Execution-time opacity control for timed automata

Étienne André[1,2†], Marie Duflot[3†], Laetitia Laversa[4†], Engel Lefaucheux[3†]

[1]CNRS, Laboratoire d'Informatique de Paris Nord, LIPN, Université Sorbonne Paris Nord, Av. Jean-Baptiste Clément, Villetaneuse, F-93430, France.
[2]Institut Universitaire de France (IUF), France.
[3]Université de Lorraine, CNRS, Inria, LORIA, France.
[4]Université Paris Cité, CNRS, IRIF, F-75013, Paris, France.

[†]These authors contributed equally to this work.

Abstract

Timing leaks in timed automata (TA) can occur whenever an attacker is able to deduce a secret by observing some timed behaviour. In execution-time opacity, the attacker aims at deducing whether a private location was visited, by observing only the execution time. In earlier work, it was shown that it can be decided whether a TA is opaque in this setting. In this work, we address control, and investigate whether a TA can be controlled by a strategy at runtime to ensure opacity, by enabling or disabling some controllable actions over time. We first show that, in general, it is undecidable to determine whether such a strategy exists. Second, we show that deciding whether a meta-strategy ensuring opacity exists can be done in EXPSPACE. Such a meta-strategy is a set of strategies allowing an arbitrarily large—yet finite—number of strategy changes per time unit, and with only weak ordering relations between such changes. Our method is constructive, in the sense that we can exhibit such a meta-strategy. We also extend our method to the case of weak opacity, when it is harmless that the attacker deduces that the private location was *not* visited. Finally, we consider a variant where the attacker cannot have an infinite precision in its observations.

Keywords: timed automata, opacity, side-channel attacks, timed control

1 Introduction

In order to infer sensitive information, side-channels attacks [Sta10] exploit various observable aspects of a system rather than directly exploiting its computational processes; such observable aspects can include power consumption, electromagnetic emissions, or time. In particular, by observing subtle differences in timing, attackers can infer valuable information about the internal state of the system. For example, in [CHS+22], a timing attack vulnerability is identified in the Chinese public key cryptography standard; the authors show how the most significant zero-bit leakage obtained from the execution time allows to extract the secret key.

Timing attacks such as timing leaks often depend on the precise duration of operations, which finite-state automata cannot model. Timed automata [AD94] (TAs), on the other hand, incorporate explicit clocks and timing constraints, making them essential for analysing and detecting vulnerabilities related to timing information. TAs are a powerful formalism to reason about real-time systems mixing timing constraints and concurrency. Timing leaks can occur whenever an

attacker is able to deduce a secret by observing some (timed) behaviour of a TA.

1.1 Related works

Opacity in timed automata

Franck Cassez proposed in [Cas09] a first definition of *timed* opacity for TAs: the system is opaque when an attacker can never deduce whether some secret sequence of actions (possibly with timestamps) was performed, by only observing a given set of observable actions together with their timestamp. It is then proved in [Cas09] that it is undecidable whether a TA is opaque. This notably relates to the undecidability of timed language inclusion for TAs [AD94]. The undecidability of opacity is strong: it holds even for the restricted class of *event-recording automata* [AFH99]—a subclass of TAs for which language inclusion is actually decidable.

The aforementioned negative result leaves hope only if the definition or the setting is changed, which was done in four main lines of work.

First, in [WZ18, WZA18], the input model is simplified to *real-time automata* [Dim01], a restricted formalism compared to TAs: real-time automata can be seen as TAs with a single clock, reset at each transition. [LLHL22] works on constant-time labeled automata, a subclass of real-time automata where events occur at constant values. In this setting, initial-state opacity ("according to the observations, what was the initial state?") and current-state opacity ("according to the observations, what is the current location?") become decidable. In [Zha24], Zhang studies labelled real-timed automata (a subclass of labelled TAs); in this setting, state-based (at the initial time, the current time, etc.) opacity is proved to be decidable by extending the observer (that is, the classical powerset construction) from finite automata to labelled real-timed automata.

Second, in [AGW+24, ADL24], the opacity was studied in the setting of Cassez' definition, but with restrictions in the model: one-clock automata, one-action automata, or over discrete time. Similarly, in [KKG24], discrete-time automata with several clocks are considered and transformed into tick automata in order to verify the current-state opacity. The discrete time setting yields decidability, while restricting the number of actions to 1 preserves undecidability; for a single clock, decidability can only be envisioned without silent actions [ADL24] (allowing silent actions or allowing two clocks immediately leads to undecidability).

Third, in [AETYM21], the authors consider a *time-bounded* notion of the opacity of [Cas09], where the attacker has to disclose the secret before a deadline, using a partial observability. This can be seen as a secrecy with an *expiration date*. The rationale is that retrieving a secret "too late" is useless; this is understandable, e.g., when the secret is the value in a cache; if the cache has been overwritten since, then knowing the secret is probably useless in most situations. In addition, the analysis is carried over a time-bounded horizon; this means there are two time bounds in [AETYM21]: one for the secret expiration date, and one for the bounded-time execution of the system. Deciding opacity in this setting is shown to be decidable for TAs.

Fourth, in [ALL+23], an alternative definition to Cassez' opacity is proposed, by studying ET-opacity (execution-time opacity): the attacker has only access to the *execution time* of the system, as opposed to Cassez' partial observations where some events (with their timestamps) are observable. The goal for the attacker is to deduce whether a special secret location was visited, by observing only the execution time. In that case, most problems for TAs become decidable, including some problems when introducing an expiration date [ALM23] (see [ALL+23] for a survey). Our current work fits in this ET-opacity context, with the additional goal to control the system.

Opacity in other formalisms

Different variants of opacity are also studied for other types of systems, such as stochastic systems. In this case, we can quantify the probability that a system is opaque [BMS15]. In particular, opacity can be related to the bandwidth of a language [JIDA22, ADDJI23] used to encode information by the delay necessary to produce it.

Non-interference in timed automata

Several works address non-interference for TAs. In this context, actions are either high-level or low-level, and only low-level actions are observable. A TA satisfies non-interference whenever

its behaviour in absence of high-level actions is equivalent to the observation of its behaviour when high-level actions occur. Different notions of equivalence (e.g., bisimulation) can be considered for this property. Several papers [BFST02, BT03, AK20] present some decidability results, while control is considered in [BCLR15].

General security problems for TAs are surveyed in [AA23].

Control

A preliminary version of control for ET-opacity in TAs was considered in [ABLM22], but only untimed, i.e., the actions could only be enabled or disabled once and for all, thus severely restricting the possibilities to render the system ET-opaque. In addition, [ALL+23] considers *parametric* versions of the opacity problems, in which timing parameters [AHV93] can be used in order to make the system ET-opaque. This parametric analysis was then used for the analysis of C code [ABC+25]. Our notion of control is orthogonal to parameter synthesis, as another way to ensure the system becomes ET-opaque.

Controller synthesis can be described and solved thanks to game theory; finding a strategy for a controller can be equivalent to computing a winning strategy in a corresponding game. Several game models have been considered, as timed games that can be used to solve synthesis problem on TAs. In this context, [AMPS98] aims to restrict the transition relation in order to satisfy certain properties, while [JT07] completes this result, minimizing the execution time, and [BFM15] studies the reachability with robust strategies only.

1.2 Contributions

In this work, we aim at tuning a system to make it ET-opaque, by *controlling* it at runtime.

Our attacker model is as follows: the attacker has a knowledge of the system model, but can only observe the execution time. This can correspond to an attacker observing the energy consumption of a device, clearly denoting the execution time of a program or process; or to an attacker observing communications over a shared network, with an observable message acknowledging the end of execution. The attacker aims at deducing—only by observing the execution time—whether a special secret location was visited.

As usual, we consider that the system actions are partitioned between controllable and uncontrollable. Our controller relies on the following notion of strategy: at each timestamp, the strategy enables only a subset of the controllable actions.

We mainly consider in this paper full ET-opacity, i.e., whenever the durations corresponding to executions visiting the secret location match the durations corresponding to executions *not* visiting the secret location. Our first contribution is to show that the full ET-opacity strategy emptiness problem, i.e., the emptiness of the set of strategies such that a TA is fully ET-opaque with such a strategy, is undecidable. Second, we move to a weaker version, that of *meta-strategies*, i.e., sets of strategies that specify a finite number of strategy changes per time unit, but not the precise time at which those changes take place. In that case, we show that not only the full ET-opacity meta-strategy emptiness problem is decidable in EXPSPACE, but our approach is also constructive, in the sense that we can build such a controller (in 2EXPTIME). Our technique relies on a novel *ad-hoc* construction inspired by the region automaton for TAs.

Depending of the system, various degrees of opacity can be interesting, and we therefore consider several variants introduced in [ALL+23] as our third contribution: in addition to full ET-opacity (the durations corresponding to executions visiting the secret location match the durations corresponding to executions *not* visiting the secret location), we also consider weak ET-opacity (the durations corresponding to executions visiting the secret location are *included* in the durations corresponding to executions not visiting the secret location), and we briefly discuss $\exists$-ET-opacity (in which we are simply interested in the *existence* of one execution time for which opacity is ensured). We show that, for both variants, the meta-strategy emptiness problem is decidable.

Finally, and as a fourth contribution, we address the case when the attacker cannot have an infinite precision in its observations, i.e., enhancing our method with a notion of robustness.

About this manuscript

This manuscript is an extended version of the paper published in the proceedings of SEFM 2024 [ADLL24]. We list the enhancements (and differences) with respect to the conference version:

1. We add the missing proofs of all results, and additional examples;
2. We correct the previous construction from [ADLL24], adding the notion of meta-strategy and, on the one hand, we prove the undecidability of the general problem of the existence of a strategy making the system fully ET-opaque while, on the other hand, we prove the decidability of the existence of a meta-strategy;
3. We investigate control with respect to weak ET-opacity and ∃-ET-opacity;
4. We investigate several robust definitions of ET-opacity and add details and proofs for results already stated in [ADLL24];
5. We rename "bad beliefs" into "leaking beliefs".

1.3 Outline

Section 2 recalls the necessary material. Section 3 defines the control problem for ET-opacity. Section 4 proves our main undecidability result. We then introduce the core of our decidable approach: Section 5 introduces the notion of belief automaton, while Section 6 solves the ET-opacity problems thanks to this notion. Section 7 then extends our approach to *existential* and *weak* opacity. The last contribution in Section 8 considers an attacker which cannot have an infinite precision in observing the execution time. Section 9 highlights future works.

2 Preliminaries

Let $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}_{\geq 0}$, $\mathbb{R}_{\geq 0}$, $\mathbb{R}_{>0}$, $\mathbb{R}_{<0}$ denote the sets of non-negative integer numbers, integer numbers, non-negative rational numbers, non-negative real numbers, positive real numbers and negative real numbers respectively.

Clock constraints

Clocks are real-valued variables that all evolve over time at the same rate. Throughout this paper, we assume a set $\mathbb{X} = \{x_1, \ldots, x_H\}$ of *clocks*. A *clock valuation* is a function $\mu : \mathbb{X} \to \mathbb{R}^H_{\geq 0}$, assigning a non-negative real value to each clock. Given $R \subseteq \mathbb{X}$, we define the reset of a valuation μ with respect to R, denoted by $[\mu]_R$, as follows: $[\mu]_R(x) = 0$ if $x \in R$, and $[\mu]_R(x) = \mu(x)$ otherwise. We write $\vec{0}$ for the clock valuation assigning 0 to all clocks. Given a constant $d \in \mathbb{R}_{\geq 0}$, $\mu + d$ denotes the valuation s.t. $(\mu + d)(x) = \mu(x) + d$, for all $x \in \mathbb{X}$.

We assume $\bowtie \in \{<, \leq, =, \geq, >\}$. A *constraint* C is a conjunction of inequalities over $\mathbb{X}$ of the form $x \bowtie d$, with $d \in \mathbb{Z}$.

A table of the notations used throughout this paper is available in Appendix A.

2.1 Timed automata

Syntax of TAs

We define timed automata as in [AD94], with an extra private location as in [ALL+23], which encodes the secret that shall not be leaked.

Definition 1 (Timed automaton)**.** A *timed automaton* (TA) $\mathcal{A}$ is a tuple $\mathcal{A} = (\Sigma, L, \ell_0, \ell_{priv}, F, \mathbb{X}, I, E)$ where:

1. Σ is a finite set of actions,
2. L is a finite set of locations,
3. $\ell_0 \in L$ is the initial location,
4. $\ell_{priv} \in L$ is the private location,
5. $F \subseteq L \setminus \{\ell_{priv}\}$ is the set of final locations,
6. $\mathbb{X} = \{x_1, \ldots, x_H\}$ is a finite set of clocks,
7. I is the invariant, assigning to every $\ell \in L$ a constraint $I(\ell)$,
8. E is a finite set of edges $\mathsf{e} = (\ell, g, a, R, \ell')$ where $\ell, \ell' \in L$ are the source and target locations, $a \in \Sigma \cup \{\varepsilon\}$, where ε denotes the silent action, $R \subseteq \mathbb{X}$ is a set of clocks to be reset, and g is a constraint over $\mathbb{X}$ (called *guard*).

□

Example 1. Fig. 1a depicts a TA $\mathcal{A}_1$ with a single clock x, where $\Sigma = \{a, b, u\}$. The edge e_1 between the initial location ℓ_0 and the private location ℓ_{priv} is available only when the valuation of x equals 0. The edge e_6 between ℓ_0 and ℓ_2 resets x.

□

Since we are only interested in the (first) arrival time in a final location, the following assumption does not restrict our framework, but simplifies the subsequent definitions and results.

Assumption 1. We consider every final location as *urgent* (where time cannot elapse): formally, there exists $x \in \mathbb{X}$ such that, for all $(\ell, g, a, R, \ell') \in$

$E, \ell' \in F$, we have $x \in R$ and "$x = 0$" $\in I(\ell')$. Moreover, final locations cannot have outgoing transitions: formally, there is no $(\ell, g, a, R, \ell') \in E$ s.t. $\ell \in F$.

Semantics of TAs

Definition 2 (Semantics of a TA)**.** Let $\mathcal{A} = (\Sigma, L, \ell_0, \ell_{priv}, F, \mathbb{X}, I, E)$ be a TA. The semantics of $\mathcal{A}$ is given by the timed transition system $TTS_{\mathcal{A}} = (S, s_0, \Sigma \cup \mathbb{R}_{\geq 0}, \delta)$, with

1. $S = \{(\ell, \mu) \in L \times \mathbb{R}^H_{\geq 0} \mid \mu \models I(\ell)\}$,
2. $s_0 = (\ell_0, \vec{0})$,
3. δ consists of the discrete and (continuous) delay transition relations:
   (a) discrete transitions: $(\ell, \mu) \overset{\mathsf{e}}{\mapsto} (\ell', \mu')$, if $(\ell, \mu), (\ell', \mu') \in S$, and there exists $\mathsf{e} = (\ell, g, a, R, \ell') \in E$, such that $\mu' = [\mu]_R$, and $\mu \models g$.
   (b) delay transitions: $(\ell, \mu) \overset{d}{\mapsto} (\ell, \mu+d)$, with $d \in \mathbb{R}_{\geq 0}$, if $\forall d' \in [0, d], (\ell, \mu + d') \in S$.

□

We write $(\ell, \mu) \xrightarrow{d,\mathsf{e}} (\ell', \mu')$ for a combination of a delay and a discrete transitions when $\exists \mu'' : (\ell, \mu) \overset{d}{\mapsto} (\ell, \mu'') \overset{\mathsf{e}}{\mapsto} (\ell', \mu')$.

Given a TA $\mathcal{A}$ with semantics $(S, s_0, \Sigma \cup \mathbb{R}_{\geq 0}, \delta)$, a *run* of $\mathcal{A}$ is a finite alternating sequence of states of $TTS_{\mathcal{A}}$ and pairs of delays and edges starting from the initial state s_0 of the form $s_0, (d_0, \mathsf{e}_0), s_1, \ldots, s_n$ where for all $i < n, \mathsf{e}_i \in E, d_i \in \mathbb{R}_{\geq 0}$ and $s_i \xrightarrow{d_i,\mathsf{e}_i} s_{i+1}$. The duration of a run $\rho = s_0, (d_0, \mathsf{e}_0), s_1, \ldots, (d_{n-1}, \mathsf{e}_{n-1}), s_n$ is $dur(\rho) = \sum_{0 \leq i \leq n-1} d_i$. We define $last(\rho) = s_n$.

Extra clock

We will need an extra clock z that will help us later to keep track of the elapsed absolute time. This clock is reset exactly every 1 time unit, and therefore each reset corresponds to a "tick" of the absolute time. (Note that its actual value remains in $[0, 1]$ and therefore always matches the fractional part of the absolute time.) In all subsequent region constructions, we assume the existence of $z \in \mathbb{X}$. For each location ℓ, we add the constraint "$z \leq 1$" to $I(\ell)$, and we add a self-loop edge $(\ell, z = 1, \varepsilon, \{z\}, \ell)$.

2.2 Regions

Given a TA $\mathcal{A}$, for a clock x_i, we denote by c_i the largest constant to which x_i is compared within the guards and invariants of $\mathcal{A}$: formally, $c_i = \max_j\{d_j \mid x_i \bowtie d_j$ appears in a guard or invariant of $\mathcal{A}\}$. Given a clock valuation μ and a clock x_i, $\lfloor \mu(x_i) \rfloor$ (resp. $\mathrm{fr}(\mu(x_i))$) denotes the integral (resp. fractional) part of $\mu(x_i)$.

We now recall the equivalence relation between clock valuations.

Definition 3 (Equivalence relation [AD94])**.** Two clocks valuations μ, μ' are *equivalent*, denoted by $\mu \approx \mu'$, when the following three conditions hold for any clocks $x_i, x_j \in \mathbb{X}$:

1. $\lfloor \mu(x_i) \rfloor = \lfloor \mu'(x_i) \rfloor$ or $\mu(x_i) > c_i$ and $\mu'(x_i) > c_i$;
2. if $\mu(x_i) \leq c_i$ and $\mu(x_j) \leq c_j$: $\mathrm{fr}(\mu(x_i)) \leq \mathrm{fr}(\mu(x_j))$ iff $\mathrm{fr}(\mu'(x_i)) \leq \mathrm{fr}(\mu'(x_j))$; and
3. if $\mu(x_i) \leq c_i$: $\mathrm{fr}(\mu(x_i)) = 0$ iff $\mathrm{fr}(\mu'(x_i)) = 0$.

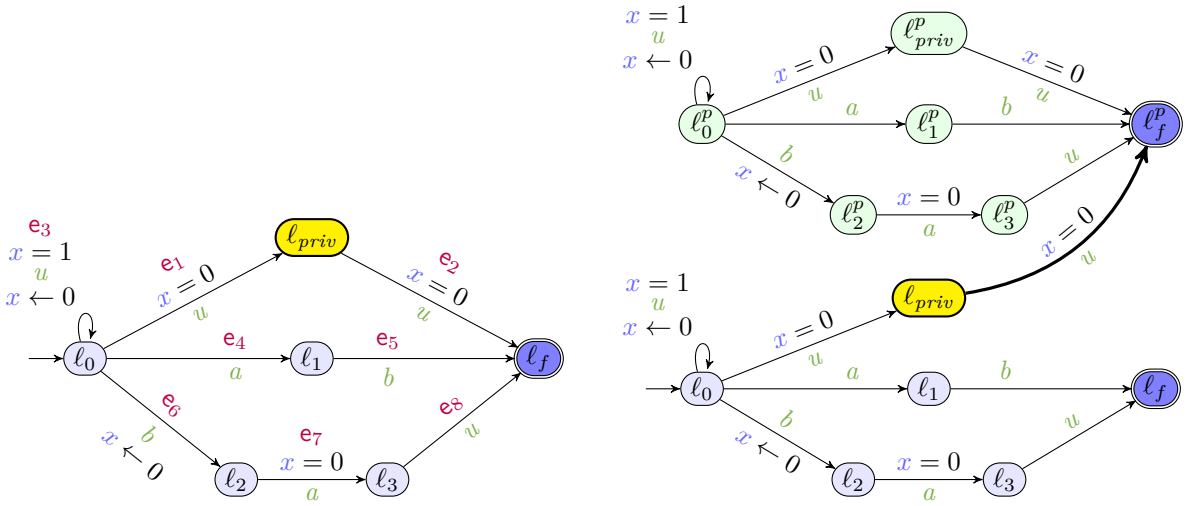
□

In other words, two valuations are equivalent when, for a given clock, the integral part is the same in both valuations or greater than the largest constant in both valuations (Item 1), for any two clocks, the fractional parts are in the same order in the two valuations (Item 2), and the fractional part is zero in both valuations or neither (Item 3).

The equivalence relation $\approx$ is extended to the states of $TTS_{\mathcal{A}}$: given two states $s = (\ell, \mu), s' = (\ell', \mu')$ of $TTS_{\mathcal{A}}$, we write $s \approx s'$ iff $\ell = \ell'$ and $\mu \approx \mu'$. We denote by $[s]$ and call *region* the equivalence class of a state s for $\approx$. Then, $s' \in [s]$ when $s \approx s'$. The set of all regions of $\mathcal{A}$ is denoted $\mathsf{R}_{\mathcal{A}}$. A region $r = [(\ell, \mu)]$ is *final* whenever $\ell \in F$. The set of final regions is denoted by $\mathsf{R}^F_{\mathcal{A}}$. A region r is *reachable* when there exists a run ρ such that $last(\rho) \in r$.

Region automaton

We now define a region automaton inspired by [BDR08, Proposition 5.3] with two-component labels on the transitions : the first component indicates how the fractional part of z evolved, with symbol "0" corresponding to an absence of change, symbol "0^+" corresponding to a change remaining in $(0, 1)$ and symbol "1" corresponding to a change where the fractional part either starts or ends at 0. The second component is either ε if the transition represents time elapsing in the TA, or



(a) TA $\mathcal{A}_1$ (b) Duplicated version of $\mathcal{A}_1$

Fig. 1: A TA and its duplicated version (introduced in Section 5)

provides the action that labels the corresponding discrete transition in the TA.

Given a state $s = (\ell, \mu)$, and $d \in \mathbb{R}_{\geq 0}$, we write $s + d$ to denote $(\ell, \mu + d)$. Given two regions r and r', we write $r \cup r'$ for $\{s \mid s \in r \text{ or } s \in r'\}$.

Definition 4 (Labelled Region Automaton)**.** For a given TA $\mathcal{A}$, the labelled region automaton $\mathcal{R}_{\mathcal{A}}$ is given by the tuple $(\mathsf{R}_{\mathcal{A}}, \Sigma^{\mathsf{R}}, \delta^{\mathsf{R}})$ where:

1. $\mathsf{R}_{\mathcal{A}}$ is the set of states,
2. $\Sigma^{\mathsf{R}} = \{0, 0^+, 1\} \times (\Sigma \cup \{\varepsilon\})$,
3. given two regions $r, r' \in \mathsf{R}_{\mathcal{A}}$ and $\zeta \in \Sigma^{\mathsf{R}}$, we have $(r, \zeta, r') \in \delta^{\mathsf{R}}$ if there exist $s = (\ell, \mu) \in r$ and $s' = (\ell', \mu') \in r'$ such that one of the following holds:
   (a) $\zeta = (0, a)$ and $(\ell, \mu) \overset{\mathsf{e}}{\mapsto} (\ell', \mu') \in \delta$ in $TTS_{\mathcal{A}}$ with $\mathsf{e} = (\ell, g, a, R, \ell')$ for some g and R;
   (b) $\zeta = (0^+, \varepsilon)$ and $\exists d \in \mathbb{R}_{>0}$ such that
      (i) $s \overset{d}{\mapsto} s'$,
      (ii) $\forall 0 < d' < d$, $s + d' \in r \cup r'$ and
      (iii) $\mathrm{fr}(\mu(z)) \neq 0$ and $\mathrm{fr}(\mu'(z)) \neq 0$;[1]
   (c) $\zeta = (1, \varepsilon)$ and $\exists d \in \mathbb{R}_{>0}$ such that
      (i) $s \overset{d}{\mapsto} s'$,
      (ii) $\forall 0 < d' < d$, $s + d' \in r \cup r'$, and
      (iii) $\mathrm{fr}(\mu'(z)) = 0$ iff $\mathrm{fr}(\mu(z)) \neq 0$.

[1] Condition (ii) ensures that we only move from one region to the "next" one (no intermediate region), and condition (iii) adds that we stay in the same region for z (changing the region for z is handled in item (c)).

□

We write $r \xrightarrow{\zeta}_{\mathsf{R}} r'$ for $(r, \zeta, r') \in \delta^{\mathsf{R}}$.

In the remainder, we will refer to labelled region automata as region automata to alleviate notation.

2.3 Execution-time opacity of a TA

Let us now recall from [ALL+23] the notions of private and public runs.

Durations

Given a TA $\mathcal{A}$ and a run ρ, we say that ℓ_{priv} is *visited on the way to a final location in* ρ when ρ is of the form $(\ell_0, \mu_0), (d_0, \mathsf{e}_0), (\ell_1, \mu_1), \ldots, (\ell_m, \mu_m), (d_m, \mathsf{e}_m), \ldots, (\ell_n, \mu_n)$ for some $m, n \in \mathbb{N}$ such that $\ell_m = \ell_{priv}$ and $\ell_n \in F$. We denote by $Visit^{priv}(\mathcal{A})$ the set of those runs, and refer to them as *private* runs. We denote by $DVisit^{priv}(\mathcal{A})$ the set of all the durations of these runs.

Conversely, we say that ℓ_{priv} is *avoided on the way to a final location in* ρ when ρ is of the form $(\ell_0, \mu_0), (d_0, \mathsf{e}_0), (\ell_1, \mu_1), \ldots, (\ell_n, \mu_n)$ with $\ell_n \in F$ and $\forall 0 \leq i < n, \ell_i \neq \ell_{priv}$. We denote the set of those runs by $Visit^{pub}(\mathcal{A})$, referring to them as *public* runs, and by $DVisit^{pub}(\mathcal{A})$ the set of all the durations of these public runs.

These concepts can be seen as the set of execution times from the initial location ℓ_0 to a

final location while visiting (resp. not visiting) the private location ℓ_{priv}.

Example 2. Consider the following two runs of the TA $\mathcal{A}_1$ in Fig. 1a. Note that we use $(\ell_0, \cdot)$ as a shortcut for (ℓ_0, μ) such that $\mu(x) = \cdot$.

$$\rho_1 = (\ell_0, 0), (1, \mathsf{e}_3), (\ell_0, 0), (0, \mathsf{e}_1), (\ell_{priv}, 0), (0, \mathsf{e}_2), (\ell_f, 0)$$

$$\rho_2 = (\ell_0, 0), (0.1, \mathsf{e}_6), (\ell_2, 0), (0, \mathsf{e}_7), (\ell_3, 0), (0.8, \mathsf{e}_8), (\ell_f, 0.8)$$

Run $\rho_1 \in \mathit{Visit}^{priv}(\mathcal{A}_1)$ is a private run, and $\mathit{dur}(\rho_1) = 1 \in \mathit{DVisit}^{priv}(\mathcal{A}_1)$. Run $\rho_2 \in \mathit{Visit}^{pub}(\mathcal{A}_1)$ is a public run with $\mathit{dur}(\rho_2) = 0.9 \in \mathit{DVisit}^{pub}(\mathcal{A}_1)$. □

Definition 5 (Full ET-opacity)**.** A TA $\mathcal{A}$ is *fully ET-opaque* when $\mathit{DVisit}^{priv}(\mathcal{A}) = \mathit{DVisit}^{pub}(\mathcal{A})$. □

That is, if for any run of duration d reaching a final location after visiting ℓ_{priv}, there exists another run of the same duration reaching a final location but not visiting ℓ_{priv}, and vice versa, then the TA is fully ET-opaque.

Example 3. Consider again $\mathcal{A}_1$ in Fig. 1a. Each time x equals 1, we can reset it via e_3, and take the edges e_1 and e_2 instantaneously. It results that $\mathit{DVisit}^{priv}(\mathcal{A}_1) = \mathbb{N}$. We have seen in Example 2 that $0.9 \in \mathit{DVisit}^{pub}(\mathcal{A}_1)$. So $\mathit{DVisit}^{priv}(\mathcal{A}_1) \neq \mathit{DVisit}^{pub}(\mathcal{A}_1)$ and $\mathcal{A}_1$ is not fully ET-opaque. □

3 Problem: Controlling TA to achieve ET-opacity

Let us formally define the main problem addressed in this work. We assume $\Sigma = \Sigma_c \uplus \Sigma_u$ where Σ_c (resp. Σ_u) denotes controllable (resp. uncontrollable) actions. The uncontrollable actions are always available, whereas the controllable actions can be enabled and disabled at runtime.

The controller has a *strategy*, i.e., a function $\sigma : \mathbb{R}_{\geq 0} \to 2^{\Sigma_c}$ which associates to each time a subset of Σ_c, denoting that these actions are enabled, while other controllable actions are disabled.

We define the semantics of a controlled TA as follows. Compared to Definition 2, we also add to the states the absolute time.

Definition 6 (Semantics of a controlled TA)**.** Given a TA $\mathcal{A} = (\Sigma, L, \ell_0, \ell_{priv}, F, \mathbb{X}, I, E)$ and a strategy $\sigma : \mathbb{R}_{\geq 0} \to 2^{\Sigma_c}$, the *semantics of the TA $\mathcal{A}$ controlled by strategy σ* is given by $(S, s_0, \Sigma \cup \mathbb{R}_{\geq 0}, \delta^\sigma)$ with

1. $S = \{(\ell, \mu, t) \in L \times \mathbb{R}^H_{\geq 0} \times \mathbb{R}_{\geq 0} \mid \mu \models I(\ell)\}$,
2. $s_0 = (\ell_0, \vec{0}, 0)$,
3. δ^σ consists of the discrete and (continuous) delay transition relation:
   (a) discrete transitions: $(\ell, \mu, t) \overset{\mathsf{e}}{\mapsto}_\sigma (\ell', \mu', t)$, if $(\ell, \mu, t), (\ell', \mu', t) \in S$ and there exists $\mathsf{e} = (\ell, g, a, R, \ell') \in E$ such that $\mu' = [\mu]_R$, $\mu \models g$, and $a \in \sigma(\tau) \cup \Sigma_u$ (that is, a is either enabled by the strategy at time τ, or uncontrollable);
   (b) delay transitions: $(\ell, \mu, t) \overset{d}{\mapsto}_\sigma (\ell, \mu + d, t + d)$, with $d \in \mathbb{R}_{\geq 0}$, if $\forall d' \in \mathbb{R}_{>0}$ such that $d' < d$, $(\ell, \mu + d', t + d') \in S$.

□

We write $(\ell, \mu, t) \xrightarrow{d,e}_\sigma (\ell', \mu', t')$ for a combination of a delay and a discrete transitions when $\exists \mu''$ such that $(\ell, \mu, t) \overset{d}{\mapsto}_\sigma (\ell, \mu'', t') \overset{\mathsf{e}}{\mapsto}_\sigma (\ell', \mu', t')$.

A run $\rho = (\ell_0, \mu_0), (d_0, \mathsf{e}_0), \ldots, (\ell_n, \mu_n)$ is *σ-compatible* when, $\forall 0 \leq i < n$, it holds that $(\ell_i, \mu_i, \sum_{j<i} d_j) \xrightarrow{d_i, \mathsf{e}_i}_\sigma (\ell_{i+1}, \mu_{i+1}, \sum_{j \leq i} d_j)$.

We let $\mathit{Visit}^{priv}_\sigma(\mathcal{A})$ the set of private and σ-compatible runs, $\mathit{Visit}^{pub}_\sigma(\mathcal{A})$ the set of public and σ-compatible runs, $\mathit{DVisit}^{priv}_\sigma(\mathcal{A})$ the set of durations of private and σ-compatible runs, and $\mathit{DVisit}^{pub}_\sigma(\mathcal{A})$ the set of durations of public and σ-compatible runs.

Definition 7 (Full ET-opacity with strategy)**.** Given a strategy σ, a TA $\mathcal{A}$ is *fully ET-opaque with σ* whenever $\mathit{DVisit}^{priv}_\sigma(\mathcal{A}) = \mathit{DVisit}^{pub}_\sigma(\mathcal{A})$. □

Example 4 (ET-opaque TA)**.** Consider the TA $\mathcal{A}_{opaque}$ in Fig. 2. Assume $\Sigma_u = \{u\}$ and $\Sigma_c = \{a\}$. First consider the strategy σ_1 such that $\forall \tau \in \mathbb{R}_{\geq 0}, \sigma_1(\tau) = \{a\}$, i.e., a is allowed anytime. We have $\mathit{DVisit}^{priv}_{\sigma_1}(\mathcal{A}) = \mathbb{N}$ while $\mathit{DVisit}^{pub}_{\sigma_1}(\mathcal{A}) =$

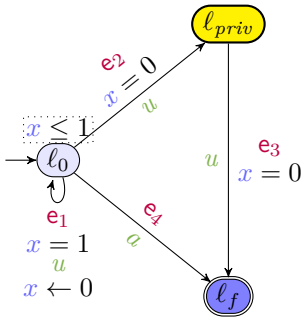


Fig. 2: TA $\mathcal{A}_{opaque}$

$\mathbb{R}_{\geq 0}$. Therefore $DVisit^{priv}_{\sigma_1}(\mathcal{A}) \neq DVisit^{pub}_{\sigma_1}(\mathcal{A})$, and hence $\mathcal{A}_{opaque}$ is not fully ET-opaque with σ_1.

Now consider the strategy σ_2 such that

$$\sigma_2(\tau) = \begin{cases} \{a\} & \text{if } \tau \in \mathbb{N} \\ \emptyset & \text{otherwise.} \end{cases}$$

We now have $DVisit^{priv}_{\sigma_2}(\mathcal{A}) = DVisit^{pub}_{\sigma_2}(\mathcal{A}) = \mathbb{N}$, hence $\mathcal{A}_{opaque}$ is fully ET-opaque with σ_2. □

Example 5 (Non-ET-opaque TA)**.** There is no strategy such that TA $\mathcal{A}_1$ in Fig. 1a is fully ET-opaque. Recall that $\Sigma_u = \{u\}$ and $\Sigma_c = \{a, b\}$. The transitions e_1 and e_2 are uncontrollable, so we can reach ℓ_f at any integer time along a run visiting ℓ_{priv}. However the set of public durations will either not contain 0, or contain $\mathbb{R}_{\geq 0}$. Indeed, in order to reach ℓ_f with a public run, the system must take transitions associated to both actions a and b. As a consequence, to contain the duration 0, the strategy must allow $\{a, b\}$ at time 0. Hence, ℓ_3 can be reached at time 0, and as the next transition e_8 is uncontrollable, it can be taken at any time. Thus, the set of public durations is $\mathbb{R}_{\geq 0}$. □

Finitely-varying strategies

In the following, to match the fact that a meta-strategy has a finite number of strategy changes between two integer time instants, we only consider strategies that behave in a "reasonable" way. We thus only consider *finitely-varying strategies*, in which the number of changes are finite for any closed time interval.

Indeed, we can assume that a controller cannot change infinitely frequently its strategy in a finite time: it is unrealistic to consider, in a bounded interval, neither a system that can perform an infinite number of actions, nor a controller that can make an infinite number of choices. Finitely-varying strategies are reminiscent of non-Zeno behaviours, in the sense that finitely-varying strategies have a finite number of strategy changes in every bounded interval.

Formally:

Definition 8 (Finitely-varying strategy)**.** A strategy σ is *finitely-varying* whenever, for any closed time interval $\mathcal{I}$, there is a finite partition $\iota_1, \ldots, \iota_n$ of $\mathcal{I}$ such that each ι_i is an interval within which σ makes the same choice. That is, for all $\tau_1, \tau_2 \in \iota_i$, $1 \leq i \leq n$, $\sigma(\tau_1) = \sigma(\tau_2)$. □

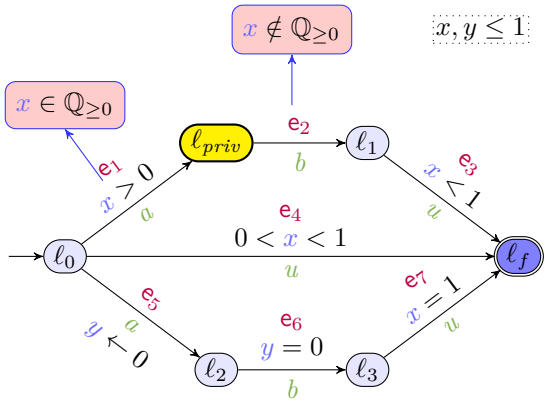


Fig. 3: Automaton $\mathcal{A}_{nfv}$

Example 6 (Non-finitely-varying strategy)**.** Let $\mathcal{A}_{nfv}$ be the automaton in Fig. 3, with a global invariant $x, y \leq 1$. Let $\Sigma_u = \{u\}$ and $\Sigma_c = \{a, b\}$. Since transition e_4 is uncontrollable, for any strategy σ, $(0, 1) \subseteq DVisit^{pub}_{\sigma}(\mathcal{A})$. To ensure a fully ET-opaque system, all those durations need to be also enabled through the private state. This implies that, for every $\alpha < 1$, both a and b should be enabled at a time instant before α. Now if we consider the bottom path, if a and b are enabled simultaneously, then state ℓ_3 is reachable and it is possible to reach the final state when $x = 1$. Then we need to prevent b to be enabled at the same time as a. We can build a strategy σ to make $\mathcal{A}_{nfv}$ fully ET-opaque with σ. This strategy (illustrated in red boxes in Fig. 3) is defined by

$$\sigma(\tau) = \begin{cases} \{a\} & \text{if } \tau \in \mathbb{Q}_{\geq 0} \\ \{b\} & \text{if } \tau \notin \mathbb{Q}_{\geq 0} \end{cases}$$

With strategy σ, it means that the order in which actions a and b can be performed is not fixed, while preventing them from being performed at the same time. This is only possible with a non-finitely-varying strategy, where the number of changes is infinite in a given interval. □

Meta-strategies

As defined previously, strategies have infinite precision in determining when to activate or deactivate controllable actions. While it might seem reasonable to assume precise knowledge of when a clock tick occurs, in practice, achieving such infinite precision between integer time points is not

feasible. Therefore, the idea is to group together all strategies that enable the same actions at exact integer times and permit subsets of these actions—maintaining the same order—between those times.

Definition 9 (Meta-strategy)**.** A meta-strategy ϕ is a partial function on integer bounded intervals, such that:

- for all $n \in \mathbb{N}$, $\phi([n,n]) \in 2^{\Sigma_c}$,
- for all $n \in \mathbb{N}$, $\phi((n,n+1)) \in (2^{\Sigma_c})^*$

□

In other words, a meta-strategy is a function which associates to each integer time a set $\nu \in 2^{\Sigma_c}$ of enabled actions, and to each open interval between two consecutive integers a finite sequence $(\nu_1, \ldots, \nu_m) \in (2^{\Sigma_c})^*$ giving the order in which sets of actions are allowed in the system.

Definition 10 (Ordered partition)**.** Given an interval $\mathcal{I}$, we call *ordered partition of* $\mathcal{I}$ a finite sequence of disjoint intervals $\iota_1, \ldots, \iota_n$ such that:

1. $\bigcup_{j=1}^{n} \iota_j = \mathcal{I}$,
2. ι_1 is left open, ι_n is right open,
3. each ι_j is non empty, and
4. for all $j \in \{1, \ldots, n-1\}$, the right boundary of ι_j is the left boundary of ι_{j+1}.

□

Example 7. The sequence $(0, 0.2], (0.2, 0.42), [0.42, 0.42], (0.42, 0.83], (0.83, 1)$ is an ordered partition of the interval $\mathcal{I} = (0, 1)$. □

Definition 11 (Meta-strategy satisfaction)**.** A strategy σ is said to *satisfy* a meta-strategy ϕ, denoted $\sigma \models \phi$, when:

- for all $n \in \mathbb{N}$, $\sigma(n) = \phi([n,n])$,
- for all $n \in \mathbb{N}$, denoting $\mathcal{I} = (n, n+1)$ and $\phi(\mathcal{I}) = \nu_1, \ldots, \nu_m$ there is an ordered partition $\iota_1, \ldots, \iota_m$ of $\mathcal{I}$ such that for all $\tau \in \iota_i$, $1 \leq i \leq m$, $\sigma(\tau) = \nu_i$.

□

The set of private (or public) durations for a meta-strategy is defined as the union of private (or public) durations of all the strategies it represents: $DVisit_{\phi}^{priv}(\mathcal{A}) = \{\tau \mid \exists \sigma \text{ s.t. } \sigma \models \phi \text{ and } \tau \in DVisit_{\sigma}^{priv}(\mathcal{A})\}$ and $DVisit_{\phi}^{pub}(\mathcal{A}) = \{\tau \mid \exists \sigma \text{ s.t. } \sigma \models \phi \text{ and } \tau \in DVisit_{\sigma}^{pub}(\mathcal{A})\}$.

Definition 12 (Full ET-opacity with a meta-strategy)**.** Given a meta-strategy ϕ, a TA $\mathcal{A}$ is *fully ET-opaque with* ϕ whenever $DVisit_{\phi}^{priv}(\mathcal{A}) = DVisit_{\phi}^{pub}(\mathcal{A})$. □

Definitions 8 and 9 immediately give a correspondence between finitely-varying strategies and meta-strategies:

Lemma 1. *For any finitely-varying strategy* σ*, there exists a unique meta-strategy* ϕ *such that* $\sigma \models \phi$*.*

For any meta-strategy ϕ*, there exists a (finitely-varying) strategy* σ *such that* $\sigma \models \phi$*.*

Problems

In this paper, we are interested in several ET-opacity control problems, i.e., related to a (meta-)strategy making the corresponding controlled TA $\mathcal{A}$ ET-opaque.

The first problem we are interested in will be the *existence* of a strategy enforcing the ET-opacity.

Full ET-opacity strategy emptiness problem:
INPUT: A TA $\mathcal{A}$
PROBLEM: Decide the emptiness of the set of strategies σ such that $\mathcal{A}$ is fully ET-opaque with σ.

Because we will show that this problem is undecidable (Theorem 1), even when restricted to finitely-varying strategies, we define a variant of the setting, and consider the existence of meta-strategies instead. We will consider both the *existence* of a meta-strategy enforcing the ET-opacity, or the *synthesis* of such a meta-strategy.

Full ET-opacity meta-strategy emptiness problem:
INPUT: A TA $\mathcal{A}$
PROBLEM: Decide the emptiness of the set of meta-strategies ϕ such that $\mathcal{A}$ is fully ET-opaque with ϕ.

Full ET-opacity meta-strategy synthesis problem:
INPUT: A TA $\mathcal{A}$
PROBLEM: Synthesize a meta-strategy ϕ such that $\mathcal{A}$ is fully ET-opaque with ϕ.

4 Undecidability of the full ET-opacity strategy emptiness problem

In this section, we show that the full ET-opacity strategy emptiness problem is undecidable, vindicating the reliance on meta-strategies. To do so, we will use a reduction from the termination of the two-counter Minsky machine problem—which is known to be undecidable [Min67].

Let us start with a quick recall of Minsky machines. A two-counter Minsky machine $\mathcal{M}$ is described by two counters C_1 and C_2, as well as a sequence of commands $c_0, \ldots, c_m$ where c_0 is the starting command, c_m ends the run of the system, and every command c_0 to c_{m-1} is of one of the following three types:

- increment counter $C \in \{C_1, C_2\}$, move to the next command
- decrement counter $C \in \{C_1, C_2\}$, move to the next command (note that the machine must be designed so that this command cannot occur when counter C is equal to 0)
- if counter $C \in \{C_1, C_2\}$ is equal to 0, move to command c_k, otherwise move to command c_j.

In summary, the machine goes through a list of commands starting with c_0, incrementing, decrementing counters, or testing whether a counter is equal to 0 in order to select the new command to jump to—and it terminates whenever it reaches c_m. The *termination problem* for two-counter Minsky machines consists in deciding, given a machine $\mathcal{M}$, whether $\mathcal{M}$ terminates.

Theorem 1. *The full ET-opacity strategy emptiness problem is undecidable.*

Proof. Let $\mathcal{M}$ be a Minsky machine over two counters C_1 and C_2, described by the commands $c_0, \ldots, c_m$. We will build a TA $\mathcal{A}$ such that there exists a strategy σ enforcing full ET-opacity of $\mathcal{A}$ iff $\mathcal{M}$ does *not* terminate.

Overall intuition of the encoding

Intuitively, in order to be opaque, the TA $\mathcal{A}$ coupled with a strategy σ will have to correctly emulate the behaviour of the Minsky machine, with the strategy's choices depending on the values of the counters. More precisely, each command of $\mathcal{M}$ will take exactly three time units to be handled by $\mathcal{A}$, and therefore the i-th step of the machine corresponds to the interval $(3i, 3i+3]$. Considering this interval modulo 3, $(0,1]$ is dedicated to actions associated to counter C_1, $(1,2]$ is dedicated to actions associated to counter C_2, and $(2,3]$ is used to detect whether the strategy makes the system fully ET-opaque; in particular, we have that no private run can reach the target within this period of time. Due to the periodic nature of the system, the intervals $(0,1]$, $(1,2]$ and $(2,3]$ should be understood modulo 3.

Let us give an idea of how counter C_1 is represented within the first interval. More detailed explanation as well as the corresponding $\mathcal{A}$ used are given later in this section. In our model, whenever C_1 has the value k, then k public runs will reach the final location at different times during this interval. In order to ensure opacity, the strategy needs to allow the action a_{C_1} at those k instants, as this action produces a private run that immediately reaches the final state. However it also produces a public run that will reach the final state three time units later. Hence, the number of public runs reaching the final location during the next $(0,1]$ interval remains the same (ignoring the potential impact of other actions), thus preserving the value of the counter as well.

Incrementing the counter can then be done by forcing an action which will add a new public duration three times unit later, while decrementing the counter is done by producing a private run immediately reaching the target, and thus removing the need for one a_{C_1}. The test command can be ensured by first requiring, at time 0, the controller to claim (allowing either the action $=_0$ or $\neq_0$) whether counter C_1 is zero or not. Then, by observing whether action a_{C_1} occurs or not within the following interval.

The actions on counter C_2 are the same, only occurring within the intervals $(1,2]$ and thus will not be detailed fully in the proof.

Finally, if a violation is made (for instance by claiming the counter is zero when it is not, or by refusing to play an action linked to an increment or decrement of a counter), a location can be reached from which one can go to the final location at any time. This means that any duration beyond this point corresponds to a public run. This violates opacity as, as mentioned earlier, no private run will be able to reach the final location within the intervals $(2,3]$. Reaching the

final command c_m will similarly trigger a violation of opacity, hence the only way for a strategy to ensure opacity is to properly emulate the Minsky machine but fail to reach c_m, and thus for the Minsky machine not to terminate.

Gadgets and actions

Our TA will be built by combining several smaller timed automata fragments called "gadgets". We first describe the construction gadget per gadget, and then explain how they should be combined. As the strategy's choice is based only on time elapsed, it does not know in which gadget the run is, and thus must assume it might be in any of them, ensuring opacity in all cases.

For all the gadgets, we fix a set of actions $\Sigma = \{u, a_{C_1}, a_{C_2}, inc_{C_1}, inc_{C_2}, dec_{C_1}, dec_{C_2}, =_0, \neq_0\}$ with u being the only uncontrollable action. The TA $\mathcal{A}$, and a fortiori the different gadgets, will rely on a single clock x. In particular, we do not rely on the extra "tick" clock z that is assumed throughout the rest of this document.

Gadget 1act

We first describe a gadget preventing the controller from allowing two actions simultaneously. Formally, the gadget *1act* is the TA (see Fig. 4) $\mathcal{A}_{1act} = (\Sigma, L_{1act}, \ell_0^{1act}, \ell_{priv}, \ell_f, \{x\}, I_{1act}, E_{1act})$ where:

- $L_{1act} = \{\ell_0^{1act}, \ell_e, \ell_f\} \cup \{\ell_v \mid v \in \Sigma\}$,
- $I_{1act}(\ell) = true$ for all $\ell \in L_{1act}$,
- $E_{1act} = \big\{(\ell_0^{1act}, true, v, \{x\}, \ell_v) \mid v \in \Sigma\big\} \cup \big\{(\ell_v, x = 0, v', \emptyset, \ell_e) \mid v, v' \in \Sigma, v \neq v'\big\} \cup \big\{(\ell_e, true, u, \emptyset, \ell_f\big\}$.

In the *1act* gadget, whenever an action v is allowed, the system can go to a location ℓ_v via e_1, resetting x; then, if any action other than v is also allowed at the same time (which is tested by requesting $x = 0$ in edge e_2), then the following location ℓ_e can be reached. From ℓ_e the system can reach the final location at any time via e_3 via the uncontrollable action u, whatever the controller does. As a consequence, every duration beyond this point corresponds to a public run. As previously mentioned, some durations (the intervals $(2, 3]$) cannot be achieved by private runs, and therefore opacity is violated.

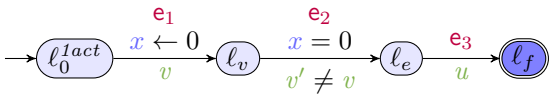


Fig. 4: *1act* gadget on a generic action v

Gadgets G_{C_1} and G_{C_2}

We now introduce the gadget G_{C_1} (resp. G_{C_2}) which forces, barring intervention by other gadgets, the strategy to repeat the same behaviour within the intervals $(0, 1)$ (resp. $(1, 2)$). Formally, the gadget G_{C_1} is the TA (see Fig. 5a) $\mathcal{A}_{G_{C_1}} = (\Sigma, L_{G_{C_1}}, \ell_0^{G_{C_1}}, \ell_{priv}, \ell_f, \{x\}, I_{G_{C_1}}, E_{G_{C_1}})$ where:

- $L_{G_{C_1}} = \{\ell_0^{G_{C_1}}, \ell_{C_1}, \ell_{priv}, \ell_f\}$,
- $I_{G_{C_1}}(\ell) = true$ for all $\ell \in L_{G_{C_1}}$,
- $E_{G_{C_1}} = \big\{(\ell_0^{G_{C_1}}, x = 3, u, \{x\}, \ell_0^{G_{C_1}}),$ $(\ell_0^{G_{C_1}}, 0 < x < 1, a_{C_1}, \{x\}, \ell_{C_1}),$ $(\ell_{C_1}, x = 0, u, \emptyset, \ell_{priv}),\ (\ell_{priv}, x = 0, u, \emptyset, \ell_f),$ $(\ell_{C_1}, x = 3, u, \emptyset, \ell_f)\big\}$.

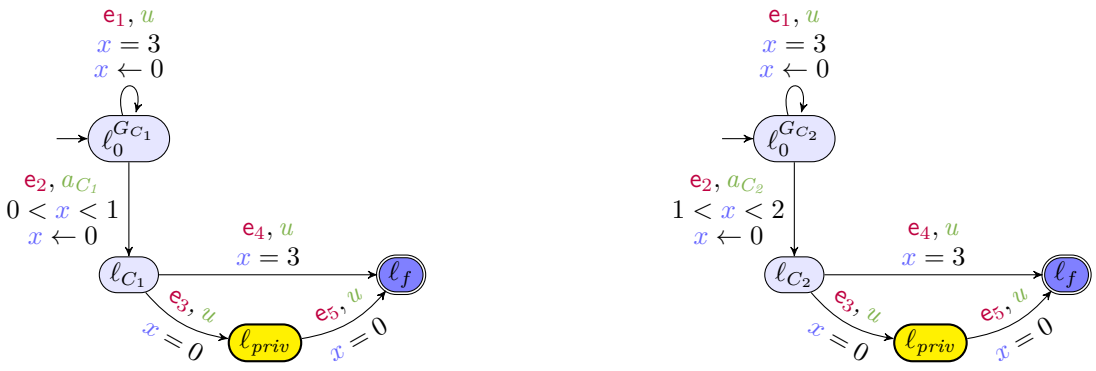
Gadget G_{C_2} only differs by one transition (see Fig. 5b), moving the impact of allowing the action a_{C_2} to the interval $(1, 2)$. Note that, since the intervals in which a_{C_1} and a_{C_2} have an effect are disjoint, we could use a single action for both. We keep two for ease of understanding. In the following, we will not present the variations associated to counter C_2.

Assume that public runs are expected to reach the target at times $\tau_1, \ldots, \tau_k$ within the interval $(0, 1)$ of this step. Ignoring future gadgets, thanks to G_{C_1}, the strategy can preserve opacity by allowing a_{C_1} exactly at times $\tau_1, \ldots, \tau_k$. This immediately produce a private run (going via e_2 then e_3). However, it also produces a public run that will reach the final location 3 time units later (via e_4), hence forcing the strategy to redo the same choices during the next step.

As the number of times a_{C_1} is allowed during one step of the process represents the value of counter C_1, barring external intervention, G_{C_1} allows to maintain the value of the counter within the strategy.

Increment gadget

We now show how external interventions modify the number of times the strategy must repeat a_{C_1} within the interval $(0, 1)$. We start with the gadget corresponding to a command incrementing



(a) Gadget G_{C_1} (b) Gadget G_{C_2}

Fig. 5: Gadgets G_{C_1} and G_{C_2}

counter C_1. More precisely, if command c_i is incrementing C_1, we build the TA (see Fig. 6a) $\mathcal{A}_i = (\Sigma, L_i, \ell_0^i, \ell_{priv}, \ell_f, \{x\}, I_i, E_i)$ where:

- $L_i = \{\ell_0^i, \ell_1^i, \ell_2^i, \ell_3^i, \ell_0^{i+1}, \ell_{priv}, \ell_f\}$,
- $I_i(\ell) = true$ for all $\ell \in L_i$,
- $E_i = \big\{(\ell_0^i, x = 3, u, \{x\}, \ell_0^{i+1}), (\ell_0^i, 0 < x < 1, inc_{C_1}, \emptyset, \ell_1^i), (\ell_0^i, 0 < x < 1, inc_{C_1}, \{x\}, \ell_2^i), (\ell_0^i, x = 1, u, \emptyset, \ell_f), (\ell_1^i, x = 1, u, \{x\}, \ell_{priv}), (\ell_{priv}, x = 0, u, \emptyset, \ell_f), (\ell_2^i, x = 3, u, \emptyset, \ell_f), (\ell_2^i, 0 < x < 1, inc_{C_1}, \emptyset, \ell_3^i), (\ell_3^i, true, u, \emptyset, \ell_f)\big\}$.

In this gadget, because of edge e_7, a public run will reach the final location at time 1. The only way to make this time opaque is via edges e_2, e_3 and e_4 (which creates a private run of the same duration). This requires allowing action inc_{C_1} somewhen in $(0, 1)$. Because of edge e_1 however, this additionally creates a public run that will reach the final location 3 time units later and will have to be made opaque thanks to G_{C_1}. At time 3, a run can then go to the initial state of command c_{i+1} via edge e_6, and thus start the next TA fragment.

As a consequence, assuming that σ allows a_{C_1} k times during the interval $(0, 1)$ of this process, (say, at times $\tau_1, \ldots, \tau_k$), then (with the exception of the case where c_{i+1} is a decrement command), σ will allow a_{C_1} $k + 1$ times during the interval $(0, 1)$ of the next process. Indeed, let τ_{k+1} be the time where σ allowed inc_{C_1}. First note that for all $i \leq k$, $\tau_i \neq \tau_{k+1}$ because of gadget *1act* forbidding several actions to be allowed at the same time. Hence, there are $k + 1$ different times at which a public run will reach the final location during the next interval $(0, 1)$, and (ignoring the decrement gadget) only allowing action a_{C_1} at those times protects opacity.

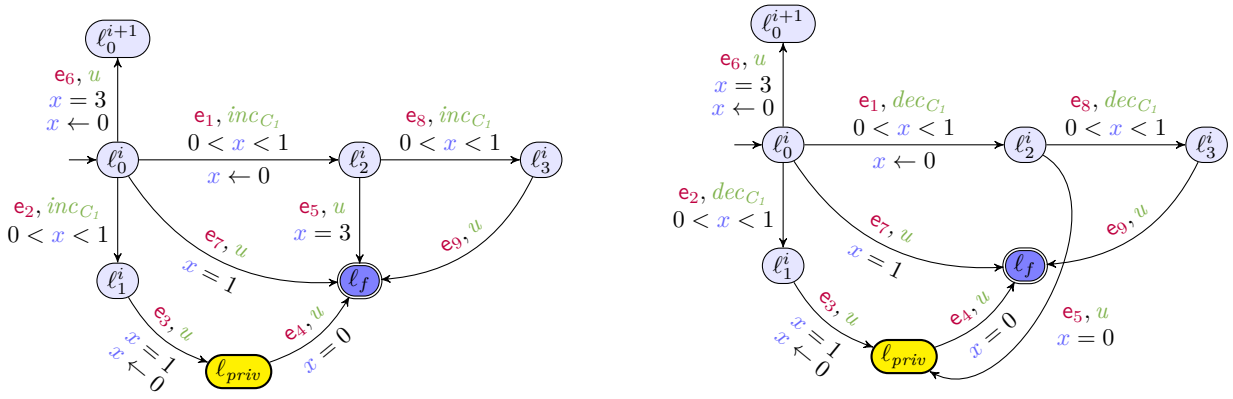
Finally, in order to avoid violating opacity, σ must not allow inc_x more than once. Indeed, if inc_x is allowed at two different times τ_1 and τ_2 during $(0, 1)$, then a run could take e_1 at time τ_1 then e_8 at time τ_2. And once ℓ_3^i is reached, the final location can be reached at any point by a public run, violating opacity.

Decrement gadget

We now move to the decrement command, which is encoded similarly to the increment command. The only difference is that from ℓ_2^i, instead of going to ℓ_f when $x = 3$, there is an uncontrollable transition going immediately ($x = 0$) to ℓ_{priv}. Formally, if command c_i is decrementing C_1, we build the TA (see Fig. 6b) $\mathcal{A}_i = (\Sigma, L_i, \ell_0^i, \ell_{priv}, \ell_f, \{x\}, I_i, E_i)$ where:

- $L_i = \{\ell_0^i, \ell_1^i, \ell_2^i, \ell_3^i, \ell_0^{i+1}, \ell_{priv}, \ell_f\}$,
- $I_i(\ell) = true$ for all $\ell \in L_i$,
- $E_i = \big\{(\ell_0^i, x = 3, u, \{x\}, \ell_0^{i+1}), (\ell_0^i, 0 < x < 1, dec_{C_1}, \emptyset, \ell_1^i), (\ell_0^i, 0 < x < 1, dec_{C_1}, \{x\}, \ell_2^i), (\ell_0^i, x = 1, u, \emptyset, \ell_f), (\ell_1^i, x = 1, u, \{x\}, \ell_{priv}), (\ell_{priv}, x = 0, u, \emptyset, \ell_f), (\ell_2^i, x = 0, u, \emptyset, \ell_{priv}), (\ell_2^i, 0 < x < 1, dec_{C_1}, \emptyset, \ell_3^i), (\ell_3^i, true, u, \emptyset, \ell_f)\big\}$.

As for the increment gadget, in order to preserve opacity, σ must allow dec_{C_1} exactly once during the interval. Moreover, as it immediately produces a private run, it should be played at a time where a public run is supposed to reach the destination. Therefore, if $\tau_1, \ldots, \tau_k$ are the times



(a) Gadget inc_{C_1}

(b) Gadget dec_{C_1}

Fig. 6: Increment and decrement gadgets

at which public runs are supposed to reach the final location within this $(0, 1)$ interval, then dec_{C_1} must be allowed at one of them (say τ_1), and a_{C_1} must be allowed at the $k-1$ others. Hence producing $k-1$ new public runs that will reach the final locations at the times $\tau_2, \ldots, \tau_k$ of the next process. Hence, effectively decrementing by one the number of times a_{C_1} must be allowed on the next step.

Zero-test gadget

We now move to the case where the command c_i is a zero-test. In this construction (see Fig. 7), the strategy must initially indicate whether it claims the counter is zero or not. Then, following this claim, we just need to check whether the action a_{C_1} is allowed at some point within the interval $(0, 1)$ or not.

More precisely, assuming the test is of the form "if $C_1 = 0$ go to c_k otherwise go to c_j", we build the TA $\mathcal{A}_i = (\Sigma, L_i, \ell_0^i, \ell_{priv}, \ell_f, \{x\}, I_i, E_i)$ where:

- $L_i = \{\ell_0^i, \ell_0^j, \ell_0^k, \ell_=^i, \ell_{=,2}^i, \ell_{\neq}^i, \ell_{\neq,2}^i, \ell_{priv}, \ell_f\}$,
- $I_i(\ell) = true$ for all $\ell \in L_i$,
- $E_i = \big\{(\ell_0^i, x = 0, =_0, \emptyset, \ell_=^i)$,
  $(\ell_0^i, x = 0, \neq_0, \emptyset, \ell_{\neq}^i), (\ell_0^i, x = 1, u, \emptyset, \ell_f)$,
  $(\ell_=^i, x = 3, u, \{x\}, \ell_0^k)$,
  $(\ell_=^i, x = 1, u, \{x\}, \ell_{priv})$,
  $(\ell_=^i, 0 < x < 1, a_{C_1}, \emptyset, \ell_{=,2}^i)$,
  $(\ell_{priv}, x = 0, u, \emptyset, \ell_f), (\ell_{=,2}^i, true, u, \emptyset, \ell_f)$,
  $(\ell_{\neq}^i, x = 3, u, \{x\}, \ell_0^j)$,
  $(\ell_{\neq}^i, 0 < x < 1, a_{C_1}, \emptyset, \ell_{\neq,2}^i)$,
  $(\ell_{\neq,2}^i, x = 1, u, \{x\}, \ell_f)\big\}$.

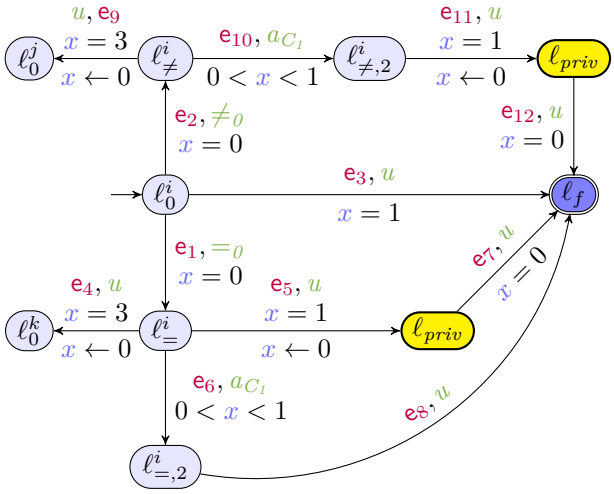


Fig. 7: Gadget if_{C_1} (the location ℓ_{priv} is duplicated to avoid crossing transitions).

Let us explain this gadget. When entering it, due to e_3, a public run will reach the final location at time 1. To avoid this, the strategy has two options. When $x = 0$ it can either claim counter C_1 is equal to 0, and allow the action $=_0$, hence letting a run take e_1, or claim it is not equal to 0, allowing $\neq_0$, hence letting a run take e_2.

Let us first consider the case where σ allowed $\neq_0$. Then, in order to produce a private run at time 1, the strategy must allow a_{C_1} in the interval $(0, 1)$, letting a run take e_{10} and then e_{12}, which means the counter is not equal to 0, and thus that the claim was correct. Then, at time 3,

a run will reach ℓ_0^j and continue the process with command c_j.

If the strategy allowed $=_0$, then a private run will be produced via edges e_5 and e_7, making time 1 opaque. Moreover, if a_{C_1} is allowed at some point within the interval $(0,1)$ (meaning the counter is not 0 and thus that the claim was false), then e_6 can be taken, ensuring that every duration from that point can be accessed by a public run, and thus violating opacity. Then, at time 3, a run will reach ℓ_0^k and continue the process with command c_k.

Hence, the only way for the strategy to avoid violating opacity during this process is to correctly select whether the counter is empty or not (and thus to allow the right action) when $x = 0$. This ensures the system continues with the correct command (c_j or c_k).

Termination gadget

Termination is achieved when c_m is reached. In our case, we wish that termination *violates* opacity. Hence the command c_m is represented by a simple TA $\mathcal{A}_m$ where, from the initial location ℓ_0^m, one can go to the final location at any time without control, i.e., via an edge labelled with u. Hence, every duration from this point becomes associated to a public run, violating opacity.

Conclusion of the proof

We build the TA $\mathcal{A}$ by combining the gadgets for every command c_i, as well as the gadgets G_{C_1}, G_{C_2} and *1act* (noting that the final and private location of each gadget can be merged), and with an additional initial location ℓ_0 from which one can reach in 0-time the locations $\ell_0^{G_{C_1}}, \ell_0^{G_{C_2}}, \ell_0^{1act}$ and ℓ_0^0 in a non-deterministic manner via uncontrollable transitions.

We have that there exists a strategy σ enforcing full ET-opacity of $\mathcal{A}$ iff $\mathcal{M}$ does *not* terminate.

Indeed, assume that $\mathcal{M}$ does not terminate. As explained throughout the gadgets, by building a strategy emulating the Minsky machine (making the adequate claim on a zero-test, and for instance allowing a_{C_1} at times $\frac{1}{2^n}$ for all n smaller or equal to the value of counter C_1, and similarly for counter C_2), then opacity is ensured. Conversely, if $\mathcal{M}$ terminates, either the strategy does not emulate correctly the Minsky machine and thus violates opacity as previously discussed, or it reaches the gadget associated to c_m—which again leads to a violation of opacity.

This proves that the full ET-opacity problem is undecidable. ■

Remark 1. Note that the construction relies on a *single clock*, hence the problem is undecidable even when restricting to one-clock TAs. □

Remark 2. This proof applies whether the strategy is assumed finitely-varying or not. Thus the finitely-varying assumption would not help in regaining decidability. □

5 The belief automaton

In this section, we build an automaton called the *belief automaton*, that will allow us to determine in which regions the system can be after a given execution time. This automaton considers a duplicated TA instead of the original TA in order to distinguish the final state reached by a private or a public run.[2]

5.1 Separating private and public runs

We define a duplicated version of a TA $\mathcal{A}$, denoted by $\mathcal{A}^{dup}$, making it possible to decide whether a given run avoided ℓ_{priv}, by just looking at the final reached location. The duplicated version $\mathcal{A}^{dup}$ is such that any run of $\mathcal{A}$ has an equivalent one in $\mathcal{A}^{dup}$ where each location is replaced by its duplicated version if a previously visited location is ℓ_{priv}. In particular, $DVisit^{pub}(\mathcal{A}) = DVisit^{pub}(\mathcal{A}^{dup})$ and $DVisit^{priv}(\mathcal{A}) = DVisit^{priv}(\mathcal{A}^{dup})$.

Definition 13 (Duplicated TA)**.** Let $\mathcal{A} = (\Sigma, L, \ell_0, \ell_{priv}, F, \mathbb{X}, I, E)$ be a TA. The associated *duplicated TA* is $\mathcal{A}^{dup} = (\Sigma, L', \ell_0, \ell_{priv}, F', \mathbb{X}, I', E')$ where:

1. $L' = L_{pub} \uplus L_{priv}$ with $L_{pub} = L \setminus \ell_{priv}$ and $L_{priv} = \{\ell^p \mid \ell \in L\} \cup \{\ell_{priv}\}$,
2. $F' = \{\ell_f^p \mid \ell_f \in F\} \cup F$,
3. I' is the invariant such that $\forall \ell \in L, I'(\ell) = I'(\ell^p) = I(\ell)$, and
4. $E' = \big\{(\ell_1, g, a, R, \ell_2) \mid (\ell_1, g, a, R, \ell_2) \in E \text{ and } \ell_1 \neq \ell_{priv}\big\} \cup \big\{(\ell_1^p, g, a, R, \ell_2^p) \mid (\ell_1, g, a, R, \ell_2) \in E\big\} \cup \big\{(\ell_{priv}, g, a, R, \ell^p) \mid (\ell_{priv}, g, a, R, \ell) \in E\big\}$.

[2] This could equally have been encoded using a Boolean variable remembering whether ℓ_{priv} was visited, as in [ALMS22].

□

That is, edges in $\mathcal{A}^{dup}$ are made of the original edges of $\mathcal{A}$ except these originating from the private location ℓ_{priv}, plus a copy of the edges between the duplicated locations ℓ_i^p, plus edges from the private location ℓ_{priv} to the duplicated version of the target locations. In other words, once ℓ_{priv} is reached, the TA moves to the copy of the original locations, thus remembering whether ℓ_{priv} was visited.

Example 8. Fig. 1b depicts $\mathcal{A}_1^{dup}$, the duplicated version of $\mathcal{A}_1$ in Fig. 1a. The thick line from ℓ_{priv} to ℓ_f^p depicts the transition from the "normal" part of the TA into the "duplicated" part, after visiting ℓ_{priv}. Observe in Fig. 1b that each run avoiding ℓ_{priv} ends in ℓ_f, and that the only outgoing transition of ℓ_{priv} is modified to go to the duplicated ℓ_f^p. □

5.2 Beliefs

A *belief*[3], denoted by $\mathfrak{b}$, represents the set of regions in which the attacker *believes* to be according to their knowledge, i.e., the current absolute time and the strategy (that is, the enabled actions by the controller over time). For a TA $\mathcal{A}$ and a meta-strategy ϕ, we denote by $\mathfrak{b}_t^\phi$ the set of regions in which the system can be after a time t while following a strategy σ such that $\sigma \models \phi$ in $\mathcal{A}$, i.e., $r \in \mathfrak{b}_t^\phi$ iff there exists a strategy σ such that $\sigma \models \phi$ and a run ρ in $\mathcal{A}^{dup}$ such that ρ is σ-compatible, $last(\rho) \in r, r \in \mathsf{R}_{\mathcal{A}^{dup}}$ and $dur(\rho) = t$.

We regroup those beliefs depending on their intervals by defining the set $\mathbb{I}_\mathcal{A}^\phi$ of *interval beliefs* reachable by a meta-strategy ϕ. Formally, for a given meta-strategy ϕ, $\mathbb{I}_\mathcal{A}^\phi = \{\mathfrak{b}_k^\phi \mid k \in \mathbb{N}\} \cup \{\mathfrak{b}_{k+}^\phi \mid k \in \mathbb{N}\}$ where $\mathfrak{b}_k^\phi$ matches the notation introduced above, and $\mathfrak{b}_{k+}^\phi = \bigcup_{t \in (k,k+1)} \mathfrak{b}_t^\phi$.

Among those beliefs, we will be particularly interested in the ones showing *leaks* of information about the system. Intuitively, a leaking belief allows to discriminate private and public runs. For a given TA $\mathcal{A}$, we denote $\mathsf{Private}_\mathcal{A} = \{[(\ell, \mu)] \mid \ell \in L_{priv}, \mu \in \mathbb{R}_{\geq 0}^H\}$ the set of regions reachable after visiting ℓ_{priv} on a run in $\mathcal{A}^{dup}$, and $\mathsf{Public}_\mathcal{A} = \{[(\ell, \mu)] \mid \ell \in L_{pub}, \mu \in \mathbb{R}_{\geq 0}^H\}$ the set of regions reachable on a run not visiting ℓ_{priv} in $\mathcal{A}^{dup}$.

[3] We follow the vocabulary from, e.g., [BFH+14]. This is also close to the concept of *estimator* (e.g., [KKG24]).

Definition 14. Given a TA $\mathcal{A}$, a belief $\mathfrak{b}$ is said to be *leaking for full ET-opacity* when exactly one of the following two conditions is satisfied:

1. $(\mathfrak{b} \cap \mathsf{R}_\mathcal{A}^F \cap \mathsf{Private}_\mathcal{A} \neq \emptyset)$, or
2. $(\mathfrak{b} \cap \mathsf{R}_\mathcal{A}^F \cap \mathsf{Public}_\mathcal{A} \neq \emptyset)$.

□

This means that finishing in this belief leaks an information to the attacker: only one final state is possible (private or public, but not both).

As we will now show, leaking interval beliefs contains the relevant information with respect to full ET-opacity.

Lemma 2. *Let $\mathcal{A}$ be a TA and ϕ a meta-strategy. $\mathcal{A}$ is fully ET-opaque with ϕ iff there is no interval belief in $\mathbb{I}_\mathcal{A}^\phi$ that is leaking for full ET-opacity.*

Proof. $\Longrightarrow$ Let $\mathcal{A}$ be a TA that is fully ET-opaque with meta-strategy ϕ. Let $\mathfrak{b} \in \mathbb{I}_\mathcal{A}^\phi$ be an interval belief. Suppose w.l.o.g. that $\mathfrak{b} \cap \mathsf{R}_\mathcal{A}^F \cap \mathsf{Private}_\mathcal{A} \neq \emptyset$. Let $r \in \mathfrak{b} \cap \mathsf{R}_\mathcal{A}^F \cap \mathsf{Private}_\mathcal{A}$. Then by definition, there is a strategy $\sigma \models \phi$ and a σ-compatible run $\rho \in Visit_\sigma^{priv}(\mathcal{A})$ such that $last(\rho) \in r$. $\mathcal{A}$ being fully ET-opaque with meta-strategy ϕ, $DVisit_\phi^{priv}(\mathcal{A}) = DVisit_\phi^{pub}(\mathcal{A})$. Thus, there exists a strategy $\sigma' \models \phi$ and a σ'-compatible run $\rho' \in Visit_\phi^{pub}(\mathcal{A})$ such that $dur(\rho') = dur(\rho)$. Denoting $r' = [last(\rho')]$, since the two runs have the same duration, they end in the same interval belief and we have $r' \in \mathfrak{b} \cap \mathsf{R}_\mathcal{A}^F \cap \mathsf{Public}_\mathcal{A}$. (Note that $r' \in \mathsf{R}_\mathcal{A}^F$ as a run ρ belongs to $Visit^{pub}(\mathcal{A}) \cup Visit^{priv}(\mathcal{A})$ only if $[last(\rho)] \in \mathsf{R}_\mathcal{A}^F$.) Hence, $\mathfrak{b} \cap \mathsf{R}_\mathcal{A}^F \cap \mathsf{Public}_\mathcal{A} \neq \emptyset$. Therefore $\mathfrak{b}$ is not leaking for full ET-opacity.

$\Longleftarrow$ Conversely, we now assume that there is no interval belief in $\mathbb{I}_\mathcal{A}^\phi$ that is leaking for full ET-opacity. W.l.o.g, we consider a time $\tau \in DVisit_\phi^{priv}(\mathcal{A})$. By definition, there exists a strategy $\sigma \models \phi$ and a σ-compatible run $\rho \in Visit_\sigma^{priv}(\mathcal{A})$ such that $dur(\rho) = \tau$. We want to prove that $\tau \in DVisit_\phi^{pub}(\mathcal{A})$, i.e., that there exists a strategy $\sigma' \models \phi$ and a σ'-compatible run $\rho' \in Visit_{\sigma'}^{pub}(\mathcal{A})$, such that $dur(\rho') = dur(\rho)$.

If $dur(\rho) = k \in \mathbb{N}$, then $r = [last(\rho)]$ is included in the interval belief $\mathfrak{b}_k^\phi$, and more precisely $r \in \mathfrak{b}_k^\phi \cap \mathsf{R}_\mathcal{A}^F \cap \mathsf{Private}_\mathcal{A}$. Since interval beliefs in $\mathbb{I}_\mathcal{A}^\phi$ are not leaking, there also exists a region $r' \in \mathfrak{b}_k^\phi \cap \mathsf{R}_\mathcal{A}^F \cap \mathsf{Public}_\mathcal{A}$. By definition of $\mathfrak{b}_k^\phi$, there must be a strategy $\sigma' \models \phi$ and a

σ'-compatible run ρ' of duration k such that $r' = [last(\rho')]$ and thus a public run of the same duration as ρ.

If $dur(\rho) \in (k, k+1)$ with $k \in \mathbb{N}$, we can use the construction above to prove the existence of a strategy $\sigma'' \models \phi$ and a σ''-compatible run $\rho'' \in Visit^{pub}_{\sigma''}(\mathcal{A})$ such that $r'' = [last(\rho'')] \in \mathfrak{b}^{\phi}_{k+}$. Problem is, we have no guarantee that ρ and ρ'' have the same duration, only that these durations are both in $(k, k+1)$.

To complete this proof, we will, using σ'' and ρ'', build a strategy $\sigma' \models \phi$ and a σ'-compatible run $\rho' \in Visit^{pub}_{\sigma'}(\mathcal{A})$ that has the same duration as ρ. This will be done creating a function *shrink* that will transform a time instant into another. Strategy σ' will then mimick σ'' by setting $\sigma'(t) = \sigma''(shrink(t))$. Each transition of ρ' at time t will mimick a similar transition of ρ'' at time $shrink(t)$. And with the additional property that $shrink(dur(\rho)) = dur(\rho''s)$, runs ρ and ρ' will have the same exact duration.

In order to define *shrink*, let us consider

$$q = \frac{\mathrm{fr}\,(dur(\rho''))}{\mathrm{fr}\,(dur(\rho))} \text{ and } q' = \frac{1-\mathrm{fr}\,(dur(\rho''))}{1-\mathrm{fr}\,(dur(\rho))}.$$

We now define:

$$shrink(t) = \begin{cases} t & \text{if } q = 1 \\ \lfloor t \rfloor + \mathrm{fr}\,(t) \times q & \text{if } q < 1 \\ \lceil t \rceil - (\lceil t \rceil - t) \times q' & \text{if } q > 1 \end{cases}$$

Note first that, since $q < 1$ in case 2, we have, if t is not an integer, $\lfloor t \rfloor < shrink(t) < t$. Similarly, using the fact that $q > 1 \iff q' < 1$, in case 3, if t is not an integer, then $t < shrink(t) < \lceil t \rceil$. Function *shrink* thus moves time instants towards the nearest integer below (case 2) or above (case 3) while preserving the integral part.

As announced before, $shrink(dur(\rho)) = dur(\rho'')$. The first case is straightforward (if $q = 1$) and for the second one we just need to remember that $\lfloor dur(\rho) \rfloor = \lfloor dur(\rho'') \rfloor$. For the third case, since we have assumed that $\mathrm{fr}\,(dur(\rho)) \neq 0$, we have $\lceil dur(\rho) \rceil = \lfloor dur(\rho) \rfloor + 1$ and $\lceil dur(\rho) \rceil - dur(\rho) = 1 - \mathrm{fr}\,(dur(\rho))$. Thus:

$$\begin{aligned} &shrink(dur(\rho)) \\ &\quad = \lceil dur(\rho) \rceil - \big(\lceil dur(\rho) \rceil - dur(\rho)\big) \times q' \\ &\quad = (\lfloor dur(\rho) \rfloor + 1) - \big(1 - \mathrm{fr}\,(dur(\rho))\big) \times \\ &\qquad\qquad \frac{\big(1 - \mathrm{fr}\,(dur(\rho''))\big)}{\big(1 - \mathrm{fr}\,(dur(\rho))\big)} \\ &\quad = \lfloor dur(\rho) \rfloor + \mathrm{fr}\,(dur(\rho'')) \\ &\quad = dur(\rho'') \end{aligned}$$

Then we can see that function *shrink* preserves integer values. When $t = k \in \mathbb{N}$, both $\mathrm{fr}\,(t)$ and $\lceil t \rceil - t$ are 0. Furthermore, function *shrink* is strictly increasing. The two last properties allow us to build the strategy σ' such that $\sigma'(t) = \sigma''(shrink(t))$. Since $\sigma'' \models \phi$ and since σ'' and σ' allow the same actions at integer times, and make the same strategy changes in the same order in between, we have $\sigma' \models \phi$ too. Since our function *shrink* is continuous, strictly increasing over $\mathbb{R}_{\geq 0}$ and has $\mathbb{R}_{\geq 0}$ as its image, it is invertible and the inverse function is defined on $\mathbb{R}_{\geq 0}$ and satisfies:

$$shrink^{-1}(t) = \begin{cases} t & \text{if } q = 1 \\ \lfloor t \rfloor + \mathrm{fr}\,(t)\,/q & \text{if } q < 1 \\ \lceil t \rceil - (\lceil t \rceil - t)/q' & \text{if } q > 1 \end{cases}$$

Using this inverse function, we can now define ρ' as the run that does the same actions as ρ'' in the same order, but where the time at which those transitions are made are transformed by *shrink*. Formally if $\rho'' = (\ell_0, \vec{0}), (d_0, \mathsf{e}_0), \ldots, (d_{n-1}, \mathsf{e}_{n-1}), (\ell_n, \mu_n)$ then we define $\rho' = (\ell_0, \vec{0}), (d'_0, \mathsf{e}_0), \ldots, (d'_{n-1}, \mathsf{e}_{n-1}), (\ell_n, \mu'_n)$ where for every $0 \leq i < n$, $d'_i = shrink^{-1}\left(\sum_{j=0}^{j \leq i} d_j\right) - shrink^{-1}\left(\sum_{j=0}^{j<i} d_j\right)$, and for every $0 < i \leq n$, $\mu'_i = [\mu'_{i-1} + d'_{i-1}]_R$. We need to prove that ρ' is actually a run of $\mathcal{A}^{dup}$, i.e., that all transitions and delays can occur.

Since ρ' visits the same locations and does the same discrete transitions as ρ'', we know that these transitions are possible from those locations. The only thing to prove is that invariants are satisfied for delay transitions,

and guards are satisfied for discrete transitions. In a timed automaton, the guards and invariants only compare a clock value to an integer constant. We need to show that function *shrink* preserves these comparisons.

Given a time instant $t \in \mathbb{R}_{\geq 0}$ and $k \in \mathbb{N}$, since t and $t+k$ have the same fractional part, we have that $shrink^{-1}(t+k) = shrink^{-1}(t) + k$. It is straightforward for case $q = 1$, quite simple for $q < 1$, and for the last case

$$\begin{aligned}&\lceil t+k\rceil - \big(\lceil t+k\rceil - (t+k)\big)/q' \\ =&\lceil t\rceil + k - \big(\lceil t\rceil + k - t - k\big)/q' \\ =&\lceil t\rceil - \big(\lceil t\rceil - t\big)/q' + k.\end{aligned}$$

Since the function $shrink^{-1}$ is increasing, and based on the above equality, we have for any integer value k and any two time instants t and t', if $t - t' \bowtie k$ then $shrink^{-1}(t) - shrink^{-1}(t') \bowtie k$ (with $\bowtie \in \{<, \leq, =, \geq, >\}$). Thus for every transition (or invariant) along ρ'' where a clock x is compared to a value k in a guard, if we take t' as the last instant x has been reset and t the time instant at which the transition is fired (or the invariant checked), the guard (or invariant) will also be satisfied at time $shrink^{-1}(t)$ along ρ'. Since this is true for every guard of every transition (and any invariant) of ρ'', ρ' is indeed a run of $\mathcal{A}^{dup}$. Since it furthermore visits the same states as ρ'', ρ' does not visit the private state, and since we applied the same transformation to create ρ' from ρ'' that we did to create strategy σ' from σ'', ρ' is σ'-compatible and thus in $Visit^{priv}_{\sigma'}(\mathcal{A})$. Adding the fact (proven earlier) that $\sigma' \models \phi$, we get that $dur(\rho') \in DVisit^{priv}_{\phi}(\mathcal{A})$, and recalling that *shrink* ensures that ρ' and ρ have the same duration, we get that $\tau = dur(\rho) \in DVisit^{priv}_{\sigma}(\mathcal{A})$ which concludes the proof.

■

5.3 Belief automaton

If the set $\mathbb{I}^{\phi}_{\mathcal{A}}$ contains the relevant information with respect to full ET-opacity, there is no immediate way to compute and manipulate it. In this endeavour, writing $\mathfrak{E} \subseteq \Sigma_c$ for a set of *enabled* actions we define as follows the belief automaton:

Definition 15 (Belief automaton)**.** Given a TA $\mathcal{A}$ with $\Sigma = \Sigma_c \uplus \Sigma_u$, we define the *belief automaton* as the tuple $\mathcal{B}_{\mathcal{A}} = (\mathfrak{S}^{\mathcal{B}_{\mathcal{A}}}, \mathfrak{A}^{\mathcal{B}_{\mathcal{A}}}, \bot, \mathfrak{d}^{\mathcal{B}_{\mathcal{A}}})$ where:

1. $\mathfrak{S}^{\mathcal{B}_{\mathcal{A}}} = 2^{\mathsf{R}_{\mathcal{A}^{dup}}} \cup \{\bot\}$ is the set of states,
2. $\mathfrak{A}^{\mathcal{B}_{\mathcal{A}}} = \{0, 0^+, 1\} \times 2^{\Sigma_c}$ is the alphabet,
3. $\bot$ is the initial state,
4. $\mathfrak{d}^{\mathcal{B}_{\mathcal{A}}} \subseteq (\mathfrak{S}^{\mathcal{B}_{\mathcal{A}}} \times \mathfrak{A}^{\mathcal{B}_{\mathcal{A}}} \times \mathfrak{S}^{\mathcal{B}_{\mathcal{A}}})$ is such that
   (a) $\big(\bot, (0, \mathfrak{E}), \mathfrak{b}\big) \in \mathfrak{d}^{\mathcal{B}_{\mathcal{A}}}$ iff $\mathfrak{b}$ is the largest set such that $\forall r \in \mathfrak{b}$, $\exists n \geq 0$, $[s_0] \xrightarrow{(0,a_1)}_{\mathsf{R}} \cdots \xrightarrow{(0,a_n)}_{\mathsf{R}} r$ in $\mathcal{R}_{\mathcal{A}^{dup}}$ with $\forall 1 \leq i \leq n$, $a_i \in (\mathfrak{E} \cup \Sigma_u)$,
   (b) $\big(\mathfrak{b}, (\dagger_1, \mathfrak{E}), \mathfrak{b}'\big) \in \mathfrak{d}^{\mathcal{B}_{\mathcal{A}}}$ iff $\mathfrak{b} \neq \bot$, $\mathfrak{b}'$ is the largest set such that $\forall r' \in \mathfrak{b}'$, $\exists r \in \mathfrak{b}$, $\exists n \geq 1$, $r \xrightarrow{(\dagger_1,\varepsilon)}_{\mathsf{R}} \cdots \xrightarrow{(\dagger_n,a_n)}_{\mathsf{R}} r'$ in $\mathcal{R}_{\mathcal{A}^{dup}}$ with $\forall 1 < i \leq n, a_i \in (\mathfrak{E} \cup \Sigma_u \cup \{\varepsilon\})$ and $\dagger_1 \in \{0^+, 1\}$ and $\forall 1 < i \leq n, \dagger_i \in \{0, 0^+\}$.

□

We first consider transitions from the initial belief $\bot$: time cannot elapse here; one can do a sequence of actions in 0-time (condition 4a). Then, from the other beliefs, a transition is made of a sequence of transitions from the region automaton. The first one lets time elapse (possibly changing region for z), and all the following actions are either discrete transitions, or delay transitions remaining in the same region for z (condition 4b).

Example 9. Because there is a single clock in our subsequent examples, as an abuse of notation, we represent each region within a belief using either an open interval, or a unique integer. We write $\big(\ell, (\tau, \tau')\big)$ for the region containing the state $\big(\ell, \mu(x_1)\big)$ with $\mu(x_1) \in (\tau, \tau'), \tau \in \mathbb{N}, \tau' = +\infty$ if $\tau = c_1$, $\tau' = \tau + 1$ otherwise. Similarly, we write (ℓ, τ) for the region containing the state $\big(\ell, \mu(x_1)\big)$, $\mu(x_1) = \tau \in \mathbb{N}$.

Let $\mathcal{A}_{opaque}$ be the TA in Fig. 2. With the global invariant $x \leq 1$, we have the following beliefs. Here, the value of clock z is not given as, in this example, it is equivalent to the value of x.

The corresponding belief automaton is depicted in Fig. 8.

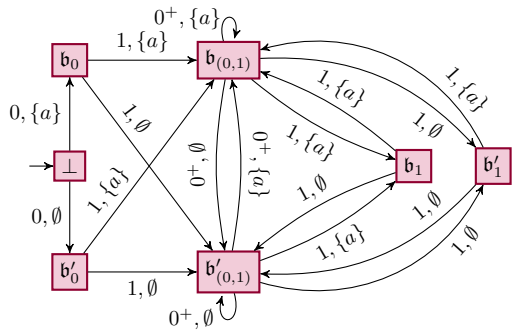


Fig. 8: Belief automaton $\mathcal{B}_{\mathcal{A}_{opaque}}$

$$\mathfrak{b}'_0 = \{(\ell_0, 0), (\ell_{priv}, 0), (\ell_f^p, 0)\}$$
$$\mathfrak{b}_0 = \mathfrak{b}'_0 \cup \{(\ell_f, 0)\}$$
$$\mathfrak{b}'_{(0,1)} = \{(\ell_0, (0,1)), (\ell_{priv}, (0,1))\}$$
$$\mathfrak{b}_{(0,1)} = \mathfrak{b}'_{(0,1)} \cup \{(\ell_f, (0,1))\}$$
$$\mathfrak{b}'_1 = \{(\ell_0, 1), (\ell_0, 0), (\ell_{priv}, 1), (\ell_{priv}, 0), (\ell_f^p, 0)\}$$
$$\mathfrak{b}_1 = \mathfrak{b}'_1 \cup \{(\ell_f, 0), (\ell_f, 1)\}$$

Consider two beliefs reachable from the same belief, with two different sets of available actions, such that one is a subset of the other. We see that the belief reachable with the smaller set is a subset of the belief reachable with the larger set. In fact, restricting the system to only a subset of actions only *restricts* the possible behaviours, and cannot add any.

□

If there is more than one clock, we extend our abuse of notation for regions to $(\ell, \tau_1, \ldots, \tau_H)$, where each τ_i is either an interval or an integer. Note that this notation does not take into account the comparison between clocks but this is acceptable in the following example as the clocks always have the same fractional part.

Example 10. Let $\mathcal{A}'_{opaque}$ the TA depicted in Fig. 9a. With the invariant $x \leq 1$ for ℓ_0 and $y \leq 2$ for ℓ_{priv}, we have the following beliefs. Each region is written $(\ell, \tau_1, \tau_2, \tau_3)$ with τ_1 for x, τ_2 for y and τ_3 for z. The corresponding belief automaton is depicted in Fig. 9b.

$$\mathfrak{b}'_0 = \{(\ell_0, 0, 0, 0), (\ell_{priv}, 0, 0, 0)\}$$
$$\mathfrak{b}_0 = \mathfrak{b}'_0 \cup \{(\ell_f, 0, 0, 0)\}$$
$$\mathfrak{b}'_{(0,1)} = \{(\ell_0, (0,1), (0,1), (0,1)), (\ell_{priv}, (0,1), (0,1), (0,1))\}$$
$$\mathfrak{b}_{(0,1)} = \mathfrak{b}'_{(0,1)} \cup \{(\ell_f, (0,1), (0,1), (0,1))\}$$
$$\mathfrak{b}'_1 = \{(\ell_0, 1, 1, 1), (\ell_0, 1, 1, 0), (\ell_0, 0, 1, 1), (\ell_0, 0, 1, 0), (\ell_{priv}, 0, 1, 1), (\ell_{priv}, 0, 1, 0), (\ell_{priv}, 1, 1, 1), (\ell_{priv}, 1, 1, 0)\}$$
$$\mathfrak{b}_1 = \mathfrak{b}'_1 \cup \{(\ell_f, 1, 1, 1), (\ell_f, 1, 1, 0), (\ell_f, 0, 1, 1), (\ell_f, 0, 1, 0)\}$$
$$\mathfrak{b}'_{(1,2)} = \{(\ell_0, (0,1), (1,2), (0,1)), (\ell_{priv}, (0,1), (1,2), (0,1)), (\ell_{priv}, (1,+\infty), (1,2), (0,1))\}$$
$$\mathfrak{b}_{(1,2)} = \mathfrak{b}'_{(1,2)} \cup \{(\ell_f, (0,1), (1,2), (0,1))\}$$
$$\mathfrak{b}'_2 = \{(\ell_0, 1, 2, 1), (\ell_0, 1, 2, 0), (\ell_0, 0, 2, 1), (\ell_0, 0, 2, 0), (\ell_{priv}, 0, 2, 1), (\ell_{priv}, 0, 2, 0), (\ell_{priv}, (1,+\infty), 2, 1), (\ell_{priv}, (1,+\infty), 2, 0), (\ell_{priv}, 1, 2, 1), (\ell_{priv}, 1, 2, 0), (\ell_f^p, 0, 2, 1), (\ell_f^p, 0, 2, 0)\}$$
$$\mathfrak{b}_2 = \mathfrak{b}'_2 \cup \{(\ell_f, 1, 2, 1), (\ell_f, 1, 2, 0), \}$$
$$\mathfrak{b}'_{(2,3)} = \{(\ell_0, (0,1), (2,+\infty), (0,1))\}$$
$$\mathfrak{b}_{(2,3)} = \mathfrak{b}'_{(2,3)} \cup \{(\ell_f, (0,1), (2,+\infty), (0,1))\}$$
$$\mathfrak{b}_3 = \mathfrak{b}'_3 \cup \{(\ell_f, 0, (2,+\infty), 1), (\ell_f, 0, (2,+\infty), 0), (\ell_f, 1, (2,+\infty), 1), (\ell_f, 1, (2,+\infty), 0)\}$$
$$\mathfrak{b}'_3 = \{(\ell_0, 1, (2,+\infty), 1), (\ell_0, 1, (2,+\infty), 0), (\ell_0, 0, (2,+\infty), 1), (\ell_0, 0, (2,+\infty), 0)\}$$

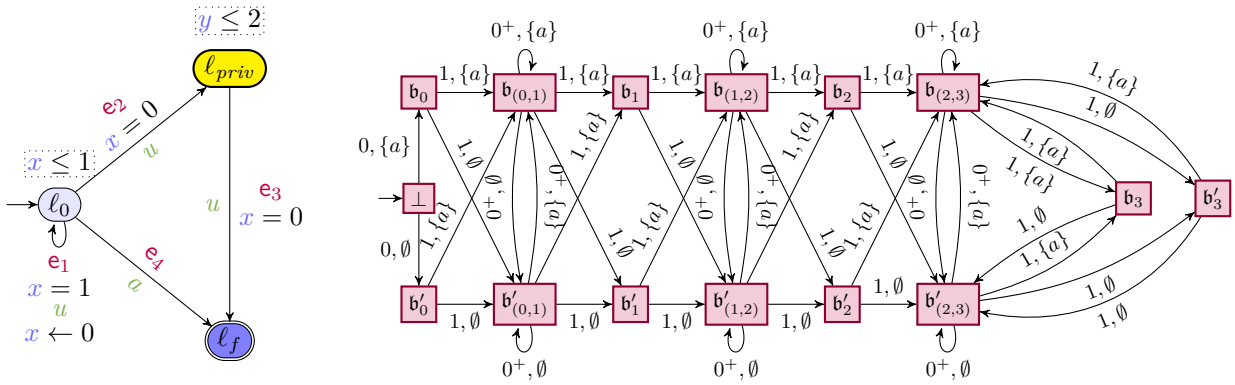
□

5.3.1 Controlled belief automaton and encountered beliefs

We will introduce in Definition 16 a version of the belief automaton controlled by a meta-strategy ϕ. One transition of the controlled belief automaton will group all possible sequences of transitions made in the belief automaton between two strategy changes in ϕ. We thus need to keep track, in the states, of the sequence of strategy choices made until that state is reached, together with the current belief. To do so, we first introduce a notation:

- $\phi_{k,0} = \phi([k,k])$;
- if $\phi((k,k+1)) = (\nu_1, \ldots, \nu_{m_k})$, then for all $1 \leq i \leq m_k$, $\phi_{k,i} = \nu_i$.

In an execution of the belief automaton, the time elapsed can be inferred from the number of actions of the form $(1, \cdot)$ that have been taken so far. If this number is of the form $2k$, then exactly k time units have elapsed. If it is of the form $2k+1$ then the time elapsed is within the interval $(k, k+1)$. We thus need a function that, given a sequence of elements in $\{0, 0^+, 1\} \times 2^{\Sigma_c}$ stating when clock z has changed from one region to another, and the consecutive choices of enabled actions made by the meta-strategy so far, gives the next choice the meta strategy will make and whether or not z will change region next.



(a) TA $\mathcal{A}'_{opaque}$

(b) Belief automaton $\mathcal{B}_{\mathcal{A}'_{opaque}}$

Fig. 9: $\mathcal{A}'_{opaque}$ and the corresponding belief automaton

Given a sequence $v \in (\{0, 0^+, 1\} \times 2^{\Sigma_c})^*$, denoting by $2k + k'$ (with $k \in \mathbb{N}$ and $k' \in \{0, 1\}$) the number of actions of the form $(1, \cdot)$ in v, by i the length of the longest suffix of v without any action of the form $(1, \cdot)$, and by m_k the length of the sequence $\phi((k, k + 1))$, the function $next_\phi$ is defined by:

- $next_\phi(\varepsilon) = (0, \phi_{0,0})$
- if $v \neq \varepsilon$ and $k' = 0$, $next_\phi(v) = (1, \phi_{k,1})$
- if $k' = 1$ and $i < m_k$, $next_\phi(v) = (0^+, \phi_{k,i+1})$
- if $k' = 1$ and $i = m_k$, $next_\phi(v) = (1, \phi_{k+1,0})$

Example 11. Let ϕ a meta-strategy defined (on the interval $[0, 1]$) by: $\phi_{0,0} = \mathfrak{E}_0$, $\phi_{0,1} = \mathfrak{E}_1$, $\phi_{0,2} = \mathfrak{E}_2$, $\phi_{1,0} = \mathfrak{E}_3$. Then,

- $next_\phi(\varepsilon) = (0, \mathfrak{E}_0)$ (first case),
- $next_\phi((0, \mathfrak{E}_0)) = (1, \mathfrak{E}_1)$ (second case),
- $next_\phi((0, \mathfrak{E}_0), (1, \mathfrak{E}_1)) = (0^+, \mathfrak{E}_2)$ (third case),
- $next_\phi((0, \mathfrak{E}_0), (1, \mathfrak{E}_1), (0^+, \mathfrak{E}_2)) = (1, \mathfrak{E}_3)$ (fourth case).

□

Definition 16 (Controlled belief automaton)**.** Given a belief automaton $\mathcal{B}_\mathcal{A} = (\mathfrak{S}^{\mathcal{B}_\mathcal{A}}, \mathfrak{A}^{\mathcal{B}_\mathcal{A}}, \bot, \mathfrak{d}^{\mathcal{B}_\mathcal{A}})$ and a meta-strategy ϕ, we define $\mathcal{B}^\phi_\mathcal{A} = \left(\mathfrak{S}^{\mathcal{B}^\phi_\mathcal{A}}, \mathfrak{A}^{\mathcal{B}^\phi_\mathcal{A}}, (\varepsilon, \bot), \mathfrak{d}^{\mathcal{B}^\phi_\mathcal{A}}\right)$ the *belief automaton controlled by* ϕ as follows:

1. $\mathfrak{S}^{\mathcal{B}^\phi_\mathcal{A}} = (\mathfrak{A}^{\mathcal{B}_\mathcal{A}})^* \times \mathfrak{S}^{\mathcal{B}_\mathcal{A}}$ is the set of states,
2. $\mathfrak{A}^{\mathcal{B}^\phi_\mathcal{A}} = \mathfrak{A}^{\mathcal{B}_\mathcal{A}} = \{0, 0^+, 1\} \times 2^{\Sigma_c}$ is the alphabet,
3. $(\varepsilon, \bot)$ is the initial state,
4. $\mathfrak{d}^{\mathcal{B}^\phi_\mathcal{A}} \subseteq (\mathfrak{S}^{\mathcal{B}^\phi_\mathcal{A}} \times \mathfrak{A}^{\mathcal{B}^\phi_\mathcal{A}} \times \mathfrak{S}^{\mathcal{B}^\phi_\mathcal{A}})$ and $\left((v, \mathfrak{b}), (\dagger, \mathfrak{E}), (v \cdot (\dagger, \mathfrak{E}), \mathfrak{b}')\right) \in \mathfrak{d}^{\mathcal{B}^\phi_\mathcal{A}}$ if $(\mathfrak{b}, (\dagger, \mathfrak{E}), \mathfrak{b}') \in \mathfrak{d}^{\mathcal{B}_\mathcal{A}}$, and $next_\phi(v) = (\dagger, \mathfrak{E})$.

□

In other words, the belief automaton controlled by a meta-strategy ϕ retains only the transitions from the belief automaton that correspond to the meta-strategy.

We now define the beliefs encountered by the controlled belief automaton as sets obtained as the union of every belief between a pair of choices of the form $(1, \mathfrak{E})$, in other words, the beliefs on an integer timestamp, or by regrouping the beliefs visited during an open interval. We will see later that this object is equal to the set of interval beliefs $\mathbb{I}^\phi_\mathcal{A}$.

Definition 17 (Belief encountered by the controlled belief automaton)**.** A belief $\mathfrak{b}$ is said to be encountered by a controlled belief automaton $\mathcal{B}^\phi_\mathcal{A}$ if $((0, \mathfrak{E}), \mathfrak{b})$ is reachable in $\mathcal{B}^\phi_\mathcal{A}$, or there exists a sequence $(v_0, \mathfrak{b}_0), \dots (v_m, \mathfrak{b}_m)$ of states of $\mathfrak{S}^{\mathcal{B}^\phi_\mathcal{A}}$ such that

- $\forall i \leq m, (v_i, \mathfrak{b}_i)$ are reachable in $\mathcal{B}^\phi_\mathcal{A}$,
- $\forall i \leq m, next_\phi(v_i) = (\dagger_i, \mathfrak{E}_i)$, $\dagger_i \in \{0^+, 1\}$ with $\dagger_i = 1$ iff $i = 0$ or $i = m$,
- $\forall i < m, v_{i+1} = v_i \cdot (\dagger_i, \mathfrak{E}_i)$,
- $\mathfrak{b} = \bigcup_{i=1}^m \mathfrak{b}_i$.

The set of beliefs encountered by $\mathcal{B}^\phi_\mathcal{A}$ is denoted by $\mathbb{E}^\phi_\mathcal{A}$. □

5.3.2 Feasible runs

Let us now relate a controlled belief automaton and runs of $\mathcal{A}$. In the following definition, v is a sequence of subsets of controllable actions (which

will be associated later to a sequence of strategy choices).

Definition 18 (Run admitting a sequence)**.** Let $\mathcal{A}$ be a TA, ρ be a run of $\mathcal{A}^{dup}$ and $v \in (\mathfrak{A}^{\mathcal{B}_\mathcal{A}})^*$. We say that ρ *admits* v, denoted by $\rho \vdash v$, when either:

(run reduced to the initial state) $\rho = (\ell_0, \vec{0})$ and $v = (0, \mathfrak{E}_0)$ with $\mathfrak{E}_0 \subseteq \Sigma_c$,
or
(normal run) $\rho = \rho', (d_{n-1}, \mathsf{e}_{n-1}), (\ell_n, \mu_n)$ with $\mathsf{e}_{n-1} = (\ell_{n-1}, g, a, R, \ell_n)$ and one of the following holds:

1. $d_{n-1} = 0$, $\rho' \vdash v$, $v = v' \cdot (\dagger, \mathfrak{E})$ and $a \in \mathfrak{E} \cup \Sigma_u \cup \{\varepsilon\}$,
2. $0 < d_{n-1} < 1$, $v = v' \cdot (1, \mathfrak{E}_0) \cdot (\dagger_1, \mathfrak{E}_1) \cdots (\dagger_{m-1}, \mathfrak{E}_{m-1}) \cdot (\dagger_m, \mathfrak{E}_m)$, $a \in \mathfrak{E}_m \cup \Sigma_u \cup \{\varepsilon\}$, for all $1 \leq k < m$, $\dagger_k = 0^+$, and one of the following holds:
   (a) $\mathrm{fr}(\mu_{n-1}(z)) \neq 0$, $\mathrm{fr}(\mu_n(z)) \neq 0$, $\dagger_m = 0^+$, and there exists $i, 0 \leq i \leq m$, $\rho' \vdash v' \cdot (1, \mathfrak{E}_0) \cdot (0^+, \mathfrak{E}_1) \cdots (\dagger', \mathfrak{E}_i)$,
   (b) $\mathrm{fr}(\mu_{n-1}(z)) = 0$, $\rho' \vdash v'$, and either $m = 0$ or $\dagger_m = 0^+$,
   (c) $\mathrm{fr}(\mu_n(z)) = 0$, $\dagger_m = 1$, and there exists $i \in \{0, \ldots, m-1\}$ such that $\rho' \vdash v' \cdot (1, \mathfrak{E}_0) \cdot (0^+, \mathfrak{E}_1) \cdots (\dagger', \mathfrak{E}_i)$,
3. $d_{n-1} = 1$, $v = v' \cdot (1, \mathfrak{E}_0) \cdot (0^+, \mathfrak{E}_1) \cdots (0^+, \mathfrak{E}_m) \cdot (1, \mathfrak{E})$ with $0 \leq m$ and such that $\rho' \vdash v'$, and $a \in \mathfrak{E} \cup \Sigma_u \cup \{\varepsilon\}$.

□

In the above definition, when the run has no transition, it is associated to a sequence of just one element, $(0, \mathfrak{E}_0)$ where 0 means that no time has passed, and $\mathfrak{E}_0$ is the first choice of actions. For a non-empty run, condition 1 states that when two transitions of ρ occur at the same time, the corresponding set of actions has to be the same (as there will be only one set of actions enabled at a given time instant). Condition 3 corresponds to two actions separated by exactly one time unit. In this case (since clock z is reset at every integer time) the two actions occur at an integer time. The sequence associated to ρ' is completed with a sequence of pairs where the first and last one correspond to a change of region for z (showed by the "1" as first element). Condition 2 is more complex. Let us denote t_{n-1} the time instant of the end of ρ', and t_n the time instant of the end of ρ. There is a sequence v'' such that $\rho' \vdash v''$ and such that the following holds. If both time instants have a non-0 fractional part (condition 2a), then we complete v'' with a (possibly empty) sequence where every pair has a first component equal to 0^+, as z does not switch region outside integer time instants. If only t_{n-1} is an integer (condition 2b), then we add to v'' a sequence starting with a region change (first component equal to 1), and if only t_n is an integer (condition 2c), then v'' is completed with a non empty sequence ending by a region change (first component of the pair equals 1).

Definition 19 (Feasible run)**.** Let $\mathcal{A}$ be a TA, ρ be a run of $\mathcal{A}^{dup}$ and ϕ a meta-strategy. We say that ρ is *feasible in* $\mathcal{B}^\phi_\mathcal{A}$ when there exist $v \in (\mathfrak{A}^{\mathcal{B}_\mathcal{A}})^*$ and a belief $\mathfrak{b} \in \mathfrak{S}^{\mathcal{B}_\mathcal{A}}$ such that $\rho \vdash v$, $(v, \mathfrak{b})$ is reachable in $\mathcal{B}^\phi_\mathcal{A}$ and $r = [last(\rho)] \in \mathfrak{b}$. □

Note that, from item 4 of Definition 16, there is only one action possible in each belief of $\mathcal{B}^\phi_\mathcal{A}$, and thus only one execution. For every feasible run ρ the corresponding v is then of the form $(0, \phi_{0,0}) \cdot (1, \phi_{0,1}) \cdot (0^+, \phi_{0,2}) \cdots (0^+, \phi_{0,m_0}) \cdot (1, \phi_{1,0}) \cdot (1, \phi_{1,1}) \cdot (0^+, \phi_{1,2}) \cdots (0^+, \phi_{1,m_1}) \cdots$, where for all k, $\phi((k, k+1)) = \phi_{k,1}, \ldots, \phi_{k,m_k}$ with $m_k \geq 1$.

Example 12. Consider again the TA $\mathcal{A}_{opaque}$ in Fig. 2. Let σ be the strategy defined as follows:

$$\sigma(\tau) = \begin{cases} \{a\} & \text{if } \tau \in \mathbb{N} \\ \emptyset & \text{otherwise} \end{cases}$$

$\mathcal{A}_{opaque}$ is fully ET-opaque with σ.

Let ϕ be a meta-strategy defined as follows:

- $\phi_{i,0} = \{a\}$, for all $i \leq 0$,
- $\phi_{i,1} = \emptyset$, for all $i \leq 0$.

First note that $\sigma \models \phi$. Then, the first states of the automaton $\mathcal{B}^\phi_{\mathcal{A}_{opaque}}$ are depicted in Fig. 10.

Runs ρ_1 and ρ_2 are feasible in $\mathcal{B}^\phi_{\mathcal{A}_{opaque}}$:

$$\begin{aligned} \rho_1 &= (\ell_0, 0), (1, \mathsf{e}_1), (\ell_0, 0), (0, \mathsf{e}_2), (\ell_{priv}, 0), \\ &\quad (0, \mathsf{e}_3), (\ell^p_f, 0) \\ \rho_2 &= (\ell_0, 0), (1, \mathsf{e}_1), (\ell_0, 0), (0, \mathsf{e}_4), (\ell_f, 0) \end{aligned}$$

For $v = (0, \{a\}) \cdot (1, \emptyset) \cdot (1, \{a\})$ a sequence of actions in the automaton $\mathcal{B}^\phi_{\mathcal{A}_{opaque}}$, we have $\rho_1 \vdash v$ and $\rho_2 \vdash v$. □

We can now link a run and a meta-strategy.

Lemma 3 (Strategies and run feasibility)**.** *Let $\mathcal{A}$ be a TA, ρ a run of $\mathcal{A}^{dup}$ and ϕ a meta-strategy. There exist $\sigma \models \phi$ such that ρ is σ-compatible iff ρ is feasible in $\mathcal{B}^\phi_\mathcal{A}$.*

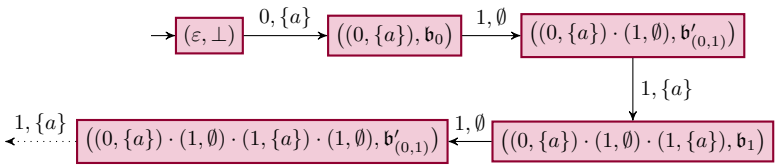


Fig. 10: $\mathcal{B}^{\phi}_{\mathcal{A}_{opaque}}$: First states of the controlled belief automaton for TA $\mathcal{A}_{opaque}$ and meta-strategy ϕ

Proof. $\Longrightarrow$ Let $\sigma \models \phi$ be a strategy and ρ be a σ-compatible run of $\mathcal{A}^{dup}$ s.t. $\rho = (\ell_0, \vec{0}), (d_0, \mathsf{e}_0), \ldots, (d_{n-1}, \mathsf{e}_{n-1}), (\ell_n, \mu_n)$. We build inductively from ρ a sequence $v \in (\mathfrak{A}^{\mathcal{B}_{\mathcal{A}}})^*$ such that $\rho \vdash v$ and the sequence v can be fired in $\mathcal{B}^{\phi}_{\mathcal{A}}$.

As ρ is σ-compatible, for all $0 \leq i < n$, $(\ell_i, \mu_i, \sum_{j<i} d_j) \xrightarrow{d_i, \mathsf{e}_i}_{\sigma} (\ell_{i+1}, \mu_{i+1}, \sum_{j \leq i} d_j)$.

First for $\rho_0 = (\ell_0, \vec{0})$, we set $v_0 = (0, \phi([0,0])) = next_{\phi}(\varepsilon)$. By construction of the controlled belief automaton, $next_{\phi}(\varepsilon)$ is precisely the action allowed in the initial state of $\mathcal{B}^{\phi}_{\mathcal{A}}$ and there exists a belief $\mathfrak{b}_0$ such that $(v_0, \mathfrak{b}_0)$ is reachable in $\mathcal{B}^{\phi}_{\mathcal{A}}$.

Now, let us assume that we have built a sequence v_i such that $\rho_i = (\ell_0, \vec{0}), (d_0, \mathsf{e}_0), \ldots, (d_{i-1}, \mathsf{e}_{i-1}), (\ell_i, \mu_i) \vdash v_i$ with $i < n$, and a belief $\mathfrak{b}_i$ such that $(v_i, \mathfrak{b}_i)$ is reachable in $\mathcal{B}^{\phi}_{\mathcal{A}}$. We define v_{i+1} by completing v_i with all the actions (in order) of the controlled belief automaton corresponding to the changes in the strategy σ between time instants $\sum_{j<i} d_j$ excluded and $\sum_{j \leq i} d_j$ included.

More precisely, following the definition of admissible sequence, we have three possible cases:

- $d_i = 0$, in which case the strategy has not changed since the two consecutive transitions occur at the same time. We thus set $v_{i+1} = v_i$.
- $d_i = 1$, the time between two resets of clock z. In this case $\sum_{j<i} d_j = k \in \mathbb{N}$[4] and, if $\phi((k, k+1)) = \phi_{k,1}, \ldots, \phi_{k,m}$, we set $v_{i+1} = v_i \cdot (1, \phi_{k,1}), (0^+, \phi_{k,2}), \ldots, (0^+, \phi_{k,m}), (1, \phi_{k+1,0})$. This construction respects the properties of Definition 18 (since $\sigma \models \phi$, the last action of ρ_{i+1} fired at time $k+1$ has to belong to $\phi_{k+1,0}$), so we have $\rho_{i+1} \vdash v_{i+1}$.
- $0 < d_i < 1$. Let k be the integral part of $\sum_{j<i} d_j$, $\phi((k, k+1)) = \phi_{k,1}, \ldots, \phi_{k,m}$, $\iota_1, \ldots, \iota_m$ the ordered partition of $(k, k+1)$ corresponding to the changes of strategy for σ in this interval and $\iota_0 = [k, k]$, $\iota_{m+1} = [k+1, k+1]$. Denoting l the index such that $\sum_{j<i} d_j \in \iota_l$ and l' the index such that $\sum_{j \leq i} d_j \in \iota_{l'}$, then if $l = l'$ we have $v_{i+1} = v_i$, otherwise $v_{i+1} = v_i \cdot (\dagger_{l+1}, \phi_{k,l+1}) \cdots (\dagger_{l'}, \phi_{k,l'})$ where:
  - if $l' = m+1$, we define $\phi_{k,m+1}$ as $\phi_{k+1,0}$;
  - $\dagger_1 = \dagger_{m+1} = 1$, and for all values $1 < l'' < m+1$, $\dagger_{l''} = 0^+$.

  In every case above (whether $l' = m+1$ or not), since $\sigma \models \phi$ and given our choice of l' such that the next action takes place during $\iota_{l'}$, we ensure that the next action a_i of e_i belongs to $\phi_{k,l'}$.

This construction of v thus respects at each step the definition of admissibility and thus $\rho \vdash v$. Furthermore, our construction also ensures that, for every v', v'' such that $v = v' \cdot (\dagger, \nu) \cdot v''$, $next_{\phi}(v') = (\dagger, \nu)$. Thus, by definition of the controlled belief automaton, we have that the actions of v can be fired in this order in $\mathcal{B}^{\phi}_{\mathcal{A}}$ and hence the existence of a belief $\mathfrak{b}$ such that $(v, \mathfrak{b})$ is reachable in $\mathcal{B}^{\phi}_{\mathcal{A}}$. Since the sequence v has been built precisely to follow all the strategy's changes that have occured during ρ, we also get that the belief $\mathfrak{b}$ contains the last state of ρ and thus $r = [last(\rho)] \in \mathfrak{b}$.

$\Longleftarrow$ Let ρ be a feasible run in $\mathcal{B}^{\phi}_{\mathcal{A}}$. By definition there is a sequence v such that $\rho \vdash v$ and such that all actions of v can be executed in order in $\mathcal{B}^{\phi}_{\mathcal{A}}$. As claimed before, this sequence is of the form $(0, \phi_{0,0}) \cdot (1, \phi_{0,1}) \cdot (0^+, \phi_{0,2}) \cdots (0^+, \phi_{0,m_0}) \cdot (1, \phi_{1,0}) \cdot (1, \phi_{1,1}) \cdot (0^+, \phi_{1,2}) \cdots (0^+, \phi_{1,m_1})$ where for

[4] Since clock z is reset precisely every time unit, $\mathrm{fr}(\mu_i(z)) = \mathrm{fr}\left(\sum_{j<i} d_j\right)$ and so $\sum_{j<i} d_j \in \mathbb{N}$ iff $\mathrm{fr}(\mu_i(z)) = 0$.

all k, $\phi((k,k+1)) = \phi_{k,1},\dots,\phi_{k,m_k}$ with $m_k \geq 1$. We will now construct a strategy σ such that $\sigma \models \phi$ and ρ is σ-compatible.

Let σ be a strategy such that $\forall k \in \mathbb{N}$:

- $\sigma(k) = \phi([k,k])$;
- if no event of ρ occur in interval $(k,k+1)$ then, with m_k the number of strategy changes for ϕ during this interval, we set $\iota_1 = (k, k+1/m_k)$ and for all $1 < j \leq m_k$, $\iota_j = [k+(j-1)/m_k, k+j/m_k)$. The strategy σ is then defined on each ι_j by $\phi_{k,j}$;
- if one or more action of ρ occur in interval $(k,k+1)$ then we need to constrain the ordered partition so that the resulting σ matches those actions. The construction of v precisely tells to which choice of the meta-strategy correspond each action of ρ. We thus get, for every $j \leq m_k$, a (possibly empty) set of time instants at which transitions occur in ρ for the strategy choice $\phi_{k,j}$. If the set is not empty, we note t_j the first of those instants. If the set is empty, we define an "artificial" t_j to help build the ordered partition afterwards. We denote by t'_j the global time of the last event that occured "before", i.e., associated to a $(\dagger, \phi_{k,j'})$ in v with $j' < j$ (in case no such event exists within $(k,k+1)$, we set $t'_j = k$ and $j' = 0$), by t''_j the global time of the first event that occured "after", i.e., associated to an element $(\dagger, \phi_{k,j''})$ in v with $j'' > j$, (in case no such event exists within $(k,k+1)$, we set $t''_j = k+1$ and $j'' = m_k + 1$), we can then set $t_j = t'_j + (t''_j - t'_j) * (j - j')/(j'' - j')$. This formula gives uniformly distributed artificial t_js. Based on these time instants, we can define an ordered partition $\iota_1, \cdots, \iota_{m_k}$ with $t_{m_k+1} = k+1$:
  - $\iota_1 = (k, t_2)$;
  - for all $1 < j \leq m_k$, $\iota_j = [t_j, t_{j+1})$.

  The strategy σ is then defined on each ι_j by $\phi_{k,j}$.

We have thus defined a strategy $\sigma \models \phi$ (both make the same choices at every integer time k and in the same order in every interval of the form $(k,k+1)$) and such that ρ is σ-compatible, which concludes the proof. ■

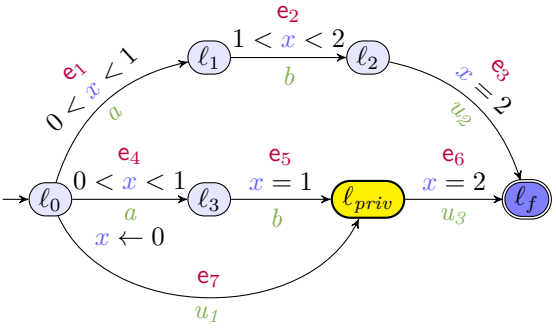


Fig. 11: TA $\mathcal{A}_{counterex}$

Corollary 1. *Let $\mathcal{A}$ be a TA, and ϕ a meta-strategy, $\mathbb{E}^{\phi}_{\mathcal{A}} = \mathbb{I}^{\phi}_{\mathcal{A}}$.*

Proof. Let $\mathcal{A}$ be a TA, and ϕ a meta-strategy. Let $\mathfrak{b}_{\mathcal{I}} \in \mathbb{I}^{\phi}_{\mathcal{A}}$. We focus on the case where there exists $k \in \mathbb{N}$, such that $\mathfrak{b}_{\mathcal{I}} = \mathfrak{b}^{\phi}_{k+}$ (the interval belief associated to $\mathcal{I} = (k,k+1)$) as the case where $\mathfrak{b}_{\mathcal{I}} = \mathfrak{b}^{\phi}_{k}$ (the interval belief associated to $\{k\}$) is simpler and can be obtained with a similar proof. Let $(v_0,\mathfrak{b}_0),\dots(v_m,\mathfrak{b}_m)$ be the sequence used in Definition 17 where $v_1,\dots v_m$ all contain $2k+1$ elements of the form $(1,\mathfrak{E})$. Let $\mathfrak{b}_{\mathcal{E}} = \bigcup_{i=1}^{m} \mathfrak{b}_i \in \mathbb{E}^{\phi}_{\mathcal{A}}$. Let us show that $\mathfrak{b}_{\mathcal{E}} = \mathfrak{b}_{\mathcal{I}}$.

Given $r \in \mathfrak{b}_{\mathcal{I}}$, then by definition of $\mathbb{I}^{\phi}_{\mathcal{A}}$, there exists a run ρ such that ρ is σ-compatible with $\sigma \models \phi$, $[last(\rho)] = r$ and $dur(\rho) \in \mathcal{I}$. Thus, by Lemma 3, ρ is feasible in $\mathcal{B}^{\phi}_{\mathcal{A}}$. By Definition 19, there exists v and $\mathfrak{b}$ such that $\rho \vdash v$, and $(v,\mathfrak{b})$ is reachable in $\mathcal{B}^{\phi}_{\mathcal{A}}$. Moreover, $dur(\rho) \in \mathcal{I}$ is equivalent to v containing exactly $2k+1$ elements of the form $(1,\mathfrak{E})$. Hence, there exists $1 \leq i \leq m$ such that $(v,\mathfrak{b}) = (v_i,\mathfrak{b}_i)$. Hence, $r \in \mathfrak{b}_i \subseteq \mathfrak{b}_{\mathcal{E}}$.

Conversely, given $r \in \mathfrak{b}_{\mathcal{E}}$, there exists $1 \leq i \leq m$ such that $r \in \mathfrak{b}_i$. By following v_i in the Belief automaton, we can build a feasible run ρ admitting v_i and such that $[last(\rho)] = r$. Thus, by Lemma 3, there exists $\sigma \models \phi$, such that ρ is σ-compatible. Again, as v_i contains exactly $2k+1$ elements of the form $(1,\mathfrak{E})$, $dur(\rho) \in \mathcal{I}$. Hence, by definition of $\mathbb{I}^{\phi}_{\mathcal{A}}$, $r \in \mathfrak{b}_{\mathcal{I}}$. ■

5.4 Justifying meta-strategies

The following example shows why the results in [ADLL24] were distorted, and why adding the notion of meta-strategy corrects this.

Example 13. Let $\mathcal{A}_{counterex}$ be the TA depicted in Fig. 11. The upper path, through locations ℓ_1 and ℓ_2, allows public runs of duration of 2 units of time. The lower path, through transition u_1 and location ℓ_{priv}, allows private runs of duration of 2 time units. Finally, the middle path, through transitions a and b, and location ℓ_{priv}, allows private runs of durations in $(2, 3)$.

Because of the reset of x on the transition between ℓ_0 and ℓ_3, finding a strategy that makes $\mathcal{A}_{counterex}$ opaque means finding a strategy that prevent reaching ℓ_{priv} by ℓ_3. An acceptable strategy allows a somewhen in $(0, 1)$ and b somewhen in $(1, 2)$ but prohibits b if a was allowed 1 time unit before.

Then, we can define a strategy σ that makes $\mathcal{A}_{counterex}$ opaque, for example:

$$\sigma(\tau) = \begin{cases} \{a\} & \text{if } \tau \in (0, 0.3) \\ \{b\} & \text{if } \tau \in (1.3, 2) \\ \emptyset & \text{otherwise} \end{cases}$$

Let ϕ the meta-strategy such that $\sigma \models \phi$. Then, the only path in $\mathcal{B}^{\phi}_{\mathcal{A}_{counterex}}$ is $v = (0, \emptyset) \cdot (1, \{a\}) \cdot (0^+, \emptyset) \cdot (1, \emptyset) \cdot (1, \emptyset) \cdot (0^+, \{b\})$.

But this meta-strategy allows to reach a belief containing a final private region but not a public one. Indeed, the run $\rho = (\ell_0, 0), (0.2, \mathsf{e}_4), (\ell_3, 0), (1, \mathsf{e}_5), (\ell_{priv}, 1), (1, \mathsf{e}_6), (\ell_f, 2)$, with $dur(\rho) = 2.2$, is feasible in $\mathcal{B}^{\phi}_{\mathcal{A}_{counterex}}$ but is not σ-compatible (but there exists $\sigma' \models \phi$ such that it is σ'-compatible.)

Finally, the conclusion is that the meta-strategy (named $\mathfrak{b}$-strategy in [ADLL24]) is less precise than the strategy can be, i.e., the meta-strategy can allow behaviours that the strategy prevent, which is why we are working here only with meta-strategies.

□

6 Solving ET-opacity problems through the belief automaton

From Lemma 2 and Corollary 1, a meta-strategy ensures full ET-opacity if it avoids leaking encountered beliefs in the controlled belief automaton. Intuitively, finding such a meta-strategy amounts to solving a one-player game on the belief automaton, where one needs to infinitely often select an action of the form $(1, \mathfrak{E})$ (in order for time to progress) and avoid the leaking beliefs. We will thus translate this into solving a one-player Büchi game.

More precisely, a one-player Büchi game can be defined by a tuple $\mathcal{G} = (Q, q_0, \Sigma, \delta^{\mathcal{G}}, G)$ where Q is a set of states, $q_0 \in Q$ is the initial state, Σ is a set of actions, $\delta^{\mathcal{G}} \subseteq Q \times \Sigma \times Q$ describes the transitions, and $G \subseteq \Sigma$ is a set of "good" actions. Starting from q_0, at each step, the player selects a transition from $\delta^{\mathcal{G}}$ to reach a new state. The player wins if transitions labelled by actions from G are taken infinitely often (note that Büchi games usually require that a set of "good" *states* is visited infinitely often instead of actions, but both frameworks are trivially equivalent).

Lemma 4 ([VW94])**.** *Deciding the existence of a winning strategy in a one-player Büchi game can be done in* NLOGSPACE*. Moreover this strategy, if it exists, can be constructed in polynomial time.*

As shown in [ALL+23], the full ET-opacity problem for timed automata (without control) is decidable in NEXPTIME. The ability to control the system slightly increases the complexity:

Theorem 2. *The full ET-opacity meta-strategy emptiness problem is decidable in* EXPSPACE*; and the full ET-opacity meta-strategy synthesis problem is solvable in* 2EXPTIME*.*

Proof. Given a TA $\mathcal{A}$, using the belief automaton of $\mathcal{A}$ as a basis, we define the one-player Büchi game $\mathcal{G} = ((\mathfrak{S}^{\mathcal{B}_\mathcal{A}})^2, (\bot, \emptyset), \mathfrak{A}^{\mathcal{B}_\mathcal{A}}, \delta_{\mathfrak{d}^{\mathcal{B}_\mathcal{A}}}, G)$ where $G = \{(\dagger, \mathfrak{E}) \in \mathfrak{A}^{\mathcal{B}_\mathcal{A}} \mid \dagger = 1\}$ and $((\mathfrak{b}_1, \mathfrak{b}_2), (\dagger, \mathfrak{E}), (\mathfrak{b}_3, \mathfrak{b}_4)) \in \delta_{\mathfrak{d}^{\mathcal{B}_\mathcal{A}}}$ iff

- $(\mathfrak{b}_1, (\dagger, \mathfrak{E}), \mathfrak{b}_3) \in \mathfrak{d}^{\mathcal{B}_\mathcal{A}}$,
- $\mathfrak{b}_4 = \mathfrak{b}_3$ if $\dagger = 1$, $\mathfrak{b}_4 = \mathfrak{b}_2 \cup \mathfrak{b}_3$ otherwise,
- if $\dagger = 1$, then $\mathfrak{b}_2$ is not a leaking belief for full ET-opacity.

In other words, we manipulate pairs of belief, the first corresponding to the current state of the belief automaton, while the second accumulates the belief. It is important to note that when an action labelled by a 1 is taken, this second component contains the encountered belief corresponding to the current interval. Hence, in $\mathcal{G}$, we cannot leave the current interval if the encountered belief is leaking.

First, assume that a given TA $\mathcal{A}$ is fully ET-opaque for a meta-strategy ϕ. By Lemma 2 and Corollary 1, it means ϕ avoids leaking encountered beliefs in the controlled belief automaton. Hence, it can be applied within $\mathcal{G}$, as in the

controlled belief automaton, and the removal of the transitions from leaking interval beliefs does not affect it. By definition of a meta-strategy, ϕ selects infinitely often an action from G, ensuring the strategy is winning. Conversely, since a winning strategy of $\mathcal{G}$ has to take "good" actions infinitely often, the number of consecutive "non good" actions (matching the changes in the meta strategy within an interval $(k, k+1)$) is finite, and it directly entails a meta-strategy ϕ which, being winning in $\mathcal{G}$, does not encounter a leaking belief. Hence, again by Lemma 2 and Corollary 1, $\mathcal{A}$ is fully ET-opaque with ϕ.

By Lemma 4, the existence of a winning strategy in this game is decidable in NLOGSPACE in the size of the game. As $\mathcal{G}$ is based on the belief automaton, it consists in a form of determinisation of the labelled Region Automaton. The latter is exponential in the size of the TA, and the determinisation can produce a second exponential, hence $\mathcal{G}$ is at most doubly exponential. Hence, solving $\mathcal{G}$ is in EXPSPACE (the non-determinism can freely be removed thanks to Savitch theorem, which implies that EXPSPACE = NEXPSPACE). Moreover, through the computation of a solution in $\mathcal{G}$, one directly obtains that the full ET-opacity meta-strategy synthesis problem is solvable in 2EXPTIME. ■

7 Weak and existential ET-opacity

In this Section as well as in Section 8, we will consider a few variants opacity notions. In most cases, the structure of the proofs are similar to the one for full ET-opacity and as such is not repeated here.

More precisely, in this Section we study two other versions of opacity [ALL+23], namely weak ET-opacity (in which it is harmless that the attacker deduces that the private location was *not* visited) and ∃-ET-opacity (in which we are simply interested in the *existence* of one execution time for which opacity is ensured). In Section 8 we will consider variants relaxing the accuraty of the measure of the attacker.

7.1 Definitions

We recall definitions of weak and existential ET-opacity from [ALL+23].

Definition 20 (Weak ET-opacity)**.** A TA $\mathcal{A}$ is *weakly ET-opaque* when $DVisit^{priv}(\mathcal{A}) \subseteq DVisit^{pub}(\mathcal{A})$. □

Definition 21 (Existential ET-opacity)**.** A TA $\mathcal{A}$ is *existentially ET-opaque* (or *∃-ET-opaque*) when $DVisit^{priv}(\mathcal{A}) \cap DVisit^{pub}(\mathcal{A}) \neq \emptyset$. □

That is, the TA is weakly ET-opaque whenever, for any run of duration d reaching a final location after visiting ℓ_{priv}, there exists another run of the same duration reaching a final location but not visiting the private location. In addition, whenever there is at least one private run such that there exists a public run of the same duration, the TA is ∃-ET-opaque.

Example 14. We have seen in Example 3 that $\mathcal{A}_1$ (given in Fig. 1a) is not fully ET-opaque. However, $\mathcal{A}_1$ is ∃-ET-opaque since, for example, $1 \in DVisit^{priv}(\mathcal{A}) \cap DVisit^{pub}(\mathcal{A}) \neq \emptyset$. Furthermore, since we can reach ℓ_f at any time without visiting ℓ_{priv} (and therefore $DVisit^{pub}(\mathcal{A}_1) = [0, \infty)$), it holds that $DVisit^{priv}(\mathcal{A}_1) \subseteq DVisit^{pub}(\mathcal{A}_1)$ and $\mathcal{A}_1$ is therefore weakly ET-opaque. □

Example 15 (weakly ET-opaque TA)**.** Consider again the TA $\mathcal{A}_{opaque}$ in Fig. 2. Recall from Example 4 that strategy σ_1 is such that $\forall \tau \in \mathbb{R}_{\geq 0}, \sigma(\tau) = \{a\}$, i.e., a is allowed anytime. Recall that $DVisit^{priv}_{\sigma_1}(\mathcal{A}) = \mathbb{N}$ while $DVisit^{pub}_{\sigma_1}(\mathcal{A}) = \mathbb{R}_{\geq 0}$. Therefore $DVisit^{priv}_{\sigma_1}(\mathcal{A}) \subseteq DVisit^{pub}_{\sigma_1}(\mathcal{A})$, and hence $\mathcal{A}_{opaque}$ is weakly ET-opaque with σ_1. □

Strategies and weak and existential ET-opacity

We lift Definition 7 to weak and existential ET-opacity.

Definition 22 (Weak and existential ET-opacity with a strategy)**.** Given a strategy σ, a TA $\mathcal{A}$ is *weakly ET-opaque with* σ whenever $DVisit^{priv}_{\sigma}(\mathcal{A}) \subseteq DVisit^{pub}_{\sigma}(\mathcal{A})$.

In addition, $\mathcal{A}$ is *∃-ET-opaque with* σ whenever $DVisit^{priv}_{\sigma}(\mathcal{A}) \cap DVisit^{pub}_{\sigma}(\mathcal{A}) \neq \emptyset$. □

Similarly, we now lift Definition 12 to weak and existential ET-opacity.

Definition 23 (Weak and existential ET-opacity with a meta-strategy)**.** Given a meta-strategy ϕ, a TA $\mathcal{A}$ is *weakly ET-opaque with* ϕ whenever $DVisit^{priv}_{\phi}(\mathcal{A}) \subseteq DVisit^{pub}_{\phi}(\mathcal{A})$.

In addition, $\mathcal{A}$ is *∃-ET-opaque with* ϕ whenever $DVisit^{priv}_{\phi}(\mathcal{A}) \cap DVisit^{pub}_{\phi}(\mathcal{A}) \neq \emptyset$. □

Problems

We consider the following variations of the emptiness problems defined in Section 3.

Weak ET-opacity (resp. ∃-ET-opacity) meta-strategy emptiness problem:
INPUT: A TA $\mathcal{A}$
PROBLEM: Decide the emptiness of the set of meta-strategies ϕ such that $\mathcal{A}$ is weakly ET-opaque (resp. ∃-ET-opaque) with ϕ.

Similarly, we define as follows their synthesis counterpart:

Weak ET-opacity (resp. ∃-ET-opacity) meta-strategy synthesis problem:
INPUT: A TA $\mathcal{A}$
PROBLEM: Synthesize a meta-strategy ϕ such that $\mathcal{A}$ is weakly ET-opaque (resp. ∃-ET-opaque) with ϕ.

7.2 Results for ∃-ET-opacity

First, we focus on ∃-ET-opacity for which control, as understood here, is useless. Indeed, the strategy of the controller can only *prevent* some behaviour, i.e., remove possible executions. However, ∃-ET-opacity wonders whether there *exists* an opaque time in the TA, so adding a controller (that only removes execution times) cannot change the result. The following theorem proves this claim.

Theorem 3. *Let $\mathcal{A}$ be a TA. $\mathcal{A}$ is ∃-ET-opaque iff there exists a strategy σ such that $\mathcal{A}$ controlled by strategy σ is ∃-ET-opaque.*

Proof. $\Longrightarrow$ Assume that $\mathcal{A}$ is ∃-ET-opaque. Therefore, a strategy enabling all controllable actions at all times does not restrict the behaviour, and therefore the controlled TA remains ∃-ET-opaque. Concretely, let σ be such that $\forall \tau \in \mathbb{R}_{\geq 0} : \sigma(\tau) = \Sigma_c$. Then, $DVisit^{priv}_{\sigma}(\mathcal{A}) = DVisit^{priv}(\mathcal{A})$ and $DVisit^{pub}_{\sigma}(\mathcal{A}) = DVisit^{pub}(\mathcal{A})$, and hence because $\mathcal{A}$ is ∃-ET-opaque, $DVisit^{priv}(\mathcal{A}) \cap DVisit^{pub}(\mathcal{A}) \neq \emptyset \implies DVisit^{priv}_{\sigma}(\mathcal{A}) \cap DVisit^{pub}_{\sigma}(\mathcal{A}) \neq \emptyset$, hence the $\mathcal{A}$ controlled by strategy σ is ∃-ET-opaque.

$\Longleftarrow$ Let $\mathcal{A}$ be a TA and σ a strategy such that $\mathcal{A}$ controlled by strategy σ is ∃-ET-opaque. From Definition 22, we have $DVisit^{priv}_{\sigma}(\mathcal{A}) \cap DVisit^{pub}_{\sigma}(\mathcal{A}) \neq \emptyset$. That is, there exists a duration d such that there exist a private run ρ and a public run ρ', both of duration d. Let $\rho = (\ell_0, \vec{0}), (d_0, \mathsf{e}_0), \cdots, (d_{n-1}, \mathsf{e}_{n-1}), (\ell_n, \mu)$. For all $0 \leq i < n$, $e_i = (\ell_i, g_i, a_i, R_i, \ell'_i)$, with $a_i \in \sigma(\sum_{j=0}^{j\leq i} d_j) \cup \Sigma_u$. As $\sigma(\sum_{j=0}^{j\leq i} d_j) \cup \Sigma_u \subseteq \Sigma_c \cup \Sigma_u = \Sigma$, the action a_i is available when $\mathcal{A}$ is not controlled and $\rho \in Visit^{priv}(\mathcal{A})$. With the same reasoning for ρ', we have that $\rho' \in Visit^{pub}(\mathcal{A})$—and therefore $\mathcal{A}$ is ∃-ET-opaque. ■

The same reasoning can apply to non-finitely-varying strategies, as well as to meta-strategies:

Corollary 2. *Let $\mathcal{A}$ be a TA.*

- *$\mathcal{A}$ is ∃-ET-opaque iff there exists a non-finitely-varying strategy σ such that $\mathcal{A}$ controlled by strategy σ is ∃-ET-opaque.*
- *$\mathcal{A}$ is ∃-ET-opaque iff there exists a meta-strategy ϕ such that $\mathcal{A}$ is ∃-ET-opaque with meta-strategy ϕ.*

7.3 Results for weak ET-opacity

We now address weak ET-opacity which can be derived from our analysis of full ET-opacity. We adapt to weak ET-opacity the concept of leaking belief from Definition 14.

Definition 24 (Leaking belief for weak ET-opacity)**.** A belief $\mathfrak{b}$ is *leaking for weak ET-opacity* when

- $(\mathfrak{b} \cap \mathsf{R}^F_{\mathcal{A}} \cap \mathsf{Private}_{\mathcal{A}} \neq \emptyset)$, and
- $(\mathfrak{b} \cap \mathsf{R}^F_{\mathcal{A}} \cap \mathsf{Public}_{\mathcal{A}} = \emptyset)$.

□

We now adapt Lemma 2 to weak ET-opacity.

Lemma 5 (Beliefs characterization for weak ET-opacity)**.** *A TA $\mathcal{A}$ is weakly ET-opaque with a meta-strategy ϕ whenever, for all $\mathfrak{b} \in \mathbb{I}^{\phi}_{\mathcal{A}}$, $\mathfrak{b}$ is not leaking for weak ET-opacity.*

Proof. This proof can be achieved similarly to the proof of Lemma 2. ■

Theorem 4. *The weak ET-opacity meta-strategy emptiness problem is decidable in* EXPSPACE*; and the weak ET-opacity meta-strategy synthesis problem is solvable in* 2EXPTIME*.*

Proof. This proof can be achieved similarly to the proof of Theorem 2, using Lemma 5 in place of Lemma 2. ■

8 Extension: robust definitions of ET-opacity

So far, the attacker needed to measure the execution time with an infinite precision—this is often unrealistic in practice [DWDMR04, BMS13]. We therefore consider variants of opacity where intervals of non-opaque execution times can be considered acceptable as long as they are hard to detect, for instance by being of size 0, i.e., reduced to a point (note that there can be an infinite number of them).

In order to formally define these new notions, we introduce new notations: given a set S, then let $[\![S]\!]$ denote the *closure* of S (i.e., the smallest closed set containing S) and let $(\!|S|\!)$ denote the *interior* of S (i.e., the largest open set contained in S). Let $\oplus$ denotes the exclusive `OR` operator such that, for two sets A and B, $A \oplus B = \{v \mid v \in (A \cup B) \setminus (A \cap B)\}$.

We introduce two new notions of opacity:

1. *almost full ET-opacity*, where every punctual opacity violation is ignored, and
2. *closed full ET-opacity*, where a punctual violation is ignored only if it is followed or preceded by an opaque interval.

8.1 Definitions and problems

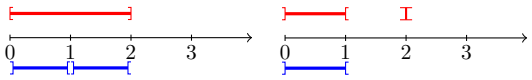
8.1.1 almost full ET-opacity

Let us first define *almost full ET-opacity*, where every punctual opacity violation is ignored. That is, a TA is almost fully ET-opaque whenever all non-opaque durations are isolated from each other; that is, since these non-opaque durations must be punctual, then the interior of intervals of non-opaque durations must be empty.

Definition 25 (Almost full ET-opacity)**.** A TA $\mathcal{A}$ is almost fully ET-opaque when $(\!|DVisit^{priv}(\mathcal{A}) \oplus DVisit^{pub}(\mathcal{A})|\!) = \emptyset$. □

8.1.2 closed full ET-opacity

Let us now define *closed full ET-opacity*, where a punctual violation is ignored only if it is followed or preceded by an opaque interval. That is, we say a TA is closed fully ET-opaque when the closure of the private durations equals the closure of the public durations.



(a) (b)

Fig. 12: Private (above in red) and public (below in blue) durations in $\mathcal{A}_2$ (a) and $\mathcal{A}_3$ (b)

Definition 26 (Closed full ET-opacity)**.** A TA $\mathcal{A}$ is closed fully ET-opaque when $[\![DVisit^{priv}(\mathcal{A})]\!] = [\![DVisit^{pub}(\mathcal{A})]\!]$. □

A difference between both definitions is, for example, whenever a non-opaque duration is such that the immediately neighbouring durations do not correspond to any accepting run (neither public nor private). In that case, this non-opaque duration will be left out by almost full ET-opacity, but will still be considered non-opaque by closed full ET-opacity.

We can consider Almost full ET-opacity and Closed full ET-opacity with a meta-strategy ϕ by replacing $DVisit^{priv}(\mathcal{A})$ by $DVisit^{priv}_{\phi}(\mathcal{A})$ and $DVisit^{pub}(\mathcal{A})$ by $DVisit^{pub}_{\phi}(\mathcal{A})$.

8.1.3 Example

Example 16. Let $\mathcal{A}_2$ be a TA such that $DVisit^{priv}(\mathcal{A}_2) = [0,2]$ and $DVisit^{pub}(\mathcal{A}_2) = (0,1) \cup (1,2)$. $\mathcal{A}_2$ is not fully ET-opaque (but it is weakly ET-opaque). But $(\!|DVisit^{priv}(\mathcal{A}_2) \oplus DVisit^{pub}(\mathcal{A}_2)|\!) = (\!|\{0\} \cup \{1\} \cup \{2\}|\!) = \emptyset$, so $\mathcal{A}_2$ is almost fully ET-opaque. Note that look at the interior of private and public interval would not be equivalent: $(\!|DVisit^{priv}(\mathcal{A}_2)|\!) \neq (\!|DVisit^{pub}(\mathcal{A}_2)|\!)$ as $(0,2) \neq ((0,1) \cup (1,2))$. Moreover, $[\![DVisit^{priv}(\mathcal{A}_2)]\!] = [\![DVisit^{pub}(\mathcal{A}_2)]\!] = [0,2]$ so $\mathcal{A}_2$ is closed fully ET-opaque.

Now, let $\mathcal{A}_3$ be a TA such that $DVisit^{priv}(\mathcal{A}_3) = (0,1) \cup \{2\}$ and $DVisit^{pub}(\mathcal{A}_3) = (0,1)$. $\mathcal{A}_3$ is not fully ET-opaque. $(\!|DVisit^{priv}(\mathcal{A}_3) \oplus DVisit^{pub}(\mathcal{A}_3)|\!) = \emptyset$ so $\mathcal{A}_3$ is almost fully ET-opaque. $[\![DVisit^{priv}(\mathcal{A}_3)]\!] = [0,1] \cup \{2\} \neq [\![DVisit^{pub}(\mathcal{A}_3)]\!] = [0,1]$ so $\mathcal{A}_3$ is not closed fully ET-opaque. □

8.1.4 Problems

We are interested in the same problems as before, this time in the context of closed ET-opacity or almost ET-opacity. Formally:

closed full ET-opacity (resp. almost full ET-opacity) meta-strategy emptiness problem:
INPUT: A TA $\mathcal{A}$
PROBLEM: Decide the emptiness of the set of meta-strategies ϕ such that $\mathcal{A}$ is closed fully ET-opaque (resp. almost fully ET-opaque) with ϕ.

closed full ET-opacity (resp. almost full ET-opacity) meta-strategy synthesis problem:
INPUT: A TA $\mathcal{A}$
PROBLEM: Synthesize a meta-strategy ϕ such that $\mathcal{A}$ is closed fully ET-opaque (resp. almost fully ET-opaque) with ϕ.

8.2 Characterization

A single belief is not sufficient to characterize a TA that is not almost ET-opaque (resp. closed). Indeed, suppose a time t such that $\mathfrak{b}_t$ is leaking for full ET-opacity. This means that a punctual violation of opacity exists. This kind of violation can be allowed in the context of almost and closed full ET-opacity. It is problematic if the times around it are also a violation of opacity.

More specifically, a violation to almost full ET-opacity corresponds to a succession of leaking beliefs, i.e., every punctual violation is ignored. On the other hand, a violation to closed full ET-opacity corresponds either to a succession of leaking beliefs, or to a unique leaking belief surrounded by beliefs that do not contain any final region. Intuitively, a punctual violation is ignored if it belongs to an interval where private and public final states can be reached.

When considering meta-strategies, this issue is partially lifted as the behaviour of the system within an open interval is the same: given $k \in \mathbb{N}$, and $t, t' \in (k, k+1)$, $\mathfrak{b}^\phi_t = \mathfrak{b}^\phi_{t'}$. This a consequence of the shrinking argument within the proof of Lemma 2. As a consequence, only a belief representing an interval can be leaking for almost full ET-opacity.

We define formally leaking belief for almost full ET-opacity and closed full ET-opacity for a meta-strategy as follows.

Definition 27 (Leaking belief for almost full ET-opacity)**.** Let ϕ a meta-strategy, a belief $\mathfrak{b} \in \mathbb{I}^\phi_{\mathcal{A}}$ is leaking for almost full ET-opacity when it is leaking for full ET-opacity and there is $k \in \mathbb{N}$ such that $\mathfrak{b} = \mathfrak{b}_{k+}$. □

Definition 28 (Leaking belief for closed full ET-opacity)**.** Let ϕ a meta-strategy, a belief $\mathfrak{b} \in \mathbb{I}^\phi_{\mathcal{A}}$ is leaking for closed full ET-opacity when either:

- it is leaking for almost ET-opacity, or
- there is $k \in \mathbb{N}$ such that $\mathfrak{b} = \mathfrak{b}_k$, and
  - $\mathfrak{b}_{(k-1)+} \cup \mathsf{R}^F_{\mathcal{A}} = \emptyset$,
  - $\mathfrak{b}_{(k+1)+} \cup \mathsf{R}^F_{\mathcal{A}} = \emptyset$ and
  - $\mathfrak{b}$ is leaking for full ET-opacity

□

As previously, the almost full ET-opacity and closed full ET-opacity of a TA can be characterized thanks to the interval beliefs. Both lemma can be achieved with a proof similar to the proof of Lemma 2.

Lemma 6 (Beliefs characterization for almost full ET-opacity)**.** *A TA $\mathcal{A}$ is almost fully ET-opaque with a meta-strategy ϕ iff, for all $\mathfrak{b} \in \mathbb{I}^\phi_{\mathcal{A}}$, $\mathfrak{b}$ is not leaking for almost full ET-opacity.*

Lemma 7 (Beliefs characterization for closed full ET-opacity)**.** *A TA $\mathcal{A}$ is closed fully ET-opaque with a meta-strategy ϕ iff, for all $\mathfrak{b} \in \mathbb{I}^\phi_{\mathcal{A}}$, $\mathfrak{b}$ is not leaking for closed full ET-opacity.*

8.3 Results

We can conclude positively for the problems on closed full ET-opacity and almost full ET-opacity.

As for full ET-opacity, this result is due to the equivalence between finding a meta-strategy for a TA and finding a strategy in a variation of the corresponding belief automaton, and to the fact that such a strategy corresponds to a winning strategy in a one-player Büchi game. Hence, both following result can be achieved similarly to the proof of Theorem 2, using Lemma 7 in place of Lemma 2. Note though that the game that one needs to build when considering closed full ET-opacity is slightly expanded compared to Theorem 2: a Boolean must be included to note whether a singleton is violating opacity and the previous interval belief did not contain a region associated to a final location. This information strengthens the constraint on the next interval belief, requiring that it must contain a region associated to a final location.

Theorem 5. *The almost full ET-opacity finitely-varying controller emptiness problem is decidable; the almost full ET-opacity finitely-varying controller synthesis problem is solvable.*

Theorem 6. *The closed full ET-opacity meta-strategy controller emptiness problem is decidable; the closed full ET-opacity meta-strategy controller synthesis problem is solvable.*

Remark 3. We can extend these notions of almost and closed full ET-opacity to their weak counterparts, with similar results. □

9 Conclusion

We addressed here the control of a system modelled by a TA to make it fully ET-opaque (execution-time opaque). On the one hand, we showed that the strategy emptiness problem is undecidable. On the other hand, we showed that not only the strategy emptiness problem becomes decidable when considering meta-strategies (i.e., in which we specify the order of—a finite number of—strategy changes within interval time units, without fixing their actual changing time), but also we can effectively solve the controller synthesis problem, by building such a controller.

In addition, we studied two other versions of opacity from [ALL+23], namely ∃-ET-opacity (in which we are simply interested in the *existence* of one execution time for which opacity is ensured), and weak ET-opacity (in which it is harmless that the attacker deduces that the private location was *not* visited).

We also addressed two extensions (closed full ET-opacity and almost full ET-opacity) which can relate to a *robust* setting where the attacker cannot have an infinite precision.

Future works

A natural next step will be to introduce timing parameters *à la* [ALL+23], and address control in that setting.

Addressing the control for the definition of opacity (based on languages) as in [Cas09] would be interesting in two settings:

1. the general setting, where the controller synthesis will be undecidable but may terminate for some semi-algorithms, and
2. decidable subclasses that remain to be exhibited, presumably one-clock TAs, as in [ADL24].

Moreover, an analysis of the strategy obtained to ensure opacity might lead to only a static modification of the structure (e.g., deletion of a transition)—which will be interesting to study.

Finally, the implementation of this work is on our agenda. While implementing the beliefs directly would be straightforward, it would probably result in an unnecessary blowup, and therefore an adaptation with structures such as zones [BBLP06] (which does not seem immediate) should be designed.

Acknowledgments

This work is partially supported by ANR BisoUS (ANR-22-CE48-0012) and by ANR TAPAS (ANR-24-CE25-5742).

References

[AA23] Johan Arcile and Étienne André. Timed automata as a formalism for expressing security: A survey on theory and practice. *ACM Computing Surveys*, 55(6):1–36, 2023.

[ABC+25] Étienne André, Jean-Luc Béchennec, Sudipta Chattopadhyay, Sébastien Faucou, Didier Lime, Dylan Marinho, Olivier H. Roux, and Jun Sun. Verifying timed properties of programs in IoT nodes using parametric time Petri nets. Technical report, March 2025.

[ABLM22] Étienne André, Shapagat Bolat, Engel Lefaucheux, and Dylan Marinho. strategFTO: Untimed control for timed opacity. In *FTSCS*, pages 27–33. ACM, 2022.

[AD94] Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[ADDJI23] Eugene Asarin, Aldric Degorre, Catalin Dima, and Bernardo Jacobo Inclán. Bandwidth of timed automata: 3 classes. In *FSTTCS*, volume 284 of *LIPIcs*, pages 10:1–10:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

[ADL24] Étienne André, Sarah Dépernet, and Engel Lefaucheux. The bright side of timed opacity. In Kazuhiro Ogata, Meng Sun, and Dominique Méry, editors, *ICFEM*, volume 15394 of *Lecture Notes in Computer Science*, pages 51–69. Springer, December 2024.

[ADLL24] Étienne André, Marie Duflot, Laetitia Laversa, and Engel Lefaucheux. Execution-time opacity control for timed automata. In Alexandre Madeira and Alexander Knapp, editors, *SEFM*, volume 15280 of *Lecture Notes in Computer Science*, pages 347–365. Springer, November 2024.

[AETYM21] Ikhlass Ammar, Yamen El Touati, Moez Yeddes, and John Mullins. Bounded opacity for timed systems. *Journal of Information Security and Applications*, 61:1–13, 2021.

[AFH99] Rajeev Alur, Limor Fix, and Thomas A. Henzinger. Event-clock automata: A determinizable class of timed automata. *Theoretical Computer Science*, 211(1-2):253–273, 1999.

[AGW+24] Jie An, Qiang Gao, Lingtai Wang, Naijun Zhan, and Ichiro Hasuo. The opacity of timed automata. In André Platzer, Kristin-Yvonne Rozier, Matteo Pradella, and Matteo Rossi, editors, *FM*, volume 14933 of *Lecture Notes in Computer Science*, pages 620–637. Springer, 2024.

[AHV93] Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. Parametric real-time reasoning. In *STOC*, pages 592–601. ACM, 1993.

[AK20] Étienne André and Aleksander Kryukov. Parametric non-interference in timed automata. In *ICECCS*, pages 37–42, 2020.

[ALL+23] Étienne André, Engel Lefaucheux, Didier Lime, Dylan Marinho, and Jun Sun. Configuring timing parameters to ensure execution-time opacity in timed automata. In *TiCSA*, volume 392 of *Electronic Proceedings in Theoretical Computer Science*, pages 1–26, 2023. Invited paper.

[ALM23] Étienne André, Engel Lefaucheux, and Dylan Marinho. Expiring opacity problems in parametric timed automata. In *ICECCS*, pages 89–98, 2023.

[ALMS22] Étienne André, Didier Lime, Dylan Marinho, and Jun Sun. Guaranteeing timed opacity using parametric timed model checking. *ACM Transactions on Software Engineering and Methodology*, 31(4):1–36, 2022.

[AMPS98] Eugene Asarin, Oded Maler, Amir Pnueli, and Joseph Sifakis. Controller synthesis for timed automata. *IFAC Proceedings*

Volumes, 31(18):447–452, 1998. Proceedings of the 5th IFAC Conference on System Structure and Control (SSC 1998).

[BBLP06] Gerd Behrmann, Patricia Bouyer, Kim Guldstrand Larsen, and Radek Pelánek. Lower and upper bounds in zone-based abstractions of timed automata. *International Journal on Software Tools for Technology Transfer*, 8(3):204–215, 2006.

[BCLR15] Gilles Benattar, Franck Cassez, Didier Lime, and Olivier H. Roux. Control and synthesis of non-interferent timed systems. *International Journal of Control*, 88(2):217–236, 2015.

[BDR08] Véronique Bruyère, Emmanuel Dall'Olio, and Jean-Francois Raskin. Durations and parametric model-checking in timed automata. *ACM Transactions on Computational Logic*, 9(2):12:1–12:23, 2008.

[BFH+14] Nathalie Bertrand, Eric Fabre, Stefan Haar, Serge Haddad, and Loïc Hélouët. Active diagnosis for probabilistic systems. In Anca Muscholl, editor, *FoSSaCS*, volume 8412 of *Lecture Notes in Computer Science*, pages 29–42. Springer, 2014.

[BFM15] Patricia Bouyer, Erwin Fang, and Nicolas Markey. Permissive strategies in timed automata and games. *Electronic Communication of the European Association of Software Science and Technology*, 72, 2015.

[BFST02] Roberto Barbuti, Nicoletta De Francesco, Antonella Santone, and Luca Tesei. A notion of non-interference for timed automata. *Fundamenta Informaticae*, 51(1-2):1–11, 2002.

[BMS13] Patricia Bouyer, Nicolas Markey, and Ocan Sankur. Robustness in timed automata. In *RP*, volume 8169 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2013. Invited paper.

[BMS15] Béatrice Bérard, John Mullins, and Mathieu Sassolas. Quantifying opacity. *Mathematical Structures in Computer Science*, 25(2):361–403, 2015.

[BT03] Roberto Barbuti and Luca Tesei. A decidable notion of timed non-interference. *Fundamenta Informaticae*, 54(2-3):137–150, 2003.

[Cas09] Franck Cassez. The dark side of timed opacity. In *ISA*, volume 5576 of *Lecture Notes in Computer Science*, pages 21–30. Springer, 2009.

[CHS+22] Aidong Chen, Chen Hong, Xinna Shang, Hongyuan Jing, and Sen Xu. Timing leakage to break SM2 signature algorithm. *Journal of Information Security and Applications*, 67:103210, 2022.

[Dim01] Catalin Dima. Real-time automata. *Journal of Automata, Languages and Combinatorics*, 6(1):3–23, 2001.

[DWDMR04] Martin De Wulf, Laurent Doyen, Nicolas Markey, and Jean-François Raskin. Robustness and implementability of timed automata. In *FORMATS and FTRTFT*, volume 3253 of *Lecture Notes in Computer Science*, pages 118–133. Springer, 2004.

[JIDA22] Bernardo Jacobo Inclán, Aldric Degorre, and Eugene Asarin. Bounded delay timed channel coding. In *FORMATS*, volume 13465 of *Lecture Notes in Computer Science*, pages 65–79. Springer, 2022.

[JT07] Marcin Jurdzinski and Ashutosh Trivedi. Reachability-time games on timed automata. In *ICALP*, volume 4596 of *Lecture Notes in Computer Science*, pages 838–849. Springer, 2007.

[KKG24] Julian Klein, Paul Kogel, and Sabine Glesner. Verifying opacity of discrete-timed automata. In *FormaliSE*, pages 55–65. ACM, 2024.

[LLHL22] Jun Li, Dimitri Lefebvre, Christoforos N. Hadjicostis, and Zhiwu Li. Observers for a class of timed

automata based on elapsed time graphs. *IEEE Transactions on Automatic Control*, 67(2):767–779, 2022.

[Min67] Marvin L. Minsky. *Computation: Finite and infinite machines*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1967.

[Sta10] François-Xavier Standaert. Introduction to side-channel attacks. In *Secure Integrated Circuits and Systems*, Integrated Circuits and Systems, pages 27–42. Springer, 2010.

[VW94] Moshe Y. Vardi and Pierre Wolper. Reasoning about infinite computations. *Information and Computation*, 115(1):1–37, 1994.

[WZ18] Lingtai Wang and Naijun Zhan. Decidability of the initial-state opacity of real-time automata. In *Symposium on Real-Time and Hybrid Systems - Essays Dedicated to Professor Chaochen Zhou on the Occasion of His 80th Birthday*, volume 11180 of *Lecture Notes in Computer Science*, pages 44–60. Springer, 2018.

[WZA18] Lingtai Wang, Naijun Zhan, and Jie An. The opacity of real-time automata. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(11):2845–2856, 2018.

[Zha24] Kuize Zhang. State-based opacity of labeled real-time automata. *Theoretical Computer Science*, 987:114373, 2024.

A Notation table

Timed automaton	
$\mathcal{A}$	a timed automaton
Σ	a finite set of actions of a TA
L	a finite set of locations of a TA
ℓ_0	the initial location of a TA
ℓ_{priv}	the private location of a TA
F	the set of final locations of a TA
$\mathbb{X}$	a finite set of clocks
x_i	the i^{th} clock
H	the number of clocks
$I(\ell)$	the invariant of location ℓ
E	the finite set of edges of a TA
e	an edge
R	a set of clocks to be reset
g	a guard
μ	a clock valuation
d	a delay
z	extra clock
c_i	largest constant for clock x_i

Semantics	
$TTS_{\mathcal{A}}$	the semantics of TA $\mathcal{A}$
S	the set of states in $TTS_{\mathcal{A}}$
s_0	the initial state in $TTS_{\mathcal{A}}$
s	a state in $TTS_{\mathcal{A}}$
δ	transition function of $TTS_{\mathcal{A}}$
$\overset{\mathsf{e}}{\mapsto}$	a discrete transition with edge e
$\overset{d}{\mapsto}$	a delay transition with delay d
ρ	a run
$last(\rho)$	last state of run ρ

Regions	
r	a region
$[s]$	equivalence class of s
$\mathsf{R}_{\mathcal{A}}$	regions set of $\mathcal{A}$
$\mathsf{R}^F_{\mathcal{A}}$	set of final regions of $\mathcal{A}$
$\mathcal{R}_{\mathcal{A}}$	region automaton of $\mathcal{A}$
Σ^{R}	set of actions of $\mathcal{R}_{\mathcal{A}}$
δ^{R}	transition function of $\mathcal{R}_{\mathcal{A}}$
$\rightarrow_{\mathsf{R}}$	a transition in $\mathcal{R}_{\mathcal{A}}$

Opacity	
$Visit^{priv}(\mathcal{A})$	set of private runs of $\mathcal{A}$
$Visit^{pub}(\mathcal{A})$	set of public runs of $\mathcal{A}$
$DVisit^{priv}(\mathcal{A})$	set of durations of private runs of $\mathcal{A}$
$DVisit^{pub}(\mathcal{A})$	set of durations of public runs of $\mathcal{A}$
$DVisit^{priv}_{\sigma}(\mathcal{A})$	set of durations of private and σ-compatible runs
$DVisit^{pub}_{\sigma}(\mathcal{A})$	set of durations of public and σ-compatible runs
$DVisit^{priv}_{\phi}(\mathcal{A})$	set of durations of private and σ-compatible runs, $\sigma \models \phi$
$DVisit^{pub}_{\phi}(\mathcal{A})$	set of durations of public and σ-compatible runs, such that $\sigma \models \phi$

Strategies	
Σ_c	controllable actions
Σ_u	uncontrollable actions
σ	a strategy
ϕ	a meta-strategy
ν	a strategy within an interval
ι	a subinterval
δ^{σ}	transition function of the semantics of $\mathcal{A}$ controlled by strategy σ
$\mapsto_{\sigma}$	a discrete or delay transition in the semantics of $\mathcal{A}$ controlled by strategy σ
$\xrightarrow{d,e}_{\sigma}$	a transition with delay d and edge e in the semantics of $\mathcal{A}$ controlled by strategy σ
$\mathfrak{E}$	set of activated controllable actions
$\sigma \models \phi$	σ satisfies ϕ

Duplicated TA	
$\mathcal{A}^{dup}$	a duplicated TA
L_{priv}	set of private states
L_{pub}	set of public states
ℓ_{priv}	a private state
$\mathsf{Private}_{\mathcal{A}}$	set of regions reachable on a run visiting ℓ_{priv}
$\mathsf{Public}_{\mathcal{A}}$	set of regions not reachable on a run visiting ℓ_{priv}

Beliefs	
$\mathfrak{b}_t$	belief for time t
$\mathfrak{b}_{k+}$	belief for the interval $(k, k+1)$
$0, 0^+, 1, \dagger$	evolution of $\mathrm{fr}(z)$
$\mathcal{B}_{\mathcal{A}}$	belief automaton of $\mathcal{A}$
$\mathfrak{S}^{\mathcal{B}_{\mathcal{A}}}$	states of $\mathcal{B}_{\mathcal{A}}$
$\mathfrak{A}^{\mathcal{B}_{\mathcal{A}}}$	actions of $\mathcal{B}_{\mathcal{A}}$
v	a sequence of actions of $\mathcal{B}_{\mathcal{A}}$
$\rho \vdash v$	ρ admits v
$\mathfrak{d}^{\mathcal{B}_{\mathcal{A}}}$	transition function of $\mathcal{B}_{\mathcal{A}}$
$\mathbb{I}^{\phi}_{\mathcal{A}}$	set of interval beliefs reachable by a meta-strategy ϕ
$\mathcal{B}^{\phi}_{\mathcal{A}}$	controlled belief automaton of $\mathcal{A}$ and meta-strategy ϕ
$\mathfrak{S}^{\mathcal{B}^{\phi}_{\mathcal{A}}}$	states of controlled belief automaton
$\mathfrak{A}^{\mathcal{B}^{\phi}_{\mathcal{A}}}$	actions of controlled belief automaton
$\mathfrak{d}^{\mathcal{B}^{\phi}_{\mathcal{A}}}$	transition function of controlled belief automaton
$\mathbb{E}^{\phi}_{\mathcal{A}}$	set of beliefs encountered by $\mathcal{B}^{\phi}_{\mathcal{A}}$