Information Processing Letters 115 (2015) 52–59

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Probabilistic opacity for Markov decision processes

Béatrice Bérard [a,b], Krishnendu Chatterjee [c], Nathalie Sznajder [a,b,*]

[a] Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6, F-75005, Paris, France
[b] CNRS, UMR 7606, LIP6, F-75005, Paris, France
[c] IST Austria (Institute of Science and Technology, Austria), Austria

ARTICLE INFO

Article history:
Received 31 December 2013
Received in revised form 26 August 2014
Accepted 2 September 2014
Available online 6 September 2014
Communicated by A. Muscholl

Keywords:
Formal methods
Security properties
Opacity
Markov decision processes
Perfect and partial information
Decidability

ABSTRACT

Opacity is a generic security property, that has been defined on (non-probabilistic) transition systems and later on Markov chains with labels. For a secret predicate, given as a subset of runs, and a function describing the view of an external observer, the value of interest for opacity is a measure of the set of runs disclosing the secret. We extend this definition to the richer framework of Markov decision processes, where non-deterministic choice is combined with probabilistic transitions, and we study related decidability problems with partial or complete observation hypotheses for the schedulers. We prove that all questions are decidable with complete observation and ω-regular secrets. With partial observation, we prove that all quantitative questions are undecidable but the question whether a system is almost surely non-opaque becomes decidable for a restricted class of ω-regular secrets, as well as for all ω-regular secrets under finite-memory schedulers.

1. Introduction

Due to the tremendous increase in network communications in the last thirty years, a large amount of work was devoted to the study of security properties, to ensure the preservation of secret data during these communications. *Information flow* characterizes the (possibly illegal and indirect) transmission of such data from a high level user to a low level one. Already in the eighties, a basic version of non-interference was defined in [20], stating that a system is secure if high level actions cannot be detected by low level observations. Among all the subsequent studies, opacity was introduced in [24,7] as a general framework where a wide range of security properties can be specified, for a system interacting with a passive attacker. For a system $\mathcal{S}$, opacity is parameterized by a secret predicate φ described as a subset of executions and an observation function over executions. The system is opaque if, for any secret run in φ, there is another run not in φ with the same observation. When this property is satisfied, the passive attacker cannot learn from the observation if the execution is secret. Ensuring opacity by controller synthesis was further studied in [18,9] while relations with two-player games were established in [23].

Deciding opacity, however, only provides a yes/no answer, but no evaluation of the amount of information gained by a passive attacker. Since more and more security protocols make use of randomization to reach some security objectives [16,29], it becomes important to extend specification frameworks in order to handle measures of information leaks. For this reason, quantitative approaches for security properties were already advocated in [25,34], mostly based on information theory. From this point on, numerous studies were devoted to the computation of (covert) channel capacity in various cases (see e.g. [22]) or more generally information leakage.

* Corresponding author.
E-mail addresses: Beatrice.Berard@lip6.fr (B. Bérard), Krishnendu.Chatterjee@ist.ac.at (K. Chatterjee), Nathalie.Sznajder@lip6.fr (N. Sznajder).

http://dx.doi.org/10.1016/j.ipl.2014.09.001
0020-0190/

To provide quantitative measures of opacity, several definitions have been proposed in a probabilistic setting [21,2,5,8,3,31]. They were, however, restricted to purely probabilistic models, based on Markov chains equipped with labels, to permit observations on runs. We show here how to extend some measures of [3] to Markov decision processes (MDPs) with infinite runs. The simplest one computes what we call here the *probabilistic disclosure*, providing a probabilistic measure for the set of runs whose observation reveals that a secret run has been executed. With the richer model of MDPs, where non-determinism is combined with probabilities, a scheduler can cooperate with the passive external observer to break the system opacity. We focus on ω-regular secrets and morphisms for the observation functions, and prove that the probabilistic disclosure can be computed when the scheduler can distinguish the states of the model. The class of ω-regular languages provides a robust specification language [32], extending classical regular languages from finite words to infinite words. Such ω-regular languages are often needed to express opacity in the non-probabilistic as well as the probabilistic setting. With partial observation for the schedulers, the question whether a system is almost surely non-opaque remains decidable for a restricted class of ω-regular secrets, as well as for all ω-regular secrets under finite-memory schedulers, whereas all quantitative problems become undecidable. Moreover, for all decidable results we present optimal complexity results: for complete observation (where the scheduler can distinguish states of the model) we present polynomial-time results with respect to the size of the model, whereas for partial observation, for all decidable results we show EXPTIME-completeness.

We recall some definitions for probabilistic models in Section 2. Opacity and disclosure are defined for Markov decision processes in Section 3 and proofs for the (un)decidability results are given in Section 4. We conclude in Section 5.

2. Preliminaries

For a finite alphabet Z, we denote by Z^* the set of finite words over Z, by Z^ω the set of infinite words over Z, with $Z^\infty = Z^* \cup Z^\omega$.

We first recall some classical notions on automata.

2.1. Automata

Definition 1. A (deterministic) automaton 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 the initial state, and F is either a subset of Q, or a mapping from Q to a finite subset of natural numbers.

Accepting conditions defined from F will be described hereafter.

A run of the automaton $\mathcal{A}$ on a word $w = a_1 a_2 \cdots \in \Sigma^\omega$ is an infinite sequence $\rho = q_0 q_1 \cdots$ such that for all $i \geq 0$, $q_{i+1} = \delta(q_i, a_{i+1})$. The accepting runs of an automaton are defined according to the acceptance condition. In the sequel, we consider Büchi, co-Büchi and parity acceptance conditions.

For a run $\rho = q_0 q_1 \cdots \in Q^\omega$, we let $\mathrm{Inf}(\rho)$ be the set of states appearing infinitely often in the sequence. When $F \subseteq Q$, we note $\text{Büchi}(F) = \{\rho \in Q^\omega \mid \mathrm{Inf}(\rho) \cap F \neq \emptyset\}$ and $\text{co-Büchi}(F) = \{\rho \in Q^\omega \mid \mathrm{Inf}(\rho) \cap F = \emptyset\}$. When $F : Q \to \{1, \ldots, k\}$, with $k \in \mathbb{N}$, the acceptance condition is a parity condition. We note $\text{Parity}(F) = \{\rho \in Q^\omega \mid \min\{F(q) \mid q \in \mathrm{Inf}(\rho)\} \text{ is even}\}$. For an acceptance condition $Acc \in \{\text{Büchi}(F), \text{co-Büchi}(F), \text{Parity}(F)\}$, we say that a run ρ over a word w is accepting if it is in Acc. The word w is then said to be accepted by ρ.

We denote respectively by $L_B(\mathcal{A})$, $L_C(\mathcal{A})$ and $L_P(\mathcal{A})$ the set of words accepted by the runs of $\mathcal{A}$ in $\text{Büchi}(F)$, $\text{co-Büchi}(F)$ and $\text{Parity}(F)$. A subset L of Σ^ω is ω-regular if there is an automaton $\mathcal{A}$ such that $L = L_P(\mathcal{A})$.

In the sequel, we write DBA for deterministic Büchi automata, DCA for deterministic co-Büchi automata and DPA for deterministic parity automata, according to the choice of acceptance condition.

2.2. Probabilistic systems

We consider systems modeled by Markov decision processes, that generalize Markov chains by combining non-deterministic actions with probabilistic transitions. To define opacity measures on Markov chains, the probabilistic transitions are equipped with labels that may be used to define an observation function on runs. In the setting of Markov decision processes, labels are also added on the probabilistic transitions. They may be observed by a passive attacker while non-deterministic actions are chosen by a scheduler, as explained below.

Given a countable set S, a discrete distribution is a mapping $\mu : S \to [0, 1]$ such that $\sum_{s \in S} \mu(s) = 1$. The set of all discrete distributions on S is denoted by $\mathcal{D}(S)$.

Definition 2 *(Markov Decision Process).* A Markov decision process (MDP) is a tuple $\mathcal{A} = (Q, \Sigma, Act, \Delta, q_0)$ where:

- Q is a finite set of states,
- Act is a finite set of actions,
- Σ is a finite alphabet for the labeling of transitions,
- $\Delta : Q \times Act \to \mathcal{D}(\Sigma \times Q)$ is a (partial) transition function that associates with a state and an action from Act a probability distribution over the possible transition labels and successor states,
- q_0 is the initial state.

Fig. 1 shows an MDP with four actions. Actions α_1 and α_2 bear two different distributions for labels a and b. They start either from state q_0 or from state q_0', and lead to either q_1 or q_2. Actions β_1 and β_2 start from q_1 and q_2 respectively and return to q_0 or q_0' with probability $\frac{1}{2}$.

The definition could be extended with an initial distribution instead of an initial state, but we restrict to this one for the sake of simplicity. When $\Delta(q, \alpha)$ is defined, α is said to be *enabled* in state q. Intuitively, in an execution of an MDP, from a given state q, an action $\alpha \in Act$ enabled in q is chosen non-deterministically, and then the next label in Σ and the next state are chosen according