathcal{A}$. A subset K of Σ^ω is ω-regular if there is an automaton $\mathcal{A}$ such that $K = \mathcal{L}(\mathcal{A})$.

Definition 2 (Probabilistic Transition System). *A* probabilistic transition system *(PTS) over alphabet Σ is a 4-tuple $\mathcal{A} = \langle Q, q_{init}, \Delta, L\rangle$ where Q is a countable set of states, with $q_{init} \in Q$ the initial state, $\Delta : Q \to \mathcal{D}ist(Q)$ is a mapping associating with any state $q \in Q$ a distribution $\Delta(q)$ over Q, with finite support, and $L : Q \to \Sigma$ is the labeling function on states.*

A (finite or infinite) run of $\mathcal{A}$ starting from state $q \in Q$ is a sequence of states $\rho = q_0q_1q_2\ldots$ such that $q_0 = q$ and for each i, $0 \leq i < |\rho|$, $\Delta(q_i)(q_{i+1}) > 0$. When the run is finite $\rho = q_0q_1 \ldots q_n$, we note $q_n = \mathrm{lst}(\rho)$. The *trace* of ρ is the word $\mathrm{tr}(\rho) = L(q_0)L(q_1)\ldots \in \Sigma^\infty$. We denote by $Runs_q(\mathcal{A})$ the set of infinite runs starting from q and we set $Runs(\mathcal{A}) = Runs_{q_{init}}(\mathcal{A})$, and $Tr(\mathcal{A}) = \{\mathrm{tr}(\rho) \mid \rho \in Runs(\mathcal{A})\}$, the set of traces of $\mathcal{A}$. We also define $FRuns_q(\mathcal{A})$ the set of finite runs starting from q, and similarly $FRuns(\mathcal{A}) = FRuns_{q_{init}}(\mathcal{A})$ and $FTr(\mathcal{A}) = \{\mathrm{tr}(\rho) \mid \rho \in FRuns(\mathcal{A})\}$, the subset of Σ^* of finite traces of $\mathcal{A}$.

Recall [16] that a probability measure $\mathbf{P}_\mathcal{A}$ can be defined on $Runs(\mathcal{A})$: measurable sets are generated by *cones*, where the cone C_ρ associated with a finite run $\rho = q_0q_1 \ldots q_n$ is the subset of infinite runs in $Runs(\mathcal{A})$ having ρ as prefix. The probability of C_ρ is $\mathbf{P}_\mathcal{A}(C_\rho) = \prod_{i=0}^{n-1} \Delta(q_i)(q_{i+1})$. The cone of a word $w \in \Sigma^*$ is defined by $C_w = \bigcup_{\rho\in \mathrm{tr}^{-1}(w)} C_\rho$.

IMCs were introduced for specifications in [9] and further investigated in [11], [13] from a verification point of view. They were also extended to *Constraint Markov Chains* in [12] and to *Parametric IMCs* in [14], with a focus on the consistency problem, *i.e.*, the problem of existence of an implementation satisfying a given specification. We denote by $\mathcal{I}$ the set of intervals in $[0,1]$.

Definition 3 (Interval Markov Chains). *An Interval Markov Chains (IMC) is a 4-tuple $\mathcal{S} = (S, s_{init}, T, \lambda)$ where S is a finite set of states, with $s_{init} \in S$ the initial state, $T : S \to (S \to \mathcal{I})$ associates with any state $s \in S$ a mapping $T(s)$ from S into $\mathcal{I}$, $\lambda : S \to \Sigma$ is the labeling function.*

By extension, $f \in T(s)$ will denote any distribution $f : S \to [0,1]$ (hence $\sum_{s'\in S} f(s') = 1$) such that for all $s' \in S$, $f(s') \in T(s)(s')$.

Several semantics have been given to IMCs [9], [11], [12], [13]. The simplest one is the Uncertain Markov Chain semantics, which corresponds to first choosing all distributions for the states, with probabilities belonging to the specified intervals to obtain an implementation. This results in a PTS, with the same structure as the specification. A richer semantics consists in introducing a scheduler choosing the distribution at each step to obtain an implementation, as in a Markov Decision Process (MDP). Finally, the most general semantics is directly given by the satisfaction relation from [9].

We consider here the MDP semantics. A run of $\mathcal{S}$ starting from a state s is a sequence $s \xrightarrow{\mu_1} s_1 \xrightarrow{\mu_2} \dots$ where $s_i \in S$ and each μ_i is a distribution over S such that $\forall s \in S, \mu_i(s) \in T(s_{i-1})(s)$. As before, we denote by $Runs_s(\mathcal{S})$ the set of runs starting from s, we set $Runs(\mathcal{S}) = Runs_{s_{init}}(\mathcal{S})$, $FRuns(\mathcal{S})$ is the set of finite runs of $\mathcal{S}$ starting from s_{init}, and for a run $\rho = s \xrightarrow{\mu_1} s_1 \xrightarrow{\mu_2} \dots s_{n-1} \xrightarrow{\mu_n} s_n$ in $FRuns(\mathcal{S})$ we define $\mathrm{lst}(\rho) = s_n$.

To associate a probability measure with the runs, it is necessary to resolve the non determinism by a scheduler that chooses a distribution at each step. More precisely:

Definition 4 (Scheduler). *A scheduler A for an IMC specification $\mathcal{S} = (S, s_{init}, T, \lambda)$, is a mapping $A : FRuns(\mathcal{S}) \to \mathcal{D}ist(S)$ such that for each run ρ with $s = lst(\rho)$, $A(\rho)(s') \in T(s)(s')$.*

We denote by $Sched(\mathcal{S})$ the set of schedulers for $\mathcal{S}$. Like for Markov Decision Processes, scheduling $\mathcal{S}$ with A produces a PTS denoted by $\mathcal{S}(A)$ where states are finite runs: $Q \subseteq FRuns(\mathcal{S})$, the initial state is the run containing only the initial state of $\mathcal{S}$: $q_{init} = s_{init}$, and for $\rho \in Q$, $L(\rho) = \lambda(\mathrm{lst}(\rho))$ and $\Delta(\rho)(\rho') = A(\rho)(s')$ for $\rho' = \rho \xrightarrow{A(\rho)} s'$. We note $\underline{sat}(\mathcal{S}) = \{\mathcal{S}(A) \mid A \in Sched(\mathcal{S})\}$.

Note that the Uncertain Markov Chains semantics corresponds to the particular case of memoryless schedulers.

Also, relating MDP semantics with the general notion introduced in [9], it can be seen that scheduling an IMC specification is a particular case of implementation in the sense of [9]. For any scheduler A, $\mathcal{S}(A)$ is a kind of *unfolding* of $\mathcal{S}$, which restricts the structure of $\mathcal{S}(A)$: at each step, the scheduler chooses a valid distribution among successor states. Hence not every implementation in the sense of [9] can be mapped to a scheduler (see [15] for details).

B. Simulation

The notion of simulation relation between probabilistic specifications was introduced in [9], where it is proved to be a sufficient condition for refinement: if $\mathcal{S}_2$ simulates $\mathcal{S}_1$, then all implementations of $\mathcal{S}_1$ are implementations of $\mathcal{S}_2$. This notion, which consists in lifting the usual simulation to distributions [8], is adapted to our setting in Definition 5 below.

Definition 5 (Simulation relation). *For two IMC specifications $\mathcal{S}_1 = (S_1, s_{1,init}, T_1, \lambda_1)$ and $\mathcal{S}_2 = (S_2, s_{2,init}, T_2, \lambda_2)$, $\mathcal{S}_2$ simulates $\mathcal{S}_1$ if there exists a relation $\mathcal{R} \subseteq S_1 \times S_2$ such that $s_{1,init} \mathcal{R} s_{2,init}$ and if $s_1 \mathcal{R} s_2$ then:*

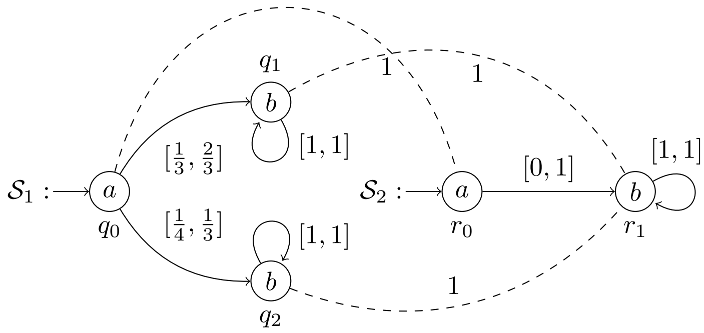


Figure 3. A simulation of $\mathcal{S}_1$ by $\mathcal{S}_2$.

(1) $\lambda_1(s_1) = \lambda_2(s_2)$,
(2) there exists a function $\delta : S_1 \to \mathcal{D}ist(S_2)$ such that for all $f \in T_1(s_1)$ and $s_2' \in S_2$, $\sum_{s_1' \in S_1} f(s_1') \cdot \delta(s_1')(s_2') \in T_2(s_2)(s_2')$,
(3) $s_1' \mathcal{R} s_2'$ whenever $\delta(s_1')(s_2') > 0$.

In a rather counter-intuitive way, Figure 3 illustrates the simulation relation $\mathcal{R}$ of $\mathcal{S}_1$ by $\mathcal{S}_2$, with dashed lines labeled by $\delta(q_i)(r_j)$: for Condition (2) above, we may uniformly use the function: $\delta(q_i)(r_j) = 1$ if $(q_i, q_j) \in \mathcal{R}$ and 0 otherwise.

Note that there is no simulation relation of $\mathcal{S}_2$ by $\mathcal{S}_1$. Indeed, let $f \in T_2(r_0)$ defined as $f(r_1) = 1$. The only way to distribute f over S_1 in order to satisfy Condition (2) is to distribute $\frac{2}{3}$ of f to q_1 and $\frac{1}{3}$ to q_2. Hence, it forces to set $\delta(r_1)(q_1)$ to $\frac{2}{3}$ and $\delta(r_1)(q_2)$ to $\frac{1}{3}$ but this choice for δ is not uniform for any $g \in T_2(r_0)$: for instance, Condition (2) is not satisfied for $g(r_1) = \frac{1}{2}$.

Finally, remark that Definition 5 also applies to PTSs, but intervals reduce to points and Condition (2) becomes (2′):

$$\sum_{s_1' \in S_1} \Delta_1(s_1)(s_1') \cdot \delta(s_1')(s_2') = \Delta_2(s_2)(s_2').$$

III. Opacity

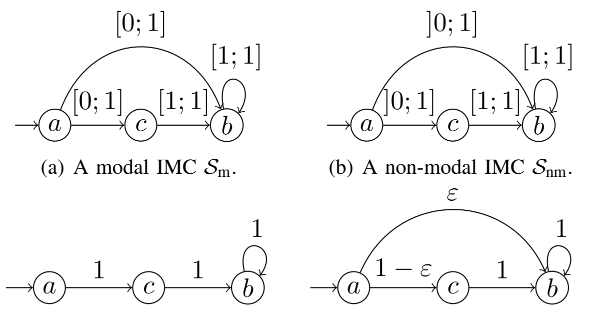
The original definition of opacity was given in [4] for (non probabilistic) transition systems, w.r.t. some observation function $\mathcal{O}$ and some predicate φ (the secret) on the runs of the system. We first define a quantitative version for IMCs and show how its value can be computed in a restricted case.

A. Opacity for probabilistic models

For an ω-regular set $\varphi \subseteq \Sigma^\omega$, a run ρ of a PTS $\mathcal{A}$ satisfies φ if its trace belongs to φ. We consider an *observation function* defined as a morphism $\mathcal{O} : \Sigma^\infty \to \Sigma_{ob}^\infty$, based on a mapping $\pi : \Sigma \to \Sigma_{ob} \cup \{\varepsilon\}$ for a finite alphabet Σ_{ob}.

The set φ is *opaque* with respect to $\mathcal{A}$ and $\mathcal{O}$ if each time a word satisfies φ, another word with the same observation does not. More precisely, the set of words violating this condition is defined by $\mathcal{V}(\mathcal{A}, \mathcal{O}, \varphi) = (Tr(\mathcal{A}) \cap \varphi) \setminus (\mathcal{O}^{-1}(\mathcal{O}(Tr(\mathcal{A}) \setminus \varphi)))$ and φ is opaque if $\mathcal{V}(\mathcal{A}, \mathcal{O}, \varphi) = \emptyset$.

This set is used in [17], [5] to define various notions of probabilistic opacity (for instance in [5] in the particular case where φ corresponds to finite runs reaching secret states). The boolean property is extended by defining the probability of this set, which is measurable since φ is ω-regular:



(a) A modal IMC $\mathcal{S}_{\text{m}}$.

(b) A non-modal IMC $\mathcal{S}_{\text{nm}}$.

(c) A disclosing implementation of $\mathcal{S}_{\text{m}}$ (but not of $\mathcal{S}_{\text{nm}}$).

(d) A non-disclosing implementation of $\mathcal{S}_{\text{nm}}$ (and $\mathcal{S}_{\text{m}}$), $\varepsilon > 0$.

Figure 4. The influence of modal transitions on disclosure.

Definition 6 (Probabilistic Disclosure). *Let $\mathcal{A}$ be a PTS, with observation function $\mathcal{O}$ and ω-regular predicate φ. The* probabilistic disclosure *of φ in $\mathcal{A}$ for $\mathcal{O}$ is $Disc(\mathcal{A}, \mathcal{O}, \varphi) = \mathbf{P}_{\mathcal{A}}(\mathcal{V}(\mathcal{A}, \mathcal{O}, \varphi))$.*

For instance, recall systems $\mathcal{A}_1$ and $\mathcal{A}_2$ of Figure 1. The secret predicate in this case is the set $\varphi_b = ab\Sigma^\omega$ (accepted by the DPA in Figure 2) and the observation function is the projection π onto $\{a, c, d\}^\omega$. This predicate is not opaque since the run abd^ω discloses the occurrence of b. This is measured by the disclosure: $Disc(\mathcal{A}_1, \pi, \varphi_b) = \mathbf{P}_{\mathcal{A}_1}(abd^\omega) = \frac{1}{4}$ and $Disc(\mathcal{A}_2, \pi, \varphi_b) = \mathbf{P}_{\mathcal{A}_2}(abd^\omega) = \frac{3}{4}$.

Remark that disclosure only measures probabilities of the observer being *sure* that the run is in the secret. For example, one can model anonymity of an agent α initiating some protocol by defining φ_α as the set of all runs initiated by α. Anonymity of α is then equivalent to opacity of φ_α. In the case where anonymity is not guaranteed, disclosure provides a measure of the threat. In the case where anonymity holds, this measure will be 0 and does not give any insight on the "strength" of anonymity. Other notions measuring this strength were proposed in [18], [19] and quantitative opacity for partial disclosure of the secret have also been defined in [6], although they are not linear hence do not fare well under standard optimization techniques.

For two PTSs $\mathcal{A}_1$ and $\mathcal{A}_2$ over the same alphabet Σ, predicate φ and observation function $\mathcal{O}$, we say that $\mathcal{A}_1$ is more opaque than $\mathcal{A}_2$ if $Disc(\mathcal{A}_1, \mathcal{O}, \varphi) \leq Disc(\mathcal{A}_2, \mathcal{O}, \varphi)$.

We now lift the notion of disclosure to the set of scheduled implementations of a specification $\mathcal{S}$ by:

$$Disc(\mathcal{S}, \mathcal{O}, \varphi) = \sup_{A \in Sched(\mathcal{S})} Disc(\mathcal{S}(A), \mathcal{O}, \varphi).$$

Note that this notion differs from the similar one in [7] for Markov Decision Processes. The notion presented here is finer since the set of runs measured by the disclosure depends on the scheduled implementation. In [7], the set of runs of the disclosure is defined on the (unscheduled) MDP, and its probability is optimized afterwards. This would not be consistent in IMCs, since two scheduled implementations can have different sets of edges with non-null probability, as explained below.

B. Computing the probabilistic disclosure of a specification

When the interval of an edge in an IMC is non-punctual and closed on the left by 0, then the corresponding action can be completely blocked by a scheduler. Following [9], we call these edges *modal* edges, and IMCs that contain such edges are called *modal IMCs*.

Definition 7 (Modal edge). *An edge $T(s)(s')$ in IMC $\mathcal{S}$ is* modal *if there exists a scheduler A such that in $\mathcal{S}(A)$, for any run ρ with $lst(\rho) = s$, $\Delta(\rho)(\rho \xrightarrow{A(\rho)} s') = 0$.*

In the context of opacity, removing an edge drastically changes the disclosure, since it can remove ambiguities. For example, consider the modal IMC $\mathcal{S}_{\text{m}}$ of Figure 4(a), where a and b are observed and the secret is the presence of c. An implementation of $\mathcal{S}_{\text{m}}$ that blocks the direct edge from a to b (Figure 4(c)) has a disclosure of 1, since the secret is guaranteed to be part of the only possible run. On the other hand, in the non-modal version of the IMC (Figure 4(b)), such implementations are banned and only implementations that retain a small probability to avoid c are allowed. In these implementations, the disclosure is 0, since every run is observed as ab^ω and it is possible that c did not occur.

The detection of modal edges is the first step toward computation of the disclosure of an IMC.

Proposition 1. *The set of modal edges can be computed in time polynomial in the number of edges.*

Proof. The decision procedure for each edge is as follows:

- if an edge is not weighted by an interval containing 0, it is not modal;
- otherwise, compute the sum of maximal interval values of all other edges stemming from the same state;
  - if this sum is > 1, the edge is modal;
  - if this sum is < 1, the edge is not modal;
  - otherwise (the sum is $= 1$), the edge is modal if, and only if, all intervals of other outgoing edges are closed on the right.

□

Note that the procedure does not rely entirely on the syntactic criterion of an interval closed on 0: it is sufficient but may lead to false positives. For example, consider a state with two outgoing edges $e_1, [\frac{1}{4}; \frac{2}{3}]$ and $e_2, [0; 1]$. The e_2 edge is not actually modal since any probability distribution satisfying the specification can give at most $\frac{2}{3}$ to e_1, hence must at least give weight $\frac{1}{3}$ to e_2. This is avoided by the pre-computation of the least possible probability that can be put on an edge.

In the case of non-modal IMCs, disclosure can be computed:

Theorem 1. *Computing the value of disclosure for an IMC $\mathcal{S}$ without modal edges can be done in 2EXPTIME.*

Proof. Note that intervals may be closed or open on any non-zero bound, which is not the case of IMCs in [11] where all intervals are closed. Hence our procedure adapts ideas from this work to deal with the general case.

First remark that there exists a regular language K such that for any scheduler A, $\mathit{Tr}(\mathcal{S}(A)) = K$. This is only true because $\mathcal{S}$ is assumed non-modal. Let $\mathcal{A}_K$ be a PTS such that $\mathit{Tr}(\mathcal{A}_K) = K$; it can be chosen of size $|\mathcal{S}|$. By the definition of disclosure, if the secret φ is ω-regular and the observation function $\mathcal{O}$ is a projection, then finding the supremum of the disclosure means finding the maximal probability to reach an ω-regular set of runs, namely $\mathcal{V}(\mathcal{A}_K, \mathcal{O}, \varphi)$.

Then we claim that open intervals can be handled as closed ones when trying to optimize the probability of $\mathcal{V}(\mathcal{A}_K, \mathcal{O}, \varphi)$. Indeed, if the optimal scheduler uses a value x which is the bound of an open interval, then one can build a family of schedulers using value $x \pm \frac{1}{2^n}$ for the nth scheduler. The limit probability of reaching $\mathcal{V}(\mathcal{A}_K, \mathcal{O}, \varphi)$ is therefore the one computed when using exact value x. Remark that using closed intervals may introduce intervals containing 0, although it is of no concern since the observation classes are already defined and may not change, only their probability may change. Said otherwise, this does not mean that we are computing disclosure of the closed version, since it is only a probability. On the example of Figure 4(b), it means trying to compute the maximal probability of the empty set, which is indeed zero.

The procedure is hence as follows. Starting from a DPA $\mathcal{A}_\varphi$ for φ, a DPA $\mathcal{A}_\mathcal{V}$ for $\mathcal{V}(\mathcal{A}_K, \mathcal{O}, \varphi)$ can be built, with size exponential in the size of $\mathcal{S}$ and $\mathcal{A}_\varphi$ (and with a number k of colors polynomial in the size of $\mathcal{A}$ and $\mathcal{A}_\varphi$). This construction relies on intersections and complementations of DPA, with a determinization step that brings the exponential blowup [20].

The construction of [11] yields a memoryless scheduler, although it is memoryless on the product, and hence is finite-memory on the original IMC. The procedure of [11] is in EXPTIME with respect to the size of its input, hence computation of disclosure is doubly exponential: $2^{2^{O(|\mathcal{A}|\times|\mathcal{A}_\varphi|)}}$. $\square$

Remarks on modal edges. When a scheduler is faced with the choice to include or exclude a modal edge, it can produce several versions of PTSs, say $\mathcal{A}_1$ and $\mathcal{A}_2$, with $\mathit{Tr}(\mathcal{A}_1) \neq \mathit{Tr}(\mathcal{A}_2)$, hence $\mathcal{V}(\mathcal{A}_1, \mathcal{O}, \varphi) \neq \mathcal{V}(\mathcal{A}_2, \mathcal{O}, \varphi)$. In addition, these choices may be history dependent, as in the example of Figure 5, with $\varphi = a\Sigma^\omega$ and only letters c and d being observed. Intuitively, a way for the scheduler to always disclose the presence of an initial a is to always follow an a by the same letter, say a c. However, this choice must be made after the first letter has been seen. Moreover, leaving the possibility of a run $ad\cdots$ to occur means that run $ac\cdots$ does not disclose φ. As a result, the scheduler should also take into account φ and the observation function before committing to a choice with respect to modal edges. So far, the general case of modal IMCs remains open.

IV. Monotonicity of opacity under simulation

The last section is devoted to the proof of the following result, establishing monotonicity for the disclosure over scheduled implementations:

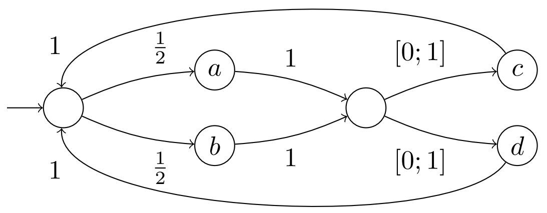


Figure 5. IMC where the choice on modal edge requires history.

$$\begin{tikzcd}
\mathcal{S}_2 \arrow[r, "\mathcal{R}"] \arrow[d, dashed, "sat_2"'] \arrow[dr, dotted, "sat_1 \circ \mathcal{R}"] & \mathcal{S}_1 \arrow[d, "sat_1"] \\
\mathcal{S}_2(A_2) \arrow[r, dashed, "\mathcal{R}'"] & \mathcal{S}_1(A_1)
\end{tikzcd}$$

Figure 6. The result of Theorem 3. Relation $sat_1 \circ \mathcal{R}$ always exists but might not be a scheduling.

Theorem 2. *Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be IMC specifications such that $\mathcal{S}_2$ simulates $\mathcal{S}_1$. Then $\mathit{Disc}(\mathcal{S}_1, \mathcal{O}, \varphi) \leq \mathit{Disc}(\mathcal{S}_2, \mathcal{O}, \varphi)$.*

Since scheduling is a restrictive way to derive implementations from a specification, it is not the case in general that $\underline{sat}(\mathcal{S}_1) \subseteq \underline{sat}(\mathcal{S}_2)$: although any scheduling $\mathcal{S}_1(A_1)$ of $\mathcal{S}_1$ with A_1 is an implementation of $\mathcal{S}_2$ this implementation may not be a scheduling. Instead, the proof builds a scheduler A_2 for $\mathcal{S}_2$, producing an implementation $\mathcal{S}_2(A_2)$ that *simulates* $\mathcal{S}_1(A_1)$ (Theorem 3, illustrated in Figure 6). Then, this simulation is shown to ensure that the probabilities of (cones of) finite words coincide (Propositions 2 and 3). The disclosure set being a measurable event, coincidence of probabilities on cones ensures coincidence of probabilities for the disclosure.

Notations. Given two specifications $\mathcal{S}_1$ and $\mathcal{S}_2$ such that $\mathcal{S}_2$ simulates $\mathcal{S}_1$ through relation $\mathcal{R}$, we define the relation $\sim$ on $\mathit{FRuns}(\mathcal{S}_1) \times \mathit{FRuns}(\mathcal{S}_2)$ by: $\rho_1 \sim \rho_2$ if $|\rho_1| = |\rho_2|$ and at any intermediate step i, the corresponding states satisfy $s_{1,i} \mathcal{R} s_{2,i}$.

Let A_1 and A_2 be two schedulers of $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. We set $\mathcal{A}_1 = \mathcal{S}_1(A_1)$ and $\mathcal{A}_2 = \mathcal{S}_2(A_2)$, with respective sets of states Q_1 and Q_2. For $\rho_2 \in Q_2$, we set $sim(\rho_2) = \{\rho_1 \in Q_1 \mid \rho_1 \sim \rho_2\}$. We now define a measure μ_{ρ_2} over $sim(\rho_2)$ by $\mu_{\rho_2}(\rho_1) = \frac{\mathbf{P}_{\mathcal{A}_1}(\rho_1)}{\mathbf{P}_{\mathcal{A}_1}(sim(\rho_2))}$ (where the probability of finite run ρ is abusively written instead of the probability of its cone C_ρ).

We first show how to build a scheduler for $\mathcal{S}_2$ that simulates the scheduling of $\mathcal{S}_1$. The proof of the theorem below is given in [15].

Theorem 3. *Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be IMC specifications such that $\mathcal{S}_2$ simulates $\mathcal{S}_1$. Then for any $A_1 \in \mathit{Sched}(\mathcal{S}_1)$ there exists $A_2 \in \mathit{Sched}(\mathcal{S}_2)$ such that $\mathcal{S}_2(A_2)$ simulates $\mathcal{S}_1(A_1)$.*

Now we show that simulation between two PTSs is sufficient to compare their disclosure. Namely, we show that the probabilities of cones of words are equal in both systems. Note that although this property is well known to hold for paths, it needs to be lifted to words in order to compare disclosure.

We start by considering the sets of traces globally; although it is folklore that simulation implies trace inclusion, we provide a proof for completeness sake.

Proposition 2. *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be PTSs such that $\mathcal{A}_2$ simulates $\mathcal{A}_1$. Then $Tr(\mathcal{A}_1) = Tr(\mathcal{A}_2)$.*

Proof. We prove the proposition by induction on a strengthened statement. Namely, we claim that for every finite run in $\mathcal{A}_1$ there exists a similar run in $\mathcal{A}_2$. Since an infinite run is the limit of the sequence of its finite prefixes, this claim is sufficient to prove the proposition. Assume by induction that the proposition holds for every word of length n. Let $w \in \mathit{FTr}(\mathcal{A}_1)$ of length $n+1$. We write $w = w_0 a$ for some $a \in \Sigma$. Consider a run of $\mathcal{A}_1$ that produces w. It is of the form $\rho_0 s'_1$ where $\lambda(s'_1) = a$; let $s_1 = \mathrm{lst}(\rho_0)$. Let ρ'_0 be a run in $\mathcal{A}_2$, similar to ρ_0, and $s_2 = \mathrm{lst}(\rho'_0)$. By definition of simulation, there exists a function δ such that for any state s'_2 of $\mathcal{A}_2$,
$$\Delta_2(s_2)(s'_2) = \sum_{\sigma_1 \in S_1} \Delta_1(s_1)(\sigma_1) \cdot \delta(\sigma_1)(s'_2).$$
Moreover, whenever $\delta(\sigma_1)(s'_2) > 0$, $\lambda(s'_1) = \lambda(s'_2)$. Since $\delta(s'_1)$ is a distribution over S_2, $\delta(s'_1)(s'_2) > 0$ for at least one state s'_2. Hence $\rho'_0 s'_2$ is similar to ρ, which shows in particular that $w \in \mathit{FTr}(\mathcal{A}_2)$. □

We additionally show that probabilities coincide:

Proposition 3. *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be PTSs such that $\mathcal{A}_2$ simulates $\mathcal{A}_1$. Then for all $w \in \Sigma^*$, $\mathbf{P}_{\mathcal{A}_1}(C_w) = \mathbf{P}_{\mathcal{A}_2}(C_w)$.*

Since a given word may be produced by several paths, their probabilities should be considered altogether. Hence the proof of the above proposition is not immediate; it is quite technical and can be found in [15].

Existing properties about simulation for PTSs can be retrieved as consequences of the above result. They were for example obtained as a particular case of sub-stochastic simulation in [8]. Although not necessary to prove the main theorem, these results illustrate how constraining simulation between PTSs is.

Recall that a probabilistic bisimulation [9] is a bisimulation that preserves transition probabilities, *i.e.*, a bisimulation relation $\mathcal{R}$ on states such that for any equivalence class R of $\mathcal{R}$, and any two related states $s\mathcal{R}s'$, $\Delta(s)(R) = \Delta(s')(R)$.

Corollary 1. ([8]) Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be PTSs such that $\mathcal{A}_2$ simulates $\mathcal{A}_1$. Then there exists a probabilistic bisimulation over the union of both PTSs.

Corollary 2. ([8]) Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be PTSs such that $\mathcal{A}_2$ simulates $\mathcal{A}_1$. Then $\mathcal{A}_1$ also simulates $\mathcal{A}_2$.

We are now ready to prove Theorem 2:

Proof. Let $\mathcal{A}_1 \in \underline{sat}(\mathcal{S}_1)$. By Theorem 3 there exists $\mathcal{A}_2 \in \underline{sat}(\mathcal{S}_2)$ that simulates $\mathcal{A}_1$. By Proposition 3, $\mathbf{P}_{\mathcal{A}_1}(C_w) = \mathbf{P}_{\mathcal{A}_2}(C_w)$ for every word $w \in \mathit{FTr}(\mathcal{A}_1)$. Hence, for any ω-regular (hence measurable) language $\mathcal{L}$, one has $\mathbf{P}_{\mathcal{A}_1}(\mathcal{L}) = \mathbf{P}_{\mathcal{A}_2}(\mathcal{L})$. It is in particular the case for $\mathcal{V}(\mathcal{A}_1, \mathcal{O}, \varphi) = \mathcal{V}(\mathcal{A}_2, \mathcal{O}, \varphi)$. Therefore, $Disc(\mathcal{A}_1, \mathcal{O}, \varphi) = Disc(\mathcal{A}_2, \mathcal{O}, \varphi)$. Consequently, the theorem holds. □

V. Conclusion

This work investigates how simulation between probabilistic models impacts the security – modeled as opacity. Directions for future work include computing disclosure for IMCs with modal edges. In addition, while we considered here only the *worst case scenario*, it would be interesting to handle also the best case, thus providing bounds on the disclosure of all possible implementations. Finally, we plan to extend our results to CMCs or Parametric IMCs from [12], [14].

References

[1] M. R. Clarkson and F. B. Schneider, "Hyperproperties," *Journal of Computer Security*, vol. 18, no. 6, pp. 1157–1210, Sep. 2010.

[2] R. Alur, P. Černý, and S. Zdancewic, "Preserving secrecy under refinement," in *Proc. ICALP'06*, ser. LNCS, vol. 4052. Springer, 2006, pp. 107–118.

[3] L. Mazaré, "Decidability of opacity with non-atomic keys," in *Proc. FAST'04*, ser. Intl. Federation for Information Processing, vol. 173. Springer, 2005, pp. 71–84.

[4] J. W. Bryans, M. Koutny, L. Mazaré, and P. Y. A. Ryan, "Opacity generalised to transition systems," *Intl. Jour. of Information Security*, vol. 7, no. 6, pp. 421–435, 2008.

[5] A. Saboori and C. N. Hadjicostis, "Current-state opacity formulations in probabilistic finite automata," *IEEE Trans. Automat. Contr.*, vol. 59, no. 1, pp. 120–133, 2014.

[6] B. Bérard, J. Mullins, and M. Sassolas, "Quantifying opacity," *Mathematical Structures in Computer Science*, vol. 25, no. 2, pp. 361–403, 2015.

[7] B. Bérard, K. Chatterjee, and N. Sznajder, "Probabilistic Opacity for Markov decision processes," *Inf. Process. Lett.*, vol. 115, no. 1, pp. 52–59, 2015.

[8] C. Baier, J.-P. Katoen, H. Hermanns, and V. Wolf, "Comparative branching-time semantics for Markov chains," *Information and Computation*, vol. 200, pp. 149–214, 2005.

[9] B. Jonsson and K. G. Larsen, "Specification and refinement of probabilistic processes," in *Proc. LICS'91*. IEEE Computer Society, 1991, pp. 266–277.

[10] R. Segala, "Modeling and Verification of Randomized Distributed Real-Time Systems," Ph.D. dissertation, MIT, Department of Electrical Engineering and Computer Science, 1995.

[11] K. Chatterjee, T. Henzinger, and K. Sen, "Model-Checking omega-Regular Properties of Interval Markov Chains," in *Proc. FoSSaCS'08*, R. M. Amadio, Ed., 2008, pp. 302–317.

[12] B. Caillaud, B. Delahaye, K. G. Larsen, A. Legay, M. L. Pedersen, and A. Wasowski, "Constraint Markov Chains," *Theor. Comput. Sci.*, vol. 412, no. 34, pp. 4373–4404, 2011.

[13] M. Benedikt, R. Lenhardt, and J. Worrell, "LTL Model Checking of Interval Markov Chains," in *Proceedings of TACAS'13*, ser. LNCS, vol. 7795. Springer, 2013, pp. 32–46.

[14] B. Delahaye, "Consistency for Parametric Interval Markov Chains," in *Proc. SynCoP'15*, ser. OASICS, É. André and G. Frehse, Eds., vol. 44. Schloss Dagstuhl - LZI, 2015, pp. 17–32.

[15] B. Bérard, O. Kouchnarenko, J. Mullins, and M. Sassolas, "Probabilistic opacity in refinement-based modeling," *CoRR*, 2015. [Online]. Available: http://arxiv.org/abs/1510.04316

[16] P. Billingsley, *Probability and Measure*, 3rd ed. Wiley, 1995.

[17] B. Bérard, J. Mullins, and M. Sassolas, "Quantifying opacity," in *Proc. QEST'10*, G. Ciardo and R. Segala, Eds. IEEE Computer Society, Sep. 2010, pp. 263–272.

[18] D. Chaum, "The dining cryptographers problem: unconditional sender and recipient untraceability," *Journal of Cryptology*, vol. 1, pp. 65–75, 1988.

[19] M. Bhargava and C. Palamidessi, "Probabilistic Anonymity," in *Proc. CONCUR'05*, ser. LNCS, M. Abadi and L. de Alfaro, Eds., vol. 3653, 2005, pp. 171–185.

[20] N. Piterman, "From Nondeterministic Büchi and Streett Automata to Deterministic Parity Automata," *Logical Methods in Computer Science*, vol. 3, no. 3, 2007.