Opacity problems in multi-energy timed automata*

Étienne André
Nantes Université, CNRS, LS2N, Nantes, France
Institut universitaire de France (IUF)
Lydia Bakiri
Université Sorbonne Paris Nord, LIPN, CNRS UMR 7030, F-93430 Villetaneuse, France
LIX, CNRS, École polytechnique, Institut Polytechnique de Paris, Palaiseau, France

Abstract

Cyber-physical systems can be subject to information leakage; in the presence of continuous variables such as time and energy, these leaks can be subtle to detect. We study here the verification of opacity problems over systems with observation over both timing and energy information. We introduce guarded multi-energy timed automata as an extension of timed automata with multiple energy variables and guards over such variables. Despite undecidability of this general formalism, we establish positive results over a number of subclasses, notably when the attacker observes the final energy and/or the execution time, but also when they have access to the value of the energy variables every time unit.

Keywords: verification, security properties, opacity, timed systems

1 Introduction

Modelling and verifying real-time systems require formal models that can accurately represent timing constraints. *Timed automata* (TAs) [AD94] became a foundational model for specifying and analysing such systems. TAs extend finite automata with real-valued clocks, enabling the representation of time-dependent behaviours.

However, in addition to timing, many cyber-physical systems are also subject to quantitative constraints, particularly related to the consumption or accumulation of resources like energy, memory, or cost. To incorporate such quantitative considerations, various extensions of timed automata have been proposed. *Priced timed automata* (also known as *weighted* or *cost timed automata*) augment the model with cost variables that accumulate according to rates and discrete updates [BFH+01, BLR05]. More specifically, *energy timed automata* model the dynamic consumption and replenishment of energy in systems (e.g., [BFL+08, BBF+21, CHL24]). These extensions enable the specification and verification of energy-aware properties, such as "the system can always recharge before the battery is depleted" or "the total energy cost never exceeds a budget".

Opacity The concept of *opacity* [Maz04, BKMR08, LLH18] pertains to the potential leakage of information from a system to an attacker. Specifically, it captures the attacker's ability to infer secret information based on publicly observable behaviours. A system is considered opaque if an attacker with access to only a subset of the system's observable actions cannot determine whether a specific sequence of actions has taken place. Time particularly influences the deductive capabilities of the attacker. It has been shown in [GMR07] that it is possible for models that are opaque when timing constraints are omitted, to become non-opaque when those constraints are added to the model.

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. The aforementioned negative result leaves hope only if the definition or the setting is changed. First, in [WZ18, WZA18], the input model is simplified to *real-time automata*. [LLHL22] exhibits decidability results for constant-time labelled automata. In [Zha24], Zhang studies decidability for labelled real-timed automata.

Second, in [AGW+24, ADL24, KKG24], decidability results are exhibited in the setting of Cassez' definition, but with restrictions in the model: one-clock automata, one-action automata, or over discrete time. 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. A somehow similar framework is considered in [SLR23], in which the attacker has

[0]This is the author version (extended with all proofs) of the manuscript of the same name published in the proceedings of the 41st ACM/SIGAPP Symposium on Applied Computing (SAC 2026).

a bounded memory, and a finite duration between distinct observations is required, in which case the problem is decidable and PSPACE-complete.

Fourth, in [ALL+23], an alternative definition is proposed, by studying 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. Control is studied in [ABLM22, ADKLL24].

Regarding non-interference for TAs, some decidability results are proved in [BFST02, BT03], while control was considered in [BCLR15] and a parametric timed extension in [AK20]. General security problems for TAs are surveyed in [AA23].

Contributions Here, we extend execution-time opacity [ALL+23] by adding the observation of energy. Our attacker model is as follows: the attacker knows the input model, and observes only final energy and/or execution time; from this observation, they aim at deducing whether a special secret location was visited. Our first contribution is to define a general framework for opacity in timed automata extended with multiple energy variables and energy constraints—a formalism called *guarded multi-energy timed automaton* (guarded META). In this general setting, energy can continuously increase or decrease, and be subject to discrete updates, thus encoding concepts such as battery or temperature; energy constraints can also be part of our model, encoding behaviours such as "when the battery level decreases below some threshold, the system enters degraded mode". Our second and presumably main contribution is to study opacity (with an observation of either final energy, or final time and energy) in *discrete* guarded METAs, in which energy is updated only via discrete updates (in an entirely continuous, real-time setting). The general subclass is undecidable without surprise (as discrete updates can simulate a 2-counter machine); we therefore exhibit several decidable subclasses. Our third contribution is to study a setting with a stronger attacker, able to observe the energy level at each time unit, and for which we also exhibit decidable subclasses. We finally identify a first decidable subclass with continuous energy rates in addition to the discrete updates. Our proofs rely on different techniques, notably by defining modified versions of the input META and/or of the region graph, and subsequently studying (untimed) regular or context-free language problems. To the best of our knowledge, this work is the first to define and study opacity problems over TAs extended with multiple energy variables.

Outline Section 2 introduces the necessary material. Section 3 introduces opacity problems in guarded METAs. Section 4 introduces common constructions for our subsequent proofs. Section 5 studies opacity in discrete guarded METAs. Section 6 studies opacity with an observation of the energy level every time unit. Section 7 exhibits a preliminary decidable subclass for non-necessarily discrete models, i.e., with multiple energy rates in addition to the discrete updates. Section 8 concludes and highlights perspectives.

2 Preliminaries

Let $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}_+$, $\mathbb{R}_{\geq 0}$, $\mathbb{R}$ denote the sets of non-negative integers, integers, non-negative rationals, non-negative reals, and reals respectively. Let $\mathbb{D} \subseteq \mathbb{R}$. Given a finite variables set V, a $\mathbb{D}$-valuation over V is a function from V to $\mathbb{D}$.

Throughout this paper, we assume a set $\mathcal{X} = \{x_1, \ldots, x_H\}$ of *clocks*, i.e., non-negative real-valued variables that evolve at the same rate. A *clock valuation* is an $\mathbb{R}_{\geq 0}$-valuation over $\mathcal{X}$.

We assume a set $\mathcal{E} = \{\eta, \ldots, \eta_M\}$ of *energy variables*, i.e., non-negative real-valued variables that evolve at potentially different rates. An *energy valuation* is an $\mathbb{R}_{\geq 0}$-valuation over $\mathcal{E}$.

We write $\vec{0}_{\mathcal{X}}$ for the valuation such that for all $x \in \mathcal{X}$, $\vec{0}_{\mathcal{X}}(x) = 0$; similarly, $\vec{0}_{\mathcal{E}}$ denotes the valuation s.t. for all $\eta \in \mathcal{E}$, $\vec{0}_{\mathcal{E}}(\eta) = 0$.

Given $R \subseteq \mathcal{X}$, we define the *reset* of a clock valuation w, denoted by $[w]_R$, as follows: $[w]_R(x) = 0$ if $x \in R$, and $[w]_R(x) = w(x)$ otherwise. Given $d \in \mathbb{R}_{\geq 0}$, $w + d$ denotes the clock valuation such that $(w + d)(x) = w(x) + d$, for all $x \in \mathcal{X}$. Similarly, given $d \in \mathbb{R}_{\geq 0}$, and a flow function $\mathit{fl} : \mathcal{E} \to \mathbb{Z}$ assigning each variable with a rate (i.e., the value of its derivative), we define the time elapsing function te as follows: $te(v, \mathit{fl}, d)$ is the valuation such that $\forall \eta \in \mathcal{E} : te(v, \mathit{fl}, d)(v) = v(\eta) + \mathit{fl}(\eta) \times d$.

We denote by U an energy update function, which assigns to each $\eta \in \mathcal{E}$ an offset in $\mathbb{Z}$. We define the *update* of an energy valuation v, denoted by $[v]_U$, as follows: $[v]_U(\eta) = v(\eta) + U(\eta)$.

In the following, we assume $\bowtie \in \{<, \leq, \geq, >\}$. A *simple constraint* g is a constraint over $\mathcal{X} \cup \mathcal{E}$ defined by a conjunction of inequalities of the form $v \bowtie d$, with $v \in \mathcal{X} \cup \mathcal{E}$, and $d \in \mathbb{Z}$. A simple clock constraint (resp. simple energy constraint) is a simple constraint over $\mathcal{X}$ (resp. $\mathcal{E}$).

A clock valuation w *satisfies* a simple clock constraint g, denoted by $w \models g$, if replacing each $x \in \mathcal{X}$ with $w(x)$ within g evaluates to true. We define similarly the satisfaction of simple energy constraints. Given a clock valuation w, an energy valuation v and a simple constraint g, we write $w|v \models g$, if replacing each $x \in \mathcal{X}$ with $w(x)$ and each $\eta \in \mathcal{E}$ with $v(\eta)$ within g evaluates to true.

2.1 Guarded multi-energy timed automata

We first define a very general formalism, extending TAs with multiple energy variables, and guards and invariants involving both energy and clock variables. Our formalism is seemingly more expressive than any formalism from the energy timed automata literature; we will restrict this formalism to various subclasses later. It can also be seen as a subclass of the highly undecidable formalism of *hybrid automata* [ACHH93].

Definition 1. A *guarded multi-energy timed automaton (guarded META)* $\mathcal{A}$ over a set AP of atomic propositions is a tuple $\mathcal{A} = (\Sigma, L, \ell_0, L_{priv}, L_f, \gamma, \mathcal{X}, \mathcal{E}, I, \phi, E)$, where:

1. Σ is a finite set of actions,
2. L is a finite set of locations, $\ell_0 \in L$ is the initial location,
3. $L_{priv} \subseteq L$ is the non-empty set of private locations,
4. $L_f \subseteq L \setminus L_{priv}$ is the non-empty set of final locations,
5. γ is a label function $\gamma : L \to 2^{AP}$,
6. $\mathcal{X}$ is a finite set of clocks, $\mathcal{E}$ is a finite set of energy variables,
7. I is the invariant, assigning to each $\ell \in L$ a simple constraint,
8. ϕ is the energy rate function, assigning to every $\ell \in L$ a flow function assigning to each energy variable a rate in $\mathbb{Z}$, and
9. E is a set of edges $e = (\ell, g, a, R, U, \ell')$ where $\ell, \ell' \in L$ are the source and target locations, g (the "guard") is a simple constraint, $a \in \Sigma \cup \{\varepsilon\}$, $R \subseteq \mathcal{X}$ is a set of clocks to be reset to 0, and U is the energy update function.

W.l.o.g., we assume that final locations do not have outgoing transitions.

Example 1. Fig. 1 models a fictitious drone that can charge and then fly (a single time). When it flies, it can emit light flashes (as in an entertainment drone show). The guarded META in Fig. 1 has one clock x and two energy variables (initially 0): t denotes the temperature of the drone while b denotes the battery percentage. Initially, the drone is on standby. When it starts charging (location "charging"), both its battery and its temperature increase, as seen in the location rates (we use $\dot{t} = 1$ for $\phi(\ell)(t) = 1$); no rate depicted denotes a 0-rate. The drone can also cool down (location "cooling"). When it flies (location "flying"), it consumes battery and the temperature rises continuously. Each time the drone emits a light flash, it consumes 5 battery units and gains 5 temperature units in 0-time, denoted by the transition from "flying" to "flashed". This latter urgent location (in which no time can elapse) serves to encode opacity: it is reachable iff the "*flash*" action occurs.

Energy guards are used to model that

1. whenever the temperature goes higher than 100 degrees, the drone explodes (encoded by the transitions from most locations to "exploded"); and
2. the battery level must remain between 0 and 100. If it reaches 0 during the drone's flight, the mission ends in location "out of battery".

In this model, we will be interested in deciding whether an attacker, by looking at time and energy (e.g., at the end of the execution), can detect that the drone has emitted at least one light flash, i.e., whether the "flashed" location was visited. □

Let us now define the concrete semantics of guarded METAs; this is a straightforward extension of the semantics of timed automata [AD94] and multiple-resource energy timed automata (e.g., [CHL24]).

Definition 2 (Semantics of a guarded META)**.** Given a guarded META $\mathcal{A} = (\Sigma, L, \ell_0, L_{priv}, L_f, \gamma, \mathcal{X}, \mathcal{E}, I, \phi, E)$, the *concrete semantics* of $\mathcal{A}$ is given by the timed transition system $(S, s_0, \to)$, with

- $S = \{(\ell, w, v) \in L \times \mathbb{R}_{\geq 0}^{|\mathcal{X}|} \times \mathbb{R}_{\geq 0}^{|\mathcal{E}|} \mid w|v \models I(\ell)\}$,
- $s_0 = (\ell_0, \vec{0}_{\mathcal{X}}, \vec{0}_{\mathcal{E}})$,
- delay transitions: $\big((\ell, w, v), d, (\ell, w + d, te(v, \phi(\ell), d))\big) \in \to$, with $d \in \mathbb{R}_{\geq 0}$, if $\forall d' \in [0, d], \big(\ell, w + d', te(v, \phi(\ell), d')\big) \in S$;
- discrete transitions: $\big((\ell, w, v), e, (\ell', w', v')\big) \in \to$ if $(\ell, w, v), (\ell', w', v') \in S$, there exists $e = (\ell, g, a, R, U, \ell') \in E$, $w' = [w]_R$, $v' = [v]_U$, and $w|v \models g$.

We assume that $s_0 \in S$, i.e., $\vec{0}_{\mathcal{X}}|\vec{0}_{\mathcal{E}} \models I(\ell_0)$, i.e., the initial clock valuation satisfies the invariant of the initial location.

We refer to the states of the concrete semantics of a guarded META as its *concrete states*. As usual, given two concrete states s, s', we write $s \xrightarrow{a} s'$, for $a \in \mathbb{R}_{\geq 0} \cup E$, instead of $(s, a, s') \in \to$. We also further define relation $\rightarrowtail$ by $\big((\ell, w, v), (d, e), (\ell', w', , v')\big) \in \rightarrowtail$ if $\exists w'' \in \mathbb{R}_{\geq 0}^{|\mathcal{X}|}, v'' \in \mathbb{R}_{\geq 0}^{|\mathcal{E}|}, e \in E, d \in \mathbb{R}_{\geq 0} : (\ell, w, v) \xrightarrow{d} (\ell, w'', v'') \xrightarrow{e} (\ell', w', v')$. A concrete run ρ of a guarded META is an alternating sequence of concrete states of S and edges of the form $(\ell_0, w_0, v_0) \overset{(d_0,e_0)}{\rightarrowtail} (\ell_1, w_1, v_1) \overset{(d_1,e_1)}{\rightarrowtail} \cdots \overset{(d_{m-1},e_{m-1})}{\rightarrowtail} (\ell_m, w_m, v_m)$, such that for all $i = 0, \ldots, m-1$, $d_i \in \mathbb{R}_{\geq 0}$, $e_i \in E$, and $\big((\ell_i, w_i, v_i), (d_i, e_i), (\ell_{i+1}, w_{i+1}, v_{i+1})\big) \in \rightarrowtail$. We say that states (ℓ_i, w_i, v_i), for $i = 0, \ldots, m$, *belong* to ρ. The duration of such a run ρ is $dur(\rho) = \sum_{0 \leq i \leq m-1} d_i$.

The *final energies* of ρ, denoted by $FE(\rho)$, is the energy valuation v_m. Given a concrete state $s = (\ell, w, v)$, we say that s is reachable (or that $\mathcal{A}$ reaches s) if s belongs to a run of $\mathcal{A}$. By extension, a location ℓ is reachable if there exists w and v such that (ℓ, w, v) is reachable. We extend the notion of timed language

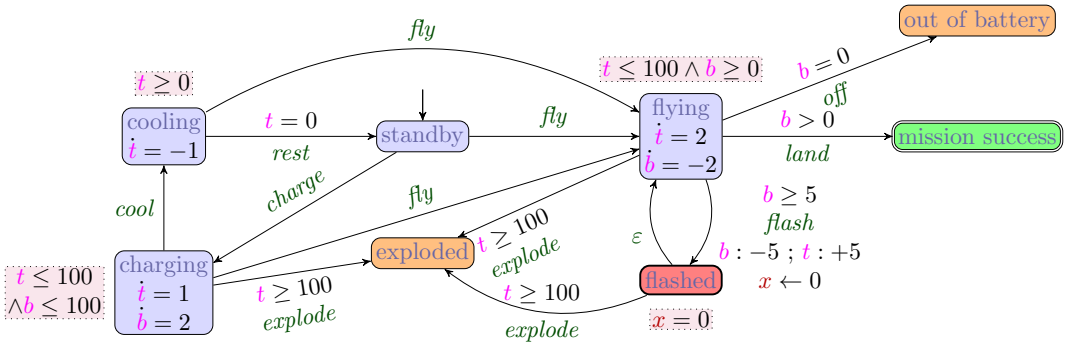


Figure 1: Example of a guarded-META: drone emitting light flashes

of TAs to guarded METAs in a straightforward manner, as the set of accepting timed word, i.e., runs whose last state is a final location then projected onto actions and time durations, i.e., keeping only the sequences $(\sum_{j=0}^{i} d_j, a_i)$ for $0 \leq i \leq m-1$, where a_i is the action of edge e_i, and removing ε transitions in an appropriate manner. Two guarded METAs are *equivalent* if their timed language is the same.

Example 2. Fig. 1 accepts the following run, where $(\text{flying}, 15, [5, \vec{20}])$ is a shorthand for $(\text{standby}, w, v)$ such that $w(x) = 15$, $v(t) = 5$ and $v(b) = 20$: $(\text{standby}, 0, [0, \vec{0}]) \overset{(0,charge)}{\longmapsto} (\text{charging}, 0, [0, \vec{0}]) \overset{(10,cool)}{\longmapsto} (\text{cooling}, 10, [10, \vec{20}]) \overset{(5,fly)}{\longmapsto} (\text{flying}, 15, [5, \vec{20}]) \overset{(2,flash)}{\longmapsto} (\text{flashed}, 0, [14, \vec{11}]) \overset{(0,\varepsilon)}{\longmapsto} (\text{flying}, 0, [14, \vec{11}]) \overset{(0,land)}{\longmapsto} (\text{mission success}, 0, [14, \vec{11}])$. The duration of this run is $10 + 5 + 2 = 17$, and the final energy values are 14 for the temperature and 11 for the battery. The associated timed word is $(0, charge)(10, cool)(15, fly)(17, flash)(17, land)$.

Subclasses

A guarded META is *positive* whenever $\forall \ell \in L, \forall \eta \in \mathcal{E} : \phi(\ell)(\eta) \in \mathbb{N}$ and $\forall (\ell, g, a, R, U, \ell') \in E, \forall \eta \in \mathcal{E} : U(\eta) \in \mathbb{N}$. A guarded META is *discrete* whenever $\forall (\ell, g, a, R, U, \ell') \in E, \forall \eta \in \mathcal{E} : U(\eta) = 0$, i.e., energy variables are only updated along discrete edges. A *META*[1] is a guarded META without energy guards and invariants, i.e., all guards and invariants are simple clock constraints. An *energy timed automaton (ETA)* [CHL24] is a META with $|\mathcal{E}| = 1$. A *timed automaton* (TA) [AD94] is a META such that $\mathcal{E} = \emptyset$.

We assume familiarity with the region automaton of TAs [AD94] (see Section B.2 for a formal definition). We denote by $\mathcal{RA}(\mathcal{A})$ the region automaton of a TA $\mathcal{A}$, which is a non-deterministic finite automaton (NFA). Its (untimed) language is denoted by $\mathcal{L}(\mathcal{RA}(\mathcal{A}))$.

3 Defining opacity in guarded METAs

Given a guarded META $\mathcal{A}$ 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 $(\ell_0, w_0, v_0), (d_0, e_0), (\ell_1, w_1, v_1), \ldots, (\ell_m, w_m, v_m), (d_m, e_m), \ldots, (\ell_n, w_n, v_n)$ for some $m, n \in \mathbb{N}$ such that $\ell_m \in L_{priv}$ and $\ell_n \in L_f$. 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. We denote by $EVisit^{priv}(\mathcal{A})$ the set of all the final energies 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 $(\ell_0, w_0, v_0), (d_0, e_0), (\ell_1, w_1, v_1), \ldots, (\ell_n, w_n, v_n)$ with $\ell_n \in L_f$ and $\forall 0 \leq i \leq n, \ell_i \notin L_{priv}$. We denote the set of those runs by $Visit^{pub}(\mathcal{A})$, referring to them as *public* runs, by $DVisit^{pub}(\mathcal{A})$ the set of all the durations of these public runs, and by $EVisit^{pub}(\mathcal{A})$ the set of all the final energies of these public runs.

$DVisit^{priv}(\mathcal{A})$ and $DVisit^{pub}(\mathcal{A})$ 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 locations L_{priv}.

We extend to guarded METAs the notion of ET-opacity [ALL+23, ADKLL24].

Definition 3 (ET-opacity)**.** A guarded META $\mathcal{A}$ is *fully ET-opaque* when $DVisit^{priv}(\mathcal{A}) = DVisit^{pub}(\mathcal{A})$. It is *weakly ET-opaque* when $DVisit^{priv}(\mathcal{A}) \subseteq DVisit^{pub}(\mathcal{A})$. It is *∃-ET-opaque* when $DVisit^{priv}(\mathcal{A}) \cap DVisit^{pub}(\mathcal{A}) \neq \emptyset$.

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. Weak ET-opacity requires the existence of a public run only if there exists a private run of same duration. Finally, when there exist at least one private and one public run of the same duration, the system is ∃-ET-opaque.

We now define a new notion of opacity w.r.t. the final energy.

Definition 4 (EN-opacity)**.** A guarded META $\mathcal{A}$ is fully opaque w.r.t. energy (*fully EN-opaque*) when $EVisit^{priv}(\mathcal{A}) = EVisit^{pub}(\mathcal{A})$. It is weakly opaque w.r.t. energy (*weakly EN-opaque*) when $EVisit^{priv}(\mathcal{A}) \subseteq EVisit^{pub}(\mathcal{A})$. It is ∃-opaque

[1] In [CHL24], a formalism close to our METAs is called "multi-variable energy TA".

w.r.t. energy ($\exists$-*EN-opaque*) when $EVisit^{priv}(\mathcal{A}) \cap EVisit^{pub}(\mathcal{A}) \neq \emptyset$.

We define similarly *ET-EN-opacity* when the runs meet both execution time and energy conditions (see Definition 10 in Section C.1 for a formal definition).

3.1 Problems

For each $\delta \in \{\exists, \text{weak}, \text{full}\}$ and each $\sigma \in \{\text{EN}, \text{ET-EN}\}$, we define:

δ-σ-opacity problem:
INPUT: A guarded META $\mathcal{A}$
PROBLEM: Is $\mathcal{A}$ δ-σ-opaque?

Example 3. Let us check some opacity problems over the guarded META in Fig. 1. The private run in Example 2 gives final energy values of $[14,\vec{1}1]$. There exists a public run with these same final values: $(\text{standby}, 0, [0,\vec{0}]) \overset{(0,charge)}{\longmapsto} (\text{charging}, 0, [0,\vec{0}]) \overset{(10,cool)}{\longmapsto} (\text{cooling}, 10, [10,\vec{2}0]) \overset{(5,fly)}{\longmapsto} (\text{flying}, 15, [5,\vec{2}0]) \overset{(4.5,land)}{\longmapsto} (\text{mission success}, 19.5, [14,\vec{1}1])$. Therefore, the guarded META is $\exists$-EN-opaque. Actually, it is also weakly-EN-opaque: one can always wait 2.5 time units in the "flying" location instead of flashing, as both actions have the same energy consumption. However, we can easily check that it is not fully-ET-opaque, as the private runs require at least 2.5 time unit of charging.

4 Generic ingredients and proofs

Given a guarded META $\mathcal{A}$, we define two further guarded METAs encoding all public runs (resp. all private runs) of $\mathcal{A}$. The *public runs guarded META* $\mathcal{A}_{pub}$ is obtained very easily from $\mathcal{A}$ by simply removing the private locations L_{priv}. Therefore, all accepting runs are public, as they do not visit L_{priv}.

Inspired by techniques from [ADL24], given a guarded META $\mathcal{A}$, the *private runs guarded META* $\mathcal{A}_{priv}$ is obtained by duplicating all locations and transitions of $\mathcal{A}$: one copy $\mathcal{A}^V$ corresponds to the paths that already visited a private location, while the other copy $\mathcal{A}^{\bar{V}}$ corresponds to the paths that did not (this is a usual way to encode a Boolean, here "L_{priv} was visited", in the locations of a guarded META). That is, for each location ℓ of $\mathcal{A}$, one location ℓ^V is defined in $\mathcal{A}^V$, and one location $\ell^{\bar{V}}$ is defined in $\mathcal{A}^{\bar{V}}$. Then, we redirect all transitions leading to a location $\ell^{\bar{V}}_{priv}$ to the copy ℓ^V_{priv} from $\mathcal{A}^V$. The initial location is $\ell^{\bar{V}}_0$ and the final locations are ℓ^V_f. Hence all runs need to go from $\mathcal{A}^{\bar{V}}$ to $\mathcal{A}^V$ before reaching a final location—which requires visiting a private location. See Definitions 12 and 13 in Section D.1 for a formal definition.

Example 4. Fig. 2 gives an example of a guarded META $\mathcal{A}$ with its public and private runs guarded METAs.

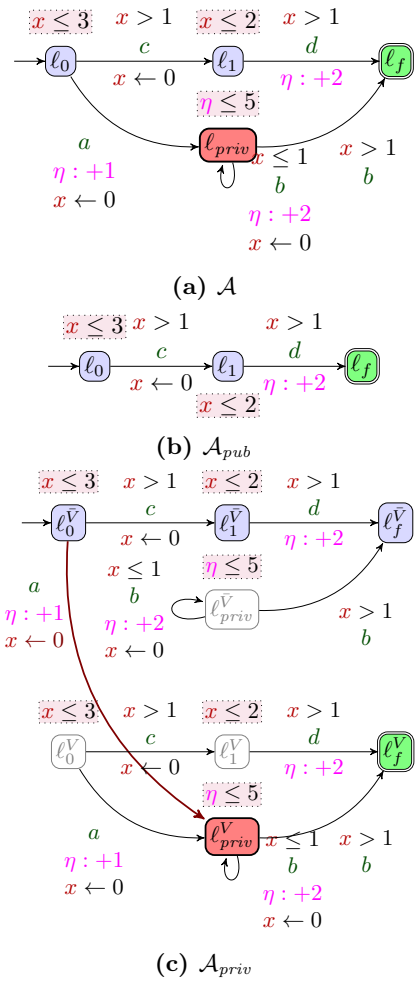


Figure 2: Public and private runs guarded METAs

In some cases, a positive guarded META can be transformed into an equivalent META (without energy guards).

Proposition 1. *Given be a discrete positive guarded META $\mathcal{A}$, $\exists$ a discrete positive META exponentially larger than $\mathcal{A}$ equivalent to $\mathcal{A}$.*

Proof sketch. From the fact that, in a discrete guarded META, the value of energy can only increase; so once it becomes $> M$ (with M denoting the maximum constant in energy guards), its value is not relevant anymore. In an ETA, considering $M+2$ copies of the automaton (0 to M encoding the exact current energy valuation while the last one encodes $> M$) correctly encodes the energy; in METAs, this construction leads to an exponential blowup. See Section D.2. □

5 Opacity problems for discrete METAs

In this section, we consider *discrete* (guarded) METAs, hence in which the evolution of the energy is limited to discrete increments or decrements—while the model remains real-time.

We first rule out guarded METAs in general, because a discrete guarded META with only 2 energies and

1 clock is sufficiently expressive to simulate a 2-counter machine.

Proposition 2. *For each $\delta \in \{\exists, weak, full\}$, and $\sigma \in \{EN, ET\text{-}EN\}$, δ-σ-opacity for discrete guarded METAs with 2 energy variables and 1 clock is undecidable.*

Proof sketch. We first prove that (constant-time) reachability is undecidable in discrete guarded METAs with only 2 energies and a single clock (always 0) (Lemma 1 in Section E.1.2). We then prove that for each $\delta \in \{\exists, \text{weak}, \text{full}\}$, and $\sigma \in \{\text{EN}, \text{ET-EN}\}$, δ-σ-opacity is harder than reachability (see Section E.1.3). □

Thus, we restrict ourselves to a single energy or to only positive rate energies.

5.1 EN-opacity in discrete positive (guarded) METAs

Our first setting is whenever the attacker observes only the final energy, without hint on the actual execution time.

Theorem 1. *For each $\delta \in \{\exists, weak, full\}$, δ-EN-opacity can be decided in* 2EXPSPACE *for discrete positive METAs, and* 3EXPSPACE *for discrete positive guarded METAs.*

Proof. Let us first consider discrete positive METAs without energy guards. We first transform the META into an equivalent model with only energy updates such that $U(\ell)(\eta_i) = 1$ for each energy $\eta_i \in \mathcal{E}$: this can be achieved by duplicating any edge such that $U(\ell)(\eta_i) = N$ (with $N > 1$) into N consecutive edges in 0-time, thanks to a new clock z_0. Then, we relabel actions as follows: any edge such that $U(\ell)(\eta_i) = 1$ is labelled by action inc_i, while any other edge is labelled with ε. We thus obtain a $|\mathcal{E}|$-action non-deterministic TA $\mathcal{A}'$. See Figs. 3a to 3c for an example.

We then build ${\mathcal{A}'}_{pub}$ and ${\mathcal{A}'}_{priv}$. Note that, up to action relabelling and edge duplication, all runs of ${\mathcal{A}'}_{pub}$ (resp. ${\mathcal{A}'}_{priv}$) correspond to the public (resp. private) runs of $\mathcal{A}$, and the number of *inc* actions along a run directly encodes the amount of energy spent.

Our third step is to build the region automaton of these TAs, i.e., $\mathcal{RA}({\mathcal{A}'}_{pub})$ and $\mathcal{RA}({\mathcal{A}'}_{priv})$. These two region automata are non-deterministic finite automata over $\{inc_i, 1 \leq i \leq |\mathcal{E}|\}$.

For example, given the private run $\rho_1 = (\ell_0, 0, 0) \overset{(0.8,a)}{\longmapsto} (\ell_{priv}, 0.8, 0) \overset{(0.3,b)}{\longmapsto} (\ell_{priv}, 1.1, 1) \overset{(0.5,b)}{\longmapsto} (\ell_{priv}, 1.6, 2) \overset{(0.7,b)}{\longmapsto} (\ell_f, 2.3, 2)$ of $\mathcal{A}$ in Fig. 3a, its counterpart in the TA in Fig. 3c is the run $(\ell_0, 0) \overset{(0.8,\varepsilon)}{\longmapsto} (\ell_{priv}, 0.8) \overset{(0.3,inc_1)}{\longmapsto} (\ell_{priv}, 1.1) \overset{(0.5,inc_1)}{\longmapsto} (\ell_{priv}, 1.6) \overset{(0.7,\varepsilon)}{\longmapsto} (\ell_f, 2.3)$. This run gives in $\mathcal{RA}({\mathcal{A}'}_{priv})$ the untimed word $inc_1 inc_1$, correctly encoding the final value of η_1. The public run $\rho_2 = (\ell_0, 0, 0) \overset{(2.3,c)}{\longmapsto} (\ell_f, 2.3, 2)$ also gives $inc_1 inc_1$.

Verifying EN-opacity amounts to comparing $\mathcal{L}(\mathcal{RA}({\mathcal{A}'}_{priv}))$ and $\mathcal{L}(\mathcal{RA}({\mathcal{A}'}_{pub}))$, by only focusing on the *number* of symbols inc_i in each word, disregarding their *order*. To do so, we use the Parikh image [Par66] of these languages (which can be represented using a semilinear set).

Verifying $\exists$-EN-opacity amounts to checking the emptiness of the intersection of these Parikh images. Checking intersection of the Parikh image of two regular languages is polynomial [KT10]. As the region automaton is exponential in the size of ${\mathcal{A}'}_{pub}$ and ${\mathcal{A}'}_{priv}$, themselves linear in $\mathcal{A}$, this gives a 2EXPSPACE procedure, since the subset construction of the region automata (required by the determinisation) adds another exponential.

Verifying weak-EN-opacity amounts to checking the inclusion of the Parikh images of $\mathcal{L}(\mathcal{RA}({\mathcal{A}'}_{priv}))$ and $\mathcal{L}(\mathcal{RA}({\mathcal{A}'}_{pub}))$. This is co-NP-complete [KT10], again yielding a 2EXPSPACE procedure. A similar reasoning holds for full-EN-opacity.

Finally, the case of discrete positive *guarded* METAs follows from Proposition 1, adding another exponential. □

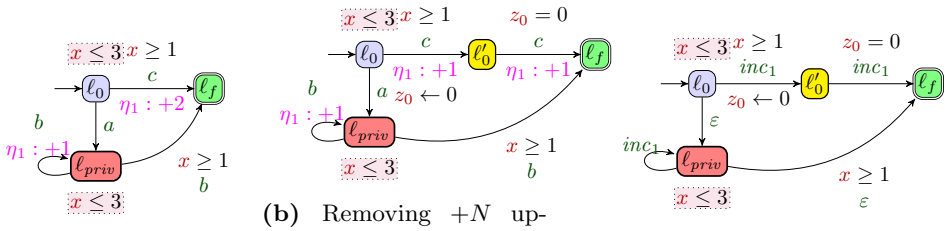
Example 5. Let $\mathcal{A}$ be the discrete META given in Fig. 3c. After transforming the updates into inc_i actions and constructing ${\mathcal{A}'}_{priv}$ and ${\mathcal{A}'}_{pub}$, we construct their region automata, see Figs. 4a and 4b (we omit some states and transitions due to space constraints). As the maximum constant for x is 3, we have 8 clock regions: $r_{\{0\}}, r_{(0,1)}, \ldots, r_{\{3\}}, r_{(3,\infty)}$, where r_i denotes the clock region such that $x \in i$. We can then test the emptiness of the intersection of the Parikh images of these two NFAs. In our example, we can directly see that the Parikh image of $\mathcal{L}(\mathcal{RA}({\mathcal{A}'}_{pub}))$ is $\{2\}$ while that of $\mathcal{L}(\mathcal{RA}({\mathcal{A}'}_{priv}))$ is $\mathbb{N}$; this corresponds indeed to $EVisit^{pub}(\mathcal{A})$ and $EVisit^{priv}(\mathcal{A})$ respectively. Therefore, $\mathcal{A}$ is $\exists$-EN-opaque. However, it is neither weakly nor fully-EN-opaque, as we do not have inclusion.

5.2 ET-EN-opacity in discrete positive (guarded) METAs

Theorem 2. *For each $\delta \in \{\exists, weak, full\}$, δ-ET-EN-opacity can be decided in* 2EXPSPACE *for discrete positive METAs, and* 3EXPSPACE *for discrete positive guarded METAs.*

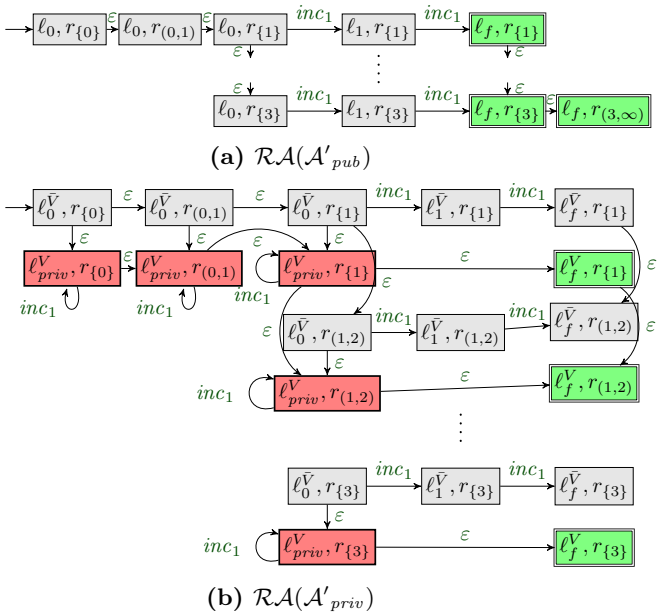
Proof. Given $\mathcal{A}$, we split increments of the form "$+N$" with $N > 1$ into N consecutive $+1$ increments in 0-time following the construction of $\mathcal{A}'$ in the proof of Theorem 1. Then, we modify $\mathcal{A}'$:

1. we add a global "ticking clock" z_t reset every time unit via action t on any location, and a global invariant $z_t \leq 1$ (not depicted in our figures);



(a) A META $\mathcal{A}$ **(b)** Removing $+N$ updates **(c)** Adding energy ticks

Figure 3: Example and transformations in the proof of Theorem 1



(a) $\mathcal{RA}(\mathcal{A}'_{pub})$

(b) $\mathcal{RA}(\mathcal{A}'_{priv})$

Figure 4: Simplified region graphs of $\mathcal{A}_{pub}$ and $\mathcal{A}_{priv}$ of Fig. 3c

2. we add to any guard the additional clock constraint $0 < z_t \le 1$, enforcing that action t occurs after the updates performed in 0-time on the same time unit;[2]

3. for each final location ℓ_f (made urgent), we add a new location ℓ'_f (the only ones to be final) reachable from ℓ_f via action t if ℓ_f is reachable on an integer time (which can be tested thanks to z_t), and via the fresh action $t_{>0}$ otherwise.

The last step of the construction is used to differentiate when the execution time is an integer, or when it is a non-integer (recall from [AAL24] that, if a location is reachable for a non-integer time, then it is for any other time in the open interval of length 1 with integer bounds to which this time belongs). We exemplify our construction on the discrete positive META of Fig. 3a in Fig. 5.

This transformation, say $\mathcal{A}''$, is linear in the size of $\mathcal{A}'$. We then build $\mathcal{A}''_{pub}$ and $\mathcal{A}''_{priv}$. The region automata

[2]For sake of readability, we do not consider in our transformation the case of updates at time 0, made impossible by the guard $0 < z_t$; a simple fix is to start in a copy of the automaton with a global invariant $z_t = 0$ to allow for updates at time 0 and, as soon as time elapses, moving to the "normal" automaton with the guards $0 < z_t \le 1$.

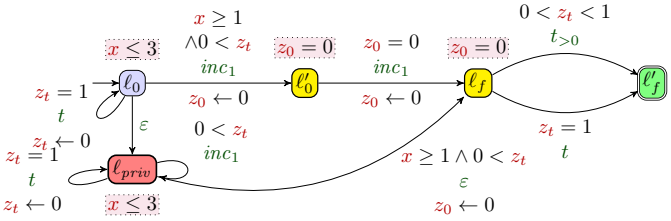


Figure 5: Exemplifying the construction in the proof of Theorem 2

$\mathcal{RA}(\mathcal{A}''_{pub})$ and $\mathcal{RA}(\mathcal{A}''_{priv})$ over $\Sigma = \{inc_i \mid 1 \le i \le |\mathcal{E}|\} \cup \{t, t_{>0}\}$ now contain not only the increments of energy thanks to the inc_i actions (as in the proof of Theorem 1), but also the discrete time elapsing thanks to the t actions; finally, the occurrence of one (resp. the absence of) $t_{>0}$ denotes a non-integer (resp. integer) execution time. Verifying $\exists$, weak and full opacity using Parikh image intersection and inclusion concludes the proof. □

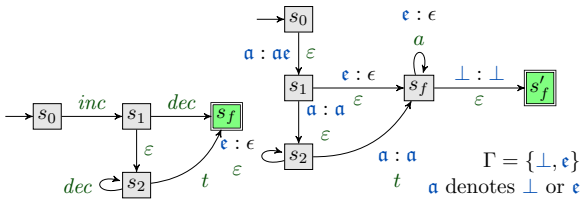
Example 6. The private run ρ_1 in the proof of Theorem 1 gives $tinc_1inc_1tt_{>0}$, while the public run ρ_2 gives $ttinc_1inc_1t_{>0}$. Even though these words differ, their Parikh image is $\{(2,2,1)\}$ (denoting 2, 2, 1 occurrences of inc_1, t, $t_{>0}$ respectively), and therefore the META in Fig. 3a is $\exists$-ET-EN-opaque.

5.3 (ET-)EN-opacity in discrete guarded ETAs

We now consider discrete guarded ETAs (with a single energy variable) with both increment and decrement. Recall that two energy variables make our problems undecidable (Proposition 2).

Theorem 3. *For each $\delta \in \{\exists, weak, full\}$ and $\sigma \in \{EN, ET\text{-}EN\}$, δ-σ-opacity is decidable for discrete ETAs.*

Proof. We start from the construction $\mathcal{A}''$ in the proof of Theorem 2, and extend it to decrements, i.e., using *dec* instead of *inc* when appropriate. As before, we then construct the NFAs $\mathcal{L}(\mathcal{RA}(\mathcal{A}''_{priv}))$ and $\mathcal{L}(\mathcal{RA}(\mathcal{A}''_{pub}))$. However, adding *dec* symbols to the construction in the proof of Theorem 2 may allow some impossible runs, typically causing the energy to drop below 0. Therefore, we turn each of these two NFAs into a pushdown automaton (PDA) (see Definition 6 recalled in Section B.1), as follows:



(a) An NFA **(b)** Transformed PDA

Figure 6: Construction in the proof of Theorem 3

1. We define a stack Γ encoding the value of the energy, over a single symbol $\mathfrak{e}$ (in addition to the empty stack symbol $\perp$);
2. We turn each transition of these NFAs with action *inc* (resp. *dec*) into a silent transition pushing a symbol $\mathfrak{e}$ to (resp. popping a symbol from) the stack;
3. For each accepting region s_f, we add a self-loop over s_f via a fresh action a popping an element $\mathfrak{e}$ of the stack; we add a new state s'_f connected from s_f via a silent transition testing for emptiness (i.e., the popped symbol is $\perp$);
4. we make all states s_f non-accepting, and all s'_f accepting.

An example of this transformation applied to a fictitious NFA is shown in Fig. 6. The languages of these PDAs are made of words of the form $\{t, a, t_{>0}\}^*$, where the number of "a" (resp. of "t") encodes the final energy (resp. the duration) of the run, and $t_{>0}$ encodes the non-integerness of the duration. Now, we can compute the intersection (resp. inclusion) of the Parikh images of the languages of these NPDAs, which is co-NP-complete (resp. Π_2^P-complete) [KT10], thus solving $\exists$-ET-EN opacity (resp. weak- and full ET-EN-opacity). Solving EN-opacity follows exactly the same reasoning, by making all $t, t_{>0}$ actions silent. □

For *guarded* ETAs, we cannot reuse the reasoning removing energy guards from Proposition 1, as the energy can also decrease. However, we can combine this mechanism with the stack, leading to the following result (proof is in Section E.2).

Theorem 4. *For each $\delta \in \{\exists, weak, full\}$ and $\sigma \in \{EN, ET\text{-}EN\}$, δ-σ-opacity is decidable for discrete guarded ETAs.*

6 Opacity with energy observation every time unit

We now consider the case when the attacker can observe the current energy level every time unit. The model remains over dense time, but the attacker has only a discrete observation power.

6.1 Observing the energy every time unit in discrete positive guarded ETAs

We first consider the setting when the attacker can observe the exact energy valuation every time unit.

Given a run $\rho = (\ell_0, w_0, v_0), (d_0, e_0), (\ell_1, w_1, v_1), \ldots, (\ell_n, w_n, v_n)$ of a discrete positive guarded META, we define the *energy level at time t* as follows:

$$EL(\rho, t) = \begin{cases} \vec{0}_{\mathcal{E}} & \text{if } t < d_0 \\ v_k \text{ where } k = \max_i \left((\sum_{j=0}^{i} d_j) \leq t \right) & \text{otherwise.} \end{cases}$$

Then, the discrete energy observation of ρ is:

$$DEO(\rho) = EL(\rho, 1), \ldots, EL(\rho, k) \text{ with } k = \lceil \sum_{j=0}^{n} d_i \rceil.$$

A guarded META $\mathcal{A}$ is *weakly discrete-energy opaque (weakly DE-opaque)* whenever, for each run $\rho \in Visit^{priv}(\mathcal{A})$, there exists a run $\rho' \in Visit^{pub}(\mathcal{A})$ such that $DEO(\rho) = DEO(\rho')$. Full and existential counterparts are defined similarly. We define the discrete-energy opacity problem as follows (for each $\delta \in \{\exists, \text{weak}, \text{full}\}$):

δ-DE-opacity problem:
INPUT: A guarded META $\mathcal{A}$
PROBLEM: Is $\mathcal{A}$ δ-DE-opaque?

Example 7. Consider again the discrete positive META in Fig. 2a. Let ρ be the private run $(\ell_0, 0, 0) \overset{(1.8,a)}{\longmapsto} (\ell_{priv}, 0, 1) \overset{(0.4,b)}{\longmapsto} (\ell_{priv}, 0, 3) \overset{(0.8,b)}{\longmapsto} (\ell_{priv}, 0, 5) \overset{(0.7,b)}{\longmapsto} (\ell_f, 0.7, 5)$. We have $DEO(\rho) = 0, 1, 5, 5$.

We first rule out non-necessarily positive guarded METAs; the proof is a simple adaptation from Proposition 2.

Proposition 3. *For each $\delta \in \{\exists, weak, full\}$, δ-DE-opacity for discrete guarded METAs with 2 energies and 1 clock is undecidable.*

The following positive result only holds for positive ETAs, i.e., with a single energy.

Theorem 5. *For each $\delta \in \{\exists, weak, full\}$, δ-DE-opacity can be decided in* EXPSPACE *for discrete positive (guarded) ETAs.*

The following result extends DE-opacity to METAs, but only for the $\exists$ variant. We use a novel construction of "Parikh by block" automaton.

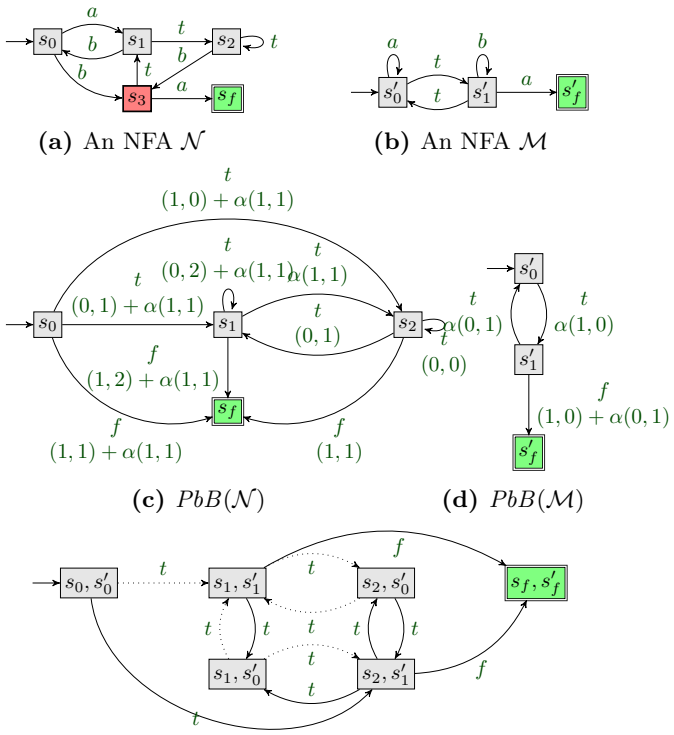
Theorem 6. *$\exists$-DE-opacity is decidable for discrete positive (guarded) METAs.*

Proof (sketch). We first build $\mathcal{A}'$ similarly to the proof of Theorem 2, and then construct the region automata

of $\mathcal{A}'_{priv}$ and $\mathcal{A}'_{pub}$, where $\mathcal{A}'$ embeds the inc_i actions (encoding discrete increments for the various energy variables), and time ticks (actions t, and at most one final action f (which was denoted by $t_{>0}$ in Theorem 2)). Then, verifying $\exists$-DE-opacity amounts to checking whether there exist a run in $\mathcal{A}'_{priv}$ and a run of $\mathcal{A}'_{pub}$ with the same execution time (same number of t), and the same energy valuations at each tick, i.e., the same number of increments inc_i in between two consecutive t actions, regardless of their order—which is indeed not visible to the attacker.

To this end, we define and use a novel construction, which we call "Parikh by block" (PbB) automaton: the idea is to count the occurrences of different actions (here the inc_i), between two consecutive ticks (actions t or f)—hence the notion of "block". Counting such numbers of actions between two ticks will be represented using a Parikh image, i.e., a semilinear set [Par66]. The resulting structure is a finite automaton over actions t and f, where each transition is additionally labelled by a Parikh image over the inc_i, encoding the number of inc_i since the last tick. We then perform the synchronised product of $PbB(\mathcal{A}'_{priv})$ and $PbB(\mathcal{A}'_{pub})$ over $\{t, f\}$; each transition in the resulting product (which is a finite automaton) is labelled by *two* Parikh images, one coming from $PbB(\mathcal{A}'_{priv})$ and one from $PbB(\mathcal{A}'_{pub})$. Finally, we check the existence of an accepting path in this automaton such that, for each transition, the intersection of the two Parikh images is non-empty. All these operations on Parikh images can be computed using the (decidable) Presburger arithmetics [BHK17]. This indeed denotes the existence of two paths, one private and one public, with the same execution time, and with the same energy level at each time unit (correctly encoded by the number of inc_i between consecutive t actions)—which decides $\exists$-DE-opacity. See additional details in Section F.3. □

Example 8. Let us illustrate our Parikh by block construction. Consider the NFAs $\mathcal{N}, \mathcal{M}$ in Figs. 7a and 7b (which could be the region automata of $\mathcal{A}'_{priv}$ and $\mathcal{A}'_{pub}$ for some discrete positive META $\mathcal{A}$, with a and b denoting for example inc_1 and inc_2). Considering a Parikh by block construction over $\{t, f\}$, we give $PbB(\mathcal{N})$ and $PbB(\mathcal{M})$ in Figs. 7c and 7d respectively, where the semilinear set $(0,2)+\alpha(1,1)$ denotes "exactly 2 bs plus an arbitrarily (yet identical) number of as and bs". Their synchronised product is given in Fig. 7e, where labels (Parikh images) are omitted; a dashed line denotes an empty intersection of the two associated Parikh images, while a plain line denotes a non-empty intersection. For example, the edge from (s_0, s'_0) to (s_1, s'_1) is labelled with "$(0,1)+\alpha(1,1)$" and "$\alpha(1,0)$"; the intersection of these two semilinear sets is empty. In contrast, the edge from (s_0, s'_0) to (s_2, s'_1) is labelled with "$(1,0)+\alpha(1,1)$" and "$\alpha(1,0)$"; the intersection is clearly non-empty, for example $(1,0)$ is a solution (denoting one a, and no b). Finally, we can exhibit a path over edges with a non-empty intersection—denoting that the (fictitious) original



(a) An NFA $\mathcal{N}$ **(b)** An NFA $\mathcal{M}$

(c) $PbB(\mathcal{N})$ **(d)** $PbB(\mathcal{M})$

(e) Synchronised product of $PbB(\mathcal{N})$ and $PbB(\mathcal{M})$

Figure 7: Illustrating the Parikh by block construction

META $\mathcal{A}$ is $\exists$-DE-opaque.

6.2 Observing the buffered energy changes every time unit in discrete METAs

We now consider an attacker able to see all the evolution of energy and their order, but with no knowledge of the exact timestamps between two consecutive time units. That is, the attacker can make a difference between an increase of 2 energy units followed by a decrease of 3 (within the same time unit) compared to a decrease of 3 followed by an increase of 2—yet, they will not be able to know the exact timestamp of these events. This can be seen as the natural setting in which a buffer storing all energy changes is read every time unit by the attacker; this is also reminiscent of the recent setting of the weakened attacker able to observe absolute time only at integer times ("intruder with discrete-time precision") in [DQY25].

Let ϵ denote the empty sequence. Given $\tau \in \mathbb{N}\setminus\{0\}$, let $bEL(\rho, \tau)$ be the sequence of energy valuations defined as follows: when $\tau > 1$, this is the sequence of energy valuations obtained from the subrun of ρ by keeping only timestamps in $(\tau-1, \tau]$ which modify at least one energy variable; the case $\tau = 1$ is obtained by keeping the timestamps in $[0, 1]$. (See Definition 11 in Section C.2 for a formal definition.) Then, the buffered discrete energy observation of ρ is:

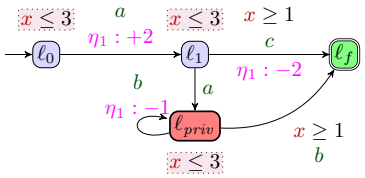


Figure 8: A discrete ETA

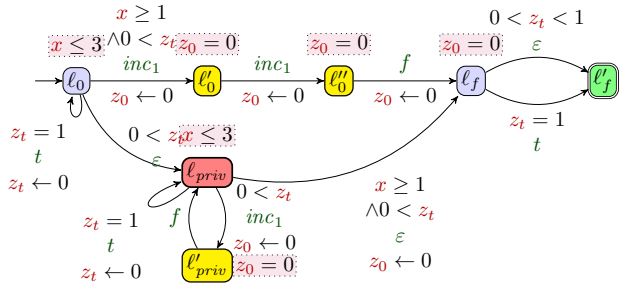


Figure 9: Exemplifying the construction in the proof of Theorem 7

$$bDEO(\rho) = bEL(\rho,1),\dots,bEL(\rho,k) \text{ with } k = \lceil \sum_{j=0}^{n-1} d_i \rceil.$$

A discrete guarded META $\mathcal{A}$ is *weakly buffered-discrete-energy opaque (weakly bDE-opaque)* when, for each run $\rho \in Visit^{priv}(\mathcal{A})$, there exists a run $\rho' \in Visit^{pub}(\mathcal{A})$ such that $bDEO(\rho) = bDEO(\rho')$. Full and existential counterparts are defined similarly. We define the buffered-discrete-energy opacity problem as expected.

Example 9. Consider the following runs of the discrete ETA $\mathcal{A}$ in Fig. 8. Let ρ_1 be the private run $(\ell_0,0,0) \overset{(1.5,a)}{\mapsto} (\ell_1,1.5,2) \overset{(1,a)}{\mapsto} (\ell_{priv},2.5,2) \overset{(0.1,b)}{\mapsto} (\ell_{priv},2.6,1) \overset{(0.1,b)}{\mapsto} (\ell_{priv},2.7,0) \overset{(0.1,b)}{\mapsto} (\ell_f,2.8,0)$. Let ρ_2 be the public run $(\ell_0,0,0) \overset{(1.5,a)}{\mapsto} (\ell_1,1.5,2) \overset{(1.3,c)}{\mapsto} (\ell_f,2.8,0)$. We have $DEO(\rho_1) = DEO(\rho_2) = 0,2,0$, and therefore $\mathcal{A}$ is $\exists$-DE-opaque. Now, let us address bDE-opacity: we have $bDEO(\rho_1) = \epsilon,(2),(1,0)$ while $bDEO(\rho_2) = \epsilon,(2),(0)$; it can actually be shown that $\mathcal{A}$ is *not* $\exists$-bDE-opaque.

Theorem 7. *For each $\delta \in \{\exists, weak, full\}$, δ-bDE-opacity can be decided in* EXPSPACE *for discrete positive METAs, and in* 2EXPSPACE *for discrete positive guarded METAs.*

Proof. See Section F.4. □

Our last result is concerned with non-necessarily positive discrete ETAs, and makes use of operations over both regular and context-free languages.

Theorem 8. *For each $\delta \in \{\exists, weak, full\}$, δ-bDE-opacity can be decided in* 2EXPSPACE *for discrete ETAs.*

Note that the case of discrete (guarded) METAs is not covered by Theorems 7 and 8 and remains open.

7 Opacity for integer-switching METAs

After considering discrete METAs in the former sections, we finally consider here a (first) case in which energy variables can evolve continuously. A guarded META is said *integer-switching (IS)* whenever all changes of energy rates are made on an (absolute) integer time. Formally, there exists no run $(\ell_0,w_0,v_0) \overset{(d_0,e_0)}{\mapsto} (\ell_1,w_1,v_1) \overset{(d_1,e_1)}{\mapsto} \cdots \overset{(d_{m-1},e_{m-1})}{\mapsto} (\ell_m,w_m,v_m)$ such that $\exists 0 \le i \le m-1, \exists \eta \in \mathcal{E}$ such that $\phi(\ell_i)(\eta) \neq \phi(\ell_{i+1})(\eta)$ and $\sum_{j=0}^{i} d_j \notin \mathbb{N}$. Note that both discrete energy updates and location changes (not modifying the rates) are allowed to occur at non-integer times, as well as any clock reset, keeping this model entirely dense-time. While this condition is essentially semantic, it can easily be checked on unguarded METAs by checking the region graph. For example, the META in Fig. 10a is IS. In addition, a guarded META is *integer execution-time (iET)* whenever any final location is reached only on integer absolute times. For example, the META in Fig. 10a is *not* iET, due to the guard $x \ge 1$ to ℓ_f.

We now introduce a transformation from IS-METAs to METAs that will allow us to transfer some decidability results.

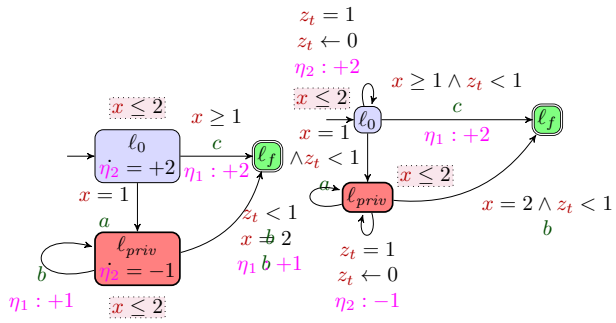
1. we add a global "ticking clock" z_t reset every time unit via action t on any location, and a global invariant $z_t \le 1$ (not depicted in our figures), as in the proof of Theorem 2;
2. we add to any guard the additional clock constraint $0 \le z_t < 1$, enforcing that outgoing guards are evaluated *after* the discrete energy updates added by our construction;
3. every time unit (when $z_t = 1$), we turn the location rate into a discrete update of all energy variables.

This transformation is linear in $\mathcal{A}$. See Fig. 10 for an example of our construction. Our transformation clearly preserves the final energies for iET-IS-METAs, as well as the energy every time unit. This gives the following results.

Proposition 4. *1. For each $\delta \in \{\exists, weak, full\}$ and $\sigma \in \{EN, ET\text{-}EN\}$, δ-σ-opacity can be decided in* 2EXPSPACE *for positive iET-IS-METAs.*

2. For each $\delta \in \{\exists, weak, full\}$ and $\sigma \in \{EN, ET\text{-}EN\}$, δ-σ-opacity is decidable for iET-IS-ETAs.

3. For each $\delta \in \{\exists, weak, full\}$, δ-DE-opacity can be decided in EXPSPACE *for positive iET-IS-ETAs; and $\exists$-DE-opacity can be decided in* EXPSPACE *for positive iET-IS-METAs.*



(a) An IS-META $\mathcal{A}$ **(b)** Transformed META

Figure 10: Turning an IS-META into a META

Proof. From Theorems 1 to 3, 5 and 6. □

Note that our transformation does not correctly encode the energy guards, hence guarded IS-METAs are left out. In addition, our transformation does not properly encode the energy elapse beyond the last integer time (which can also make the energy become negative in between two time units), hence the iET assumption. Lifting these assumptions is the subject of ongoing works.

8 Conclusion

In this paper, we introduced the formalism of guarded METAs to express opacity properties over cyber-physical systems. While the general class is unsurprisingly undecidable, we exhibited a number of decidability results for discrete subclasses (i.e., continuous models, where the energy can be incremented or decremented at integer or rational times); in addition, we exhibited first decidability results for models including continuous energy elapsing over multiple energy variables. We summarize results from Sections 5 and 6 in Tables 1 and 2.

Beyond the open cases in Tables 1 and 2 and the lower bounds of the complexities, future works include observation of the sole execution time over guarded METAs, i.e., where the energy, while unobservable, constrains the system. In addition, we will be interested in verification under energy uncertainty, demanding parametric verification techniques [BBF+21].

Acknowledgments

We thank Engel Lefaucheux and Sarah Dépernet for preliminary discussions regarding timed opacity with energy. This work is partially supported by ANR BisoUS (ANR-22-CE48-0012) and by Agence de l'Innovation de Défense (AID) via Centre Interdisciplinaire Mers et Océan (CIMO) project 2025 CHRoMM.

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), 2023.

[AAL24] Étienne André, Johan Arcile, and Engel Lefaucheux. Execution-time opacity problems in one-clock parametric timed automata. In *FSTTCS*, volume 323 of *LIPIcs*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024.

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

[ACHH93] Rajeev Alur, Costas Courcoubetis, Thomas A. Henzinger, and Pei-Hsin Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In *Hybrid Systems 1992*, volume 736 of *Lecture Notes in Computer Science*. Springer, 1993.

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

[ADKLL24] Étienne André, Marie Duflot-Krémer, Laetitia Laversa, and Engel Lefaucheux. Execution-time opacity control for timed automata. In *SEFM*, volume 15280 of *Lecture Notes in Computer Science*. Springer, 2024.

[ADL24] Étienne André, Sarah Dépernet, and Engel Lefaucheux. The bright side of timed opacity. In *ICFEM*, volume 15394 of *Lecture Notes in Computer Science*. Springer, 2024.

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

[AGW+24] Jie An, Qiang Gao, Lingtai Wang, Naijun Zhan, and Ichiro Hasuo. The opacity of timed automata. In *FM*, volume 14933 of *Lecture Notes in Computer Science*. Springer, 2024.

[AK20] Étienne André and Aleksander Kryukov. Parametric non-interference in timed automata. In *ICECCS*, 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 *Elec-*

For each $\delta \in \{\exists, \text{weak}, \text{full}\}$	δ-EN-opacity	δ-ET-EN-opacity
Discrete positive METAs	2EXPSPACE (Theorem 1)	2EXPSPACE (Theorem 2)
Discrete positive guarded METAs	3EXPSPACE (Theorem 1)	3EXPSPACE (Theorem 2)
Discrete (guarded) ETAs	decidable (Theorems 3 and 4)	
Discrete guarded METAs	undecidable (Proposition 2)	

Table 1: Decidability of (ET-)EN-opacity in discrete METAs

	DE-opacity		bDE-opacity	
	$\exists$	Weak / full	$\exists$	Weak / full
Discrete positive (guarded) ETAs	EXPSPACE (Theorem 5)		2EXPSPACE* (Theorem 7)	
Discrete positive (guarded) METAs	decidable (Theorem 6)		2EXPSPACE* (Theorem 7)	
Discrete ETAs			2EXPSPACE (Theorem 8)	
Discrete METAs				
Discrete guarded METAs	undecidable (Proposition 3)			

Table 2: Decidability of (b)DE-opacity in discrete METAs (*EXPSPACE without energy guards)

tronic Proceedings in Theoretical Computer Science, 2023. Invited paper.

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

[ALR22] Étienne André, Didier Lime, and Olivier H. Roux. Reachability and liveness in parametric timed automata. *Logical Methods in Computer Science*, 18(1), 2022.

[BBF+21] Giovanni Bacci, Patricia Bouyer, Uli Fahrenberg, Kim Guldstrand Larsen, Nicolas Markey, and Pierre-Alain Reynier. Optimal and robust controller synthesis using energy timed automata with uncertainty. *Formal Aspects of Computing*, 33(1), 2021.

[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), 2015.

[BDFP04] Patricia Bouyer, Catherine Dufourd, Emmanuel Fleury, and Antoine Petit. Updatable timed automata. *Theoretical Computer Science*, 321(2-3), 2004.

[BFH+01] Gerd Behrmann, Ansgar Fehnker, Thomas Hune, Kim Guldstrand Larsen, Paul Pettersson, Judi Romijn, and Frits W. Vaandrager. Minimum-cost reachability for priced timed automata. In *HSCC*, volume 2034 of *Lecture Notes in Computer Science*. Springer, 2001.

[BFL+08] Patricia Bouyer, Ulrich Fahrenberg, Kim Guldstrand Larsen, Nicolas Markey, and Jiří Srba. Infinite runs in weighted timed automata with energy constraints. In *FORMATS*, volume 5215 of *Lecture Notes in Computer Science*. Springer, 2008.

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

[BHK17] Simon Beier, Markus Holzer, and Martin Kutrib. On the descriptional complexity of operations on semilinear sets. In *AFL*, volume 252 of *EPTCS*, 2017.

[BKMR08] Jeremy W. Bryans, Maciej Koutny, Laurent Mazaré, and Peter Y. A. Ryan. Opacity generalised to transition systems. *International Journal of Information Security*, 7(6), 2008.

[BLR05] Gerd Behrmann, Kim Guldstrand Larsen, and Jacob Illum Rasmussen. Optimal scheduling using priced timed automata. *SIGMETRICS Performance Evaluation Review*, 32(4), 2005.

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

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

[CHL24] Pieter J. L. Cuijpers, Jonas Hansen, and Kim Guldstrand Larsen. Safe and infinite resource scheduling using energy timed automata. In *TASE*, volume 14777 of *Lecture Notes in Computer Science*. Springer, 2024.

[CM14] Dmitry Chistikov and Rupak Majumdar. Unary pushdown automata and straight-line programs. In *ICALP, Part II*, volume 8573 of *Lecture Notes in Computer Science*. Springer, 2014.

[DQY25] Weilin Deng, Daowen Qiu, and Jingkai Yang. New insights into the decidability of opacity in timed automata. Technical Report abs/2504.00625, arXiv, 2025.

[GMR07] Guillaume Gardey, John Mullins, and Olivier H. Roux. Non-interference control synthesis for security timed automata. *Electronic Notes in Theoretical Computer Science*, 180(1), 2007.

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

[KMT17] Markus Krötzsch, Tomás Masopust, and Michaël Thomazo. Complexity of universality and related problems for partially ordered NFAs. *Information and Computation*, 255, 2017.

[KT10] Eryk Kopczyński and Anthony Widjaja To. Parikh images of grammars: Complexity and applications. In *LiCS*. IEEE Computer Society, 2010.

[LLH18] Stéphane Lafortune, Feng Lin, and Christoforos N. Hadjicostis. On the history of diagnosability and opacity in discrete event systems. *Annual Reviews in Control*, 45, 2018.

[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), 2022.

[Maz04] Laurent Mazaré. Using unification for opacity properties. In *WITS*, 2004.

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

[Par66] Rohit Parikh. On context-free languages. *Journal of the ACM*, 13(4), 1966.

[SLR23] Anthony Spriet, Didier Lime, and Olivier H. Roux. Timed non-interference under partial observability and bounded memory. In *FORMATS*, volume 14138 of *Lecture Notes in Computer Science*. Springer, 2023.

[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*. 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), 2018.

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

A Mathematical concepts

Definition 5 (Ordered partition of $\mathbb{R}_{\geq 0}$)**.** We call *ordered integer partition of* $\mathbb{R}_{\geq 0}$ (for short *ordered partition of* $\mathbb{R}_{\geq 0}$) a finite sequence of disjoint intervals $\nu_1, \ldots, \nu_n$ such that:

1. $\bigcup_{j=1}^{n} \nu_j = \mathbb{R}_{\geq 0}$,
2. all boundaries except the right boundary of ν_n belong to $\mathbb{N}$,
3. each ν_j is non empty, and
4. for all $j \in \{1, \ldots, n-1\}$, the right boundary of ν_j is the left boundary of ν_{j+1}.

Example 10. The sequence $[0,2], (2,5), [5,10), [10,+\infty)$ is an ordered partition of $\mathbb{R}_{\geq 0}$.

B Common automata concepts

B.1 Formal definition of pushdown automata

Definition 6. A *pushdown automaton (PDA)* is a tuple $(\mathfrak{Q}, \Sigma, \Gamma, \mathfrak{q}_0, \bot, \mathfrak{Q}_F, \mathfrak{E})$ where:

1. $\mathfrak{Q}$ is a finite set of states,
2. Σ is a finite set of actions ("input alphabet"),
3. Γ is a finite set of stack symbols ("stack alphabet"),
4. $\mathfrak{q}_0 \in \mathfrak{Q}$ is the initial state,
5. $\bot \in \Gamma$ is the initial stack symbol,
6. $\mathfrak{Q}_F \subseteq \mathfrak{Q}$ is the set of accepting states,
7. $\mathfrak{E}$ is a finite set of edges $\mathfrak{e} = (\mathfrak{q}, a, \mathfrak{a}, \mathfrak{q}', \mathfrak{a}*)$ where $\mathfrak{q}, \mathfrak{q}' \in \mathfrak{Q}$ are the source and target states, $a \in \Sigma \cup \{\varepsilon\}$ is the transition symbol, and $\mathfrak{a} \in \Gamma$, $\mathfrak{a}* \in (\Gamma)^*$ are the symbols respectively popped from and pushed to the stack.

B.2 Region automaton

We recall that the region automaton of timed automata is obtained by quotienting the set of clock valuations out by an equivalence relation $\simeq$ recalled below. As an abuse of notation, we denote a TA $\mathcal{A} = (\Sigma, L, \ell_0, L_{priv}, L_f, \gamma, \mathcal{X}, I, E)$, since the set of energy variables is empty, and all rates are 1.

Given a TA $\mathcal{A}$ and its set of clocks $\mathcal{X}$, we define $M : \mathcal{X} \to \mathbb{N}$ the map that associates to a clock x the greatest value to which the interpretations of x are compared within the guards and invariants; if x appears in no constraint, we set $M(x) = 0$.

Given $\alpha \in \mathbb{R}$, we write $\lfloor \alpha \rfloor$ and $\mathrm{frac}(\alpha)$ for the integral and fractional parts of α, respectively.

Definition 7 (Equivalence relation $\simeq$ for valuations [AD94])**.** Let w, w' be two clock valuations. We say that w and w' are *equivalent*, denoted by $w \simeq w'$ when, for each $x \in \mathcal{X}$, either $w(x) > M(x)$ and $w'(x) > M(x)$ or the three following conditions hold:

1. $\lfloor w(x) \rfloor = \lfloor w'(x) \rfloor$;
2. $\mathrm{frac}(w(x)) = 0$ if and only if $\mathrm{frac}(w'(x)) = 0$; and
3. for each $y \in \mathcal{X}$, $\mathrm{frac}(w(x)) \leq \mathrm{frac}(w(y))$ if and only if $\mathrm{frac}(w'(x)) \leq \mathrm{frac}(w'(y))$.

The equivalence relation is extended to the concrete states of $\mathcal{A}$: let $s = (\ell, w)$ and $s' = (\ell', w')$ be two concrete states in $\mathcal{A}$, then $s \simeq s'$ if and only if $\ell = \ell'$ and $w \simeq w'$.

The equivalence class of a valuation w is denoted $[w]$ and is called a *clock region*, and the equivalence class of a concrete state $s = (\ell, w)$ is denoted $[s]$ and called a *region* of $\mathcal{A}$. In other words, a region is a pair made of a location and of a clock region. Clock regions are denoted by the enumeration of the constraints defining the equivalence class. Thus, values of a clock x that go beyond $M(x)$ are merged and described in the regions by the inequality "$x > M(x)$".

The set of regions of $\mathcal{A}$ is denoted by $\mathcal{R}_{\mathcal{A}}$. These regions are of finite number: this allows us to construct a finite "untimed" regular automaton, the *region automaton* $\mathcal{RA}(\mathcal{A})$. Locations of $\mathcal{RA}(\mathcal{A})$ are regions of $\mathcal{A}$, and the transitions of $\mathcal{RA}(\mathcal{A})$ convey the reachable valuations associated with each concrete state in $\mathcal{A}$.

To formalize the construction, we need to transform discrete and time-elapsing transitions of $\mathcal{A}$ into transitions between the regions of $\mathcal{A}$. To do that, we define a "time-successor" relation that corresponds to time-elapsing transitions.

Definition 8 (Time-successor relation [ALM23])**.** Let $r = (\ell, [w]), r' = (\ell', [w']) \in \mathcal{R}_{\mathcal{A}}$. We say that r' is a time-successor of r when $r \neq r'$, $\ell = \ell'$ and for each concrete state (ℓ, w) in r, there exists $d \in \mathbb{R}_{\geq 0}$ such that $(\ell, w + d)$ is in r' and for all $d' < d$, $(\ell, w + d') \in r \cup r'$.

A region $r = (\ell, [w])$ is *unbounded* when, for all x in $\mathcal{X}$ and all $w' \in [w]$, $w'(x) > M(x)$.

Definition 9 (Region automaton [AD94])**.** Given a TA $\mathcal{A} = (\Sigma, L, \ell_0, L_{priv}, L_f, \gamma, \mathcal{X}, I, E)$, the region automaton is the tuple $\mathcal{RA}(\mathcal{A}) = (\Sigma_{\mathcal{R}}, \mathcal{R}, r_0, \mathcal{R}_f, E_{\mathcal{R}})$ where

1. $\Sigma_{\mathcal{R}} = \Sigma \cup \{\varepsilon\}$;
2. $\mathcal{R} = \mathcal{R}_{\mathcal{A}}$;
3. $r_0 = [s_0]$;
4. $\mathcal{R}_f$ is the set of regions whose first component is a final location $\in L_f$;
5. $E_{\mathcal{R}}$ is defined as follows:
   - *(discrete transitions)* For every $r = (\ell, [w])$ with $\ell \notin L_f$, $r' = (\ell', [w']) \in \mathcal{R}_{\mathcal{A}}$ and $a \in \Sigma \cup \{\varepsilon\}$: $(r, a, r') \in E_{\mathcal{R}}$ if $\exists w'' \in [w], \exists w''' \in [w'], \big((\ell, w''), e, (\ell', w''')\big) \in \to$,

with $e = (\ell, g, a, R, \ell') \in E$ for some guard g and some clock set R.

- *(delay transitions)* For every $r = (\ell, [w])$ with $\ell \notin L_f$, $r' \in \mathcal{R}_{\mathcal{A}}$: $(r, \varepsilon, r') \in E_{\mathcal{R}}$ if r' is a time-successor of r or if $r = r'$ is unbounded.

A *run* of $\mathcal{RA}(\mathcal{A})$ is an alternating sequence of regions of $\mathcal{RA}(\mathcal{A})$ and actions starting from the initial region r_0 and ending in a final region, of the form $r_0, a_0, r_1, a_1, \dots r_{n-1}, a_{n-1}, r_n$ for some $n \in \mathbb{N}$, with $r_n \in \mathcal{R}_f$ and for each $0 \leq i \leq n-1$, $r_i \notin \mathcal{R}_f$, and $(r_i, a_i, r_{i+1}) \in E_{\mathcal{R}}$.

$\mathcal{RA}(\mathcal{A})$ is a non-deterministic finite automaton (NFA); its language is defined as usual as the set of accepting words, i.e., as the sequence of actions (ε excluded) contained in a run. The (untimed) language of $\mathcal{RA}(\mathcal{A})$ is denoted by $\mathcal{L}(\mathcal{RA}(\mathcal{A}))$.

C Formal definitions of opacity

C.1 ET-EN-opacity

Definition 10 (ET-EN-opacity)**.** A guarded META $\mathcal{A}$ is *fully ET-EN-opaque* when, for each $d \in \mathbb{R}_{\geq 0}$ and $v \in \mathbb{R}_{\geq 0}^{|\mathcal{E}|}$, there exists a private run ρ such that $dur(\rho) = d$ and $FE(\rho) = v$ iff there exists a public run ρ' such that $dur(\rho') = d$ and $FE(\rho') = v$.

A guarded META $\mathcal{A}$ is *weakly ET-EN-opaque* when for each private run ρ, there exists a public run ρ' such that $dur(\rho) = dur(\rho')$ and $FE(\rho) = FE(\rho')$.

A guarded META $\mathcal{A}$ is $\exists$-*ET-EN-opaque* when there exist a private run ρ and a public run ρ' such that $dur(\rho) = dur(\rho')$ and $FE(\rho) = FE(\rho')$.

Remark 1. Note that verifying δ-ET-opacity and δ-EN-opacity is not a sufficient condition for verifying δ-ET-EN opacity. Indeed, a model can be δ-ET-opaque and δ-EN-opaque but not δ-ET-EN opaque. The guarded META in Fig. 11 is such that:

- $DVisit^{priv}(\mathcal{A}) = DVisit^{pub}(\mathcal{A}) = [0, 10]$
- $EVisit^{priv}(\mathcal{A}) = EVisit^{pub}(\mathcal{A}) = [0, 10]$

It is therefore both fully-ET-opaque and fully-EN-opaque. However, there exists a private run such that no public run has the same execution time and the same final energy (for example, execution time of 4 and final energy at 6). Therefore, it is neither fully nor weakly-ET-EN-opaque.

C.2 Formal definition of function *bEL*

We first need to embed the absolute time of each state of a run, and project onto the states, i.e., removing the transitions. Given a run $\rho = (\ell_0, w_0, v_0), (d_0, e_0), (\ell_1, w_1, v_1), \dots, (\ell_n, w_n, v_n)$, let $absT(\rho) = (\ell_0, w_0, v_0, t_0), (\ell_1, w_1, v_1, t_1), \dots, (\ell_n, w_n, v_n, t_n)$, where $t_i = \sum_{j=0}^{i-1} d_j$.

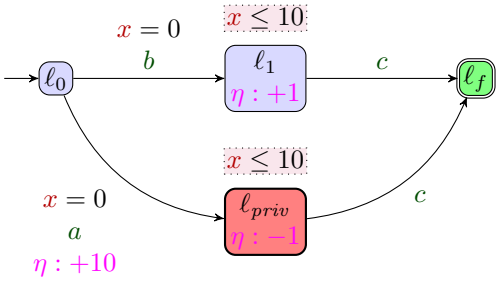


Figure 11: Guarded META ET-opaque and EN-opaque but not fully-ET-EN-opaque

We also define a function *destutter* that, given a non-empty sequence *seq* of quadruples made of a location, a clock valuation, an energy valuation and an absolute time (typically the result of $absT(\rho)$), returns the subsequence of *seq* where only the quadruples that modify at least one energy variable are kept. The formal definition is given in Fig. 12.

Then, given a non-empty sequence *seq* of quadruples made of a location, a clock valuation, an energy valuation and an absolute time, and given a time $\tau \in \mathbb{N} \setminus \{0\}$, we define the *subsequence in the interval ending in τ, projected onto the energy variables*, denoted by $subseqProj(seq, \tau)$, as in Fig. 13.

We can finally define function *bEL*.

Definition 11. Let ρ be a run, and let $\tau \in \mathbb{N} \setminus \{0\}$ be an absolute time. The sequence of energy valuations $bEL(\rho, \tau)$ is defined by

$$bEL(\rho, \tau) = subseqProj\Big(destutter\big(absT(\rho)\big), \tau\Big).$$

D Definitions and proofs of Section 4

D.1 Formal definitions of $\mathcal{A}_{pub}$ and $\mathcal{A}_{priv}$

Definition 12 ($\mathcal{A}_{pub}$)**.** Let $\mathcal{A} = (\Sigma, L, \ell_0, L_{priv}, L_f, \gamma, \mathcal{X}, \mathcal{E}, I, \phi, E)$ be a guarded META. Then, $\mathcal{A}_{pub}$ is defined as $(\Sigma, L, \ell_0, L_{priv}, L_f, \gamma, \mathcal{X}, \mathcal{E}, I, \phi, E')$, where $E' = E \setminus \{(\ell, g, a, R, U, \ell') \mid \ell \in \ell_{priv} \vee \ell' \in \ell_{priv}\}$.

Definition 13 ($\mathcal{A}_{priv}$)**.** Let $\mathcal{A} = (\Sigma, L, \ell_0, L_{priv}, L_f, \gamma, \mathcal{X}, \mathcal{E}, I, \phi, E)$ be a guarded META. Then, $\mathcal{A}_{priv}$ is defined as $(\Sigma, L', \ell_0', L_{priv}', L_f', \gamma', \mathcal{X}, \mathcal{E}, I', \phi', E')$, where

- $L' = \{\ell^{\bar{V}} \mid \ell \in L\} \cup \{\ell^{V} \mid \ell \in L\}$,
- $\ell_0' = \ell_0^{\bar{V}}$,
- $L_f' = \{\ell_f^{V} \mid \ell_f \in L_f\}$,
- $\forall \ell \in L, \gamma'(\ell^{V}) = \gamma'(\ell^{\bar{V}}) = \gamma(\ell)$,
- $\forall \ell \in L, I'(\ell^{V}) = I'(\ell^{\bar{V}}) = I(\ell)$,
- $\forall \ell \in L, \phi'(\ell^{V}) = \phi'(\ell^{\bar{V}}) = \phi(\ell)$,
- $E' = \big\{(\ell_1^{\bar{V}}, g, a, R, U, \ell_2^{\bar{V}}) \mid (\ell_1, g, a, R, U, \ell_2) \in E \wedge \ell_2 \notin L_{priv}\big\} \cup \big\{(\ell_1^{V}, g, a, R, U, \ell_2^{V}) \mid$

$$destutter(seq) = \begin{cases} seq & \text{if } seq = (\ell_0, w_0, v_0, t_0) \text{ or } seq = \epsilon \\ destutter\big((\ell_i, w_i, v_i, t_i), seq'\big) & \text{if } seq = (\ell_i, w_i, v_i, t_i), (\ell_j, w_j, v_j, t_j), seq' \text{ and } v_i = v_j \\ (\ell_i, w_i, v_i, t_i), destutter\big((\ell_j, w_j, v_j, t_j), seq'\big) & \text{if } seq = (\ell_i, w_i, v_i, t_i), (\ell_j, w_j, v_j, t_j), seq' \text{ and } v_i \neq v_j \end{cases}$$

Figure 12: Formal definition of *destutter*

$$subseqProj(seq, \tau) = \begin{cases} (v_0, \ldots, v_k) \text{ where } t_k \leq 1 < t_{k+1} & \text{if } \tau = 1 \\ (v_i, \ldots, v_k) \text{ where } t_{i-1} < \tau - 1 \leq t_i \wedge t_k \leq \tau < t_{k+1} & \text{otherwise} \end{cases}$$

Figure 13: Formal definition of *subseqProj*, with $seq = (\ell_0, w_0, v_0, t_0), \ldots, (\ell_n, w_n, v_n, t_n)$

$(\ell_1, g, a, R, U, \ell_2) \in E\big\} \cup \big\{(\ell_1^{\bar{V}}, g, a, R, U, \ell_2^{V}) \mid (\ell_1, g, a, R, U, \ell_2) \in E \wedge \ell_2 \in L_{priv}\big\}$.

D.2 Proof of Proposition 1

Proposition 1. *Given be a discrete positive guarded META $\mathcal{A}$, $\exists$ a discrete positive META exponentially larger than $\mathcal{A}$ equivalent to $\mathcal{A}$.*

Proof. We first give a construction for a discrete positive ETA, and then generalise it to discrete positive METAs. Let M be the maximum constant appearing in energy guards (and invariants). Let us build $\mathcal{A}'$ as follows:

1. we build $M+2$ copies of $\mathcal{A}$, where the first $M+1$ copies encode the exact value of the energy from 0 to M, while the last one encodes $> M$.
2. in each copy i, we redirect each transition from ℓ to ℓ' where the energy increases by m units to location ℓ' of the copy $i+m$—unless $i+m > M$, in which case we redirect to the last copy;
3. for each transition with energy guard $\eta \bowtie m$ in copy $0 \leq i \leq M$, we either remove the guard whenever $i \bowtie m$ holds, or we remove the transition otherwise (the case for the last copy is similar).

This transformation clearly preserves the semantics of $\mathcal{A}$. This transformation extends to multiple energy variables in a straightforward manner, by multiplying the number of location by $(M+2)^{|\mathcal{E}|}$, which leads to an exponential blowup.

Fig. 14 illustrates this transformation. □

E Proofs of Section 5

E.1 Proof of Proposition 2

E.1.1 Two-counter machines

Our subsequent undecidability proof works by reduction from the halting problem for 2-counter machines.

We borrow the following material from [ALR22]. A deterministic 2-counter machine ("2CM") [Min67] has two non-negative counters $\mathcal{C}_1$ and $\mathcal{C}_2$, a finite number of states and a finite number of transitions, which can be of the form (for $l \in \{1, 2\}$):

1. "when in state $\mathtt{q}_i$, increment $\mathcal{C}_l$ and go to $\mathtt{q}_j$"; or
2. "when in state $\mathtt{q}_i$, if $\mathcal{C}_l = 0$ then go to $\mathtt{q}_k$, otherwise decrement $\mathcal{C}_l$ and go to $\mathtt{q}_j$".

The 2CM starts in state $\mathtt{q}_0$ with the counters set to 0. The machine follows a deterministic transition function, meaning for each combination of state and counter conditions, there is exactly one action to take. The *halting problem* consists in deciding whether some distinguished state called $\mathtt{q}_{\text{halt}}$ can be reached or not. This problem is known to be undecidable [Min67].

E.1.2 Undecidability of (constant-time) reachability in discrete guarded METAs

We first prove an intermediate lemma; the construction is straightforward, and similar to the construction in the proof of [BDFP04, Proposition 1], that was given in a slightly different context (updatable timed automata).

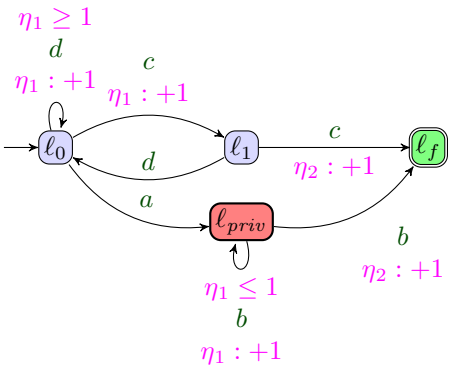
Lemma 1. *Reachability in discrete METAs is undecidable, with two energy variables and a single clock.*

Proof. We reduce from the halting problem of 2-counter machines, which is undecidable [Min67]. Given a 2CM $\mathcal{M}$, we encode it as a discrete guarded META $\mathcal{A}$ over two energy variables $\{\eta_1, \eta_2\}$ and a single clock t.

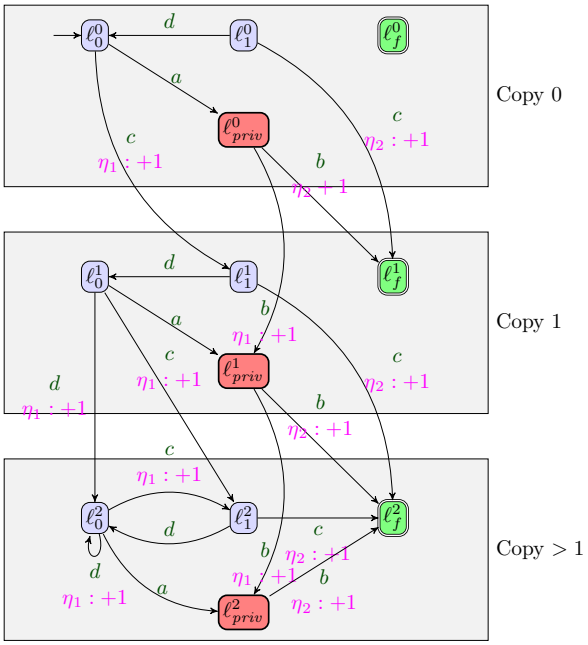
Each state $\mathtt{q}_i$ of the machine is encoded as a location of the discrete guarded META, which we call q_i. The encoding is such that energy η_l directly encodes the value of $\mathcal{C}_l$. The entire computation is done over 0-time, i.e., the unique clock only serves as a global invariant $t = 0$. See Figs. 15a and 15b for the META fragment encoding increment, and zero-test and decrement, respectively.

Obviously, $\mathcal{A}$ can reach q_{halt} iff $\mathcal{M}$ can reach $\mathtt{q}_{\text{halt}}$. Since reachability of $\mathtt{q}_{\text{halt}}$ in $\mathcal{M}$ is undecidable, then reachability of q_{halt} in $\mathcal{A}$ is undecidable. □

Corollary 1. *Constant-time reachability in discrete METAs is undecidable, with two energy variables and a single clock.*



(a) Example with $M = 1$



(b) Transformation

Figure 14: Exemplifying the energy guards removal

Proof. From the fact the construction in the proof of Lemma 1 works in 0-time. Any other constant can be achieved by delaying the start of the simulation of the 2-counter machine, thanks to the (unique) clock. □

E.1.3 Opacity is harder than reachability for discrete METAs

Proposition 5. *For each $\delta \in \{\exists, weak, full\}$, reachability in discrete METAs can be solved using δ-EN-opacity.*

Proof. Let $\mathcal{A}$ be a discrete META with initial location ℓ_0; let ℓ_f be one of its locations. We let $\mathcal{A}'$ be the discrete META extending $\mathcal{A}$ as follows:

- we add a fresh location ℓ_{priv} reachable from ℓ_0 without guard;
- we add a fresh location ℓ_1 reachable from ℓ_f without guard;
- we add a fresh location ℓ'_f reachable from ℓ_1 and from ℓ_{priv} without guard;
- for each energy variable η_i ($1 \leq i \leq |\mathcal{E}|$), we add a self-loop over ℓ_{priv} incrementing η_i;
- for each energy variable η_i, we add two self-loops over ℓ_1, one incrementing and one decrementing η_i.

$\{\ell_{priv}\}$ is the private location set of $\mathcal{A}'$, while $\{\ell'_f\}$ is its final location set. Fig. 16 exemplifies this construction, where $\eta_i : +1$ denotes $|\mathcal{E}|$ distinct self-loops incrementing one energy variable each.

First note that $EVisit^{priv}(\mathcal{A}) = \mathbb{N}^{\mathcal{E}}$, i.e., the final energy can be any integer value for each energy variable. Furthermore, if ℓ_f is reachable, then $EVisit^{pub}(\mathcal{A}) = \mathbb{N}^{\mathcal{E}}$ (note that the energy can be arbitrary when reaching ℓ_f, but the self-loops over ℓ_1 ensure that any value, smaller or larger, can be reached); in that case, all $\exists$, weak and full EN-opacity hold. Conversely, if ℓ_f is not reachable, then $EVisit^{pub}(\mathcal{A}) = \emptyset$; in that case, none of the opacity definitions hold in $\mathcal{A}'$.

As a consequence, for each $\delta \in \{\exists, \text{weak}, \text{full}\}$, δ-EN-opacity holds in $\mathcal{A}'$ iff ℓ_f is reachable in $\mathcal{A}$. □

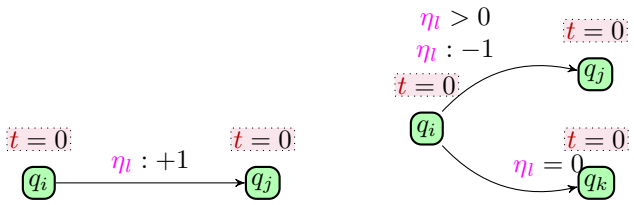
Proposition 6. *Reachability in discrete METAs can be solved using $\exists$-ET-EN-opacity.*

Proof. One can reuse the construction from the proof of Proposition 5 (given in Fig. 16). (The self-loops over ℓ_1 are actually not needed here.) Due to the absence of invariant in ℓ_{priv}, ℓ'_f can be reached for any duration and energy valuation via ℓ_{priv}; then, there is at least one duration and one energy valuation reaching ℓ'_f without visiting ℓ_{priv} (via ℓ_f and ℓ_1) iff ℓ_f is reachable in $\mathcal{A}$. As a consequence, $\exists$-ET-EN-opacity holds in $\mathcal{A}'$ iff ℓ_f is reachable in $\mathcal{A}$. □

Proposition 7. *Constant-time reachability in discrete METAs can be solved using weak-ET-EN-opacity (resp. full-ET-EN-opacity).*

Proof. Let $\mathcal{A}$ be a discrete META with initial location ℓ_0; let ℓ_f be one of its locations. We let $\mathcal{A}'$ be the discrete META extending $\mathcal{A}$ as follows:

- we add a fresh location ℓ_{priv} reachable from ℓ_0 without guard;
- we add a fresh location ℓ_1 reachable from ℓ_f without guard;
- we add a fresh location ℓ'_f reachable from ℓ_1 and from ℓ_{priv} without guard;
- we add a fresh location ℓ_2 reachable from ℓ_0, guarded by $x < C$ and resetting x (where x is an arbitrary clock from $\mathcal{A}$);
- we add a transition from ℓ_0 to ℓ_1 guarded by $x > C$;
- we add an urgent transition from ℓ_2 to ℓ'_f (i.e., guarded by $x = 0$);



(a) Incrementing $\mathcal{C}_l$ **(b)** Decrementing $\mathcal{C}_l$

Figure 15: Encoding a 2CM as a discrete guarded META

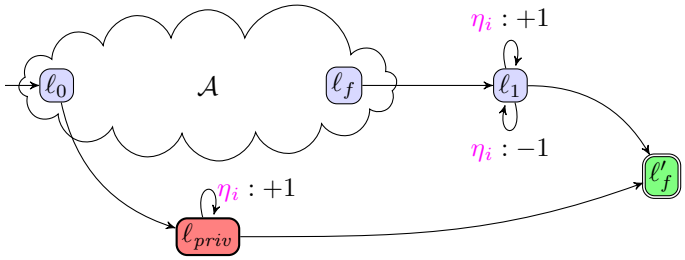


Figure 16: Solving reachability using EN-opacity for discrete METAs

- for each energy variable η_i, we add a self-loop over ℓ_{priv} (resp. over ℓ_2) incrementing η_i;
- for each energy variable η_i, we add two self-loops over ℓ_1, one incrementing and one decrementing η_i.

$\{\ell_{priv}\}$ is the private location set of $\mathcal{A}'$, while $\{\ell'_f\}$ is its final location set. Fig. 17 exemplifies this construction.

Let $C \in \mathbb{N}$. First note that any energy and any duration can reach ℓ'_f via ℓ_{priv}. Second, note that any duration $> C$ (resp. $< C$) and any energy can reach ℓ'_f without visiting ℓ_{priv}, via ℓ_1 (resp. via ℓ_2). Third, any energy for duration $= C$ can reach ℓ'_f via ℓ_f and ℓ_1 without visiting ℓ_{priv}, iff ℓ_f is reachable in time C.

From this construction, it follows that weak-ET-EN-opacity and full-ET-EN-opacity hold in $\mathcal{A}'$ iff ℓ_f is reachable in $\mathcal{A}$ in time C. □

E.1.4 Proof of Proposition 2

We can finally prove Proposition 2.

Proposition 2. *For each $\delta \in \{\exists, weak, full\}$, and $\sigma \in \{EN, ET\text{-}EN\}$, δ-σ-opacity for discrete guarded METAs with 2 energy variables and 1 clock is undecidable.*

Proof. Recall that:

- For each $\delta \in \{\exists, \text{weak}, \text{full}\}$, δ-EN-opacity is harder than reachability in discrete METAs (Proposition 5);
- $\exists$-ET-EN-opacity is harder than reachability in discrete METAs (Proposition 6);
- Weak-ET-EN-opacity and full-ET-EN-opacity are harder than constant-time reachability in discrete METAs (Proposition 7);
- Reachability is undecidable in discrete guarded METAs with two energy variables and a single clock (Lemma 1); and
- Constant-time reachability is undecidable in discrete guarded METAs with two energy variables and a single clock (Corollary 1).

As a consequence, for each $\delta \in \{\exists, \text{weak}, \text{full}\}$, and $\sigma \in \{\text{EN}, \text{ET-EN}\}$, δ-σ-opacity is undecidable in discrete guarded METAs with 2 energy variables and 1 clock. □

E.2 Proof of Theorem 4

A first attempt to prove Theorem 4 would be to follow the reasoning from Proposition 1, by using copies of the guarded META, each copy encoding the current (discrete) value of the energy. However, this leads to the following problem: the last copy (encoding a value of energy greater than the maximum constant M used in energy guards) does not keep track of the exact energy value—this prevents decreasing the energy to the appropriate copy. We will therefore combine two ideas, namely from the proofs of Proposition 1 and Theorem 3.

Theorem 4. *For each $\delta \in \{\exists, weak, full\}$ and $\sigma \in \{EN, ET\text{-}EN\}$, δ-σ-opacity is decidable for discrete guarded ETAs.*

Proof (sketch). To circumvent the issue in the proof of Proposition 1 losing the value of the energy after it exceeds the maximum constant M used in energy guards, we will use a stack. To avoid using two stacks (in addition to the stack used in the proof of Theorem 3), we use the *same* stack, in addition to the copy mechanism, which we use this time in the region automaton.

We embed the energy guards into the actions, i.e.,

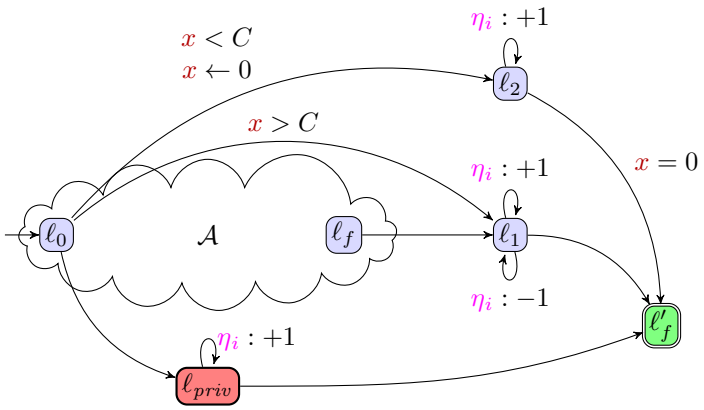


Figure 17: Solving reachability in constant time C using ET-EN-opacity for discrete METAs

each energy guard becomes a fresh action, potentially associated to the original action (if any). The region automaton is therefore equivalent to the system without any energy guard—but we will reintroduce them subsequently. Once the region automaton is built, we first copy $M+2$ times this automaton, in the line of the mechanism in the proof of Proposition 1. We then evaluate the transitions when needed: an energy "hard-coded" as an action in one of the first $M+1$ copies can now either be replaced with a silent action (or with the original action) if the current copy satisfies the energy guard, or its transition completely deleted otherwise. Note that if a decrement is found in the copy encoding the energy 0, then the action must be removed, as our systems doesn't allow energies below zero. In the last copy of the region automaton (encoding $> M$), we use the stack: when performing increments or decrements (actions *inc* and *dec* in the region automata), we start pushing symbols or popping symbols, in the line of the proof of Theorem 3. Finally, we modify the last items from the proof of Theorem 3 as follows: in addition to the self-loops over accepting regions s_f, we then add for each such region a transition labelled via a to the copy encoding an immediately lower energy level. Only regions s'_f in the copy encoding energy 0 are made accepting. This ensures that, for a value of energy greater than the maximum constant, first the stack is emptied with actions a; only after the stack is emptied, the automaton has to go down to the level 0, using an action a each time a level is traversed. The number of a correctly encodes the final energy level.

Fig. 18 exemplifies our construction. □

F Proofs of Section 6

F.1 Proof of Proposition 3

Proposition 3. *For each $\delta \in \{\exists, weak, full\}$, δ-DE-opacity for discrete guarded METAs with 2 energies and 1 clock is undecidable.*

Proof. The proof if a straightforward adaptation of the proof of Proposition 2. First note that, for a system running in time ≤ 1, DE-opacity is equivalent to EN-opacity, since observing the energy every time unit is equivalent to observe energy only at the end of the execution. The proof then follows by choosing $C = 1$ in the proof of Propositions 2 and 7. □

F.2 Proof of Theorem 5

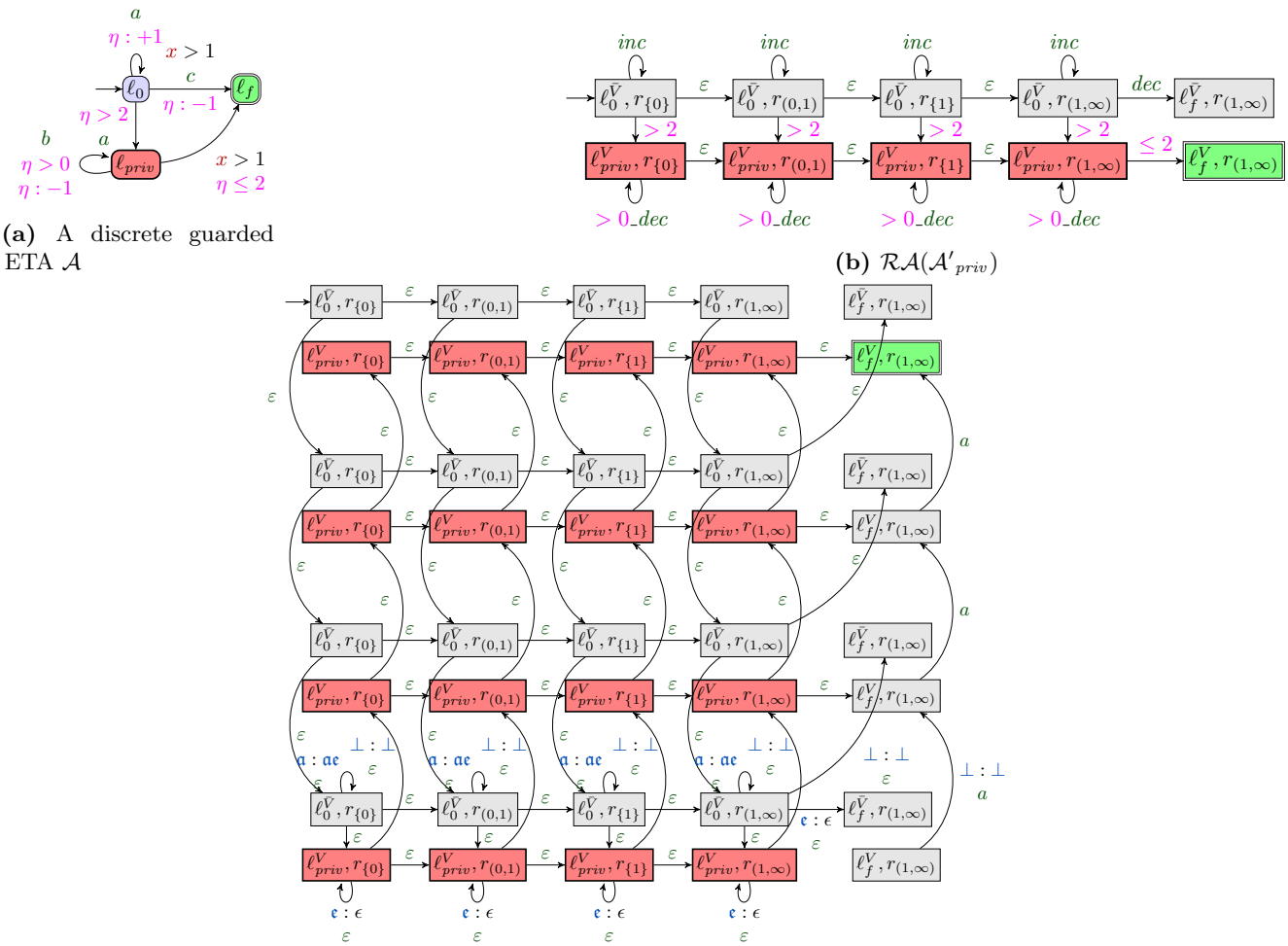
Theorem 5. *For each $\delta \in \{\exists, weak, full\}$, δ-DE-opacity can be decided in* EXPSPACE *for discrete positive (guarded) ETAs.*

Proof. Let $\mathcal{A}$ be a discrete positive ETA. We construct $\mathcal{L}(\mathcal{RA}(\mathcal{A}''_{priv}))$ and $\mathcal{L}(\mathcal{RA}(\mathcal{A}''_{pub}))$ the same way as in the proof of Theorem 2, except that we replace the last action $t_{>0}$ with t; this way, we encode the fact that we observe the last time unit, without making a difference between energy changes occurring in 0-time before the last observable time, or in the preceding open interval. Verifying weak DE-opacity amounts to checking $\mathcal{L}(\mathcal{RA}(\mathcal{A}''_{priv})) \subseteq \mathcal{L}(\mathcal{RA}(\mathcal{A}''_{pub}))$, which is a language inclusion problem in non-deterministic finite automata (over 2 symbols), which is PSPACE-complete [KMT17]. Full DE-opacity is similar. $\exists$-DE-opacity is solved by checking the non-emptiness of the intersection, which can be verified in polynomial time [KMT17]. The complexity remains unchanged for positive *guarded* ETAs since the transformation in the proof of Proposition 1 is linear for ETAs. □

F.3 Details on the proof of Theorem 6

We give additional details regarding the formalisation of our construction of "Parikh by block" automaton. Given an NFA $\mathcal{A}$ with alphabet Σ, and given $\Sigma' \subseteq \Sigma$, the construction $PbB(\mathcal{A})$ will construct an NFA (technically an extension of NFAs with two labels, one being an action as usual, and the other one being a semilinear set), encoding the number of actions of each symbol in $\Sigma \setminus \Sigma'$ in between two consecutive actions in Σ'.

1. Add to a list L the initial state of $\mathcal{A}$ and every state that has an incoming transition with some action t, for $t \in \Sigma'$.



(a) A discrete guarded ETA $\mathcal{A}$

(b) $\mathcal{RA}(\mathcal{A}'_{priv})$

(c) PDA resulting from the construction

Figure 18: Exemplifying the construction of Theorem 4

2. For all $s \in L$, and for all state s'' such that there exists in $\mathcal{A}$ a transition (s', t, s''), compute the Parikh's image of the automaton where
   - All transitions with action t, for $t \in \Sigma'$, are removed,
   - s is the initial state,
   - s' is the only final state.

   Let $P_t(s, s'')$ denote this semilinear set.
3. Finally build a new automaton $PbB(\mathcal{A})$ such that:
   - The initial state and final states are the same as the original $\mathcal{A}$,
   - For all non-empty set $P_t(s, s'), t \in \Sigma'$, add states s and s', and add a transition from s to s' with t as action, and with the semilinear $P_t(s, s')$ as label.

Theorem 6. *$\exists$-DE-opacity is decidable for discrete positive (guarded) METAs.*

Proof. We begin by constructing the Parikh by block automaton of each NFA over $\Sigma' = \{t, f\}$, i.e., we build $PbB(\mathcal{RA}(\mathcal{A}'_{priv}))$ and $PbB(\mathcal{RA}(\mathcal{A}'_{pub}))$. These structures encode exactly the number of increments between two time ticks.

Now construct the synchronised product of $PbB(\mathcal{RA}(\mathcal{A}'_{priv}))$ and $PbB(\mathcal{RA}(\mathcal{A}'_{pub}))$ over $\{t, f\}$, as follows: in the resulting product, each transition has the original synchronised label (in $\{t, f\}$), while the other component is the pair of semilinear sets, coming from each of the two synchronised automata. Then, for each synchronised transition, we verify the emptiness of the intersection between these two semilinear sets. Because the Parikh image of a regular language is a semilinear set [Par66], this can be checked using Presburger arithmetic [BHK17]. After removing transitions with such an empty intersection, verifying $\exists$-DE-opacity simply consists in exhibiting an accepting run in that structure. □

Finally recall that we exemplify the construction in Example 8.

F.4 Proof of Theorem 7

Theorem 7. *For each $\delta \in \{\exists, weak, full\}$, δ-bDE-opacity can be decided in* EXPSPACE *for discrete positive METAs, and in* 2EXPSPACE *for discrete positive guarded METAs.*

Proof. We use inclusion and intersection problems over non-deterministic finite automata, with a modified construction. Given a discrete positive META $\mathcal{A}$, we transform it into $\mathcal{A}'$ as follows:

1. we add a global "ticking clock" z_t reset every time unit via action t as in the proof of Theorem 2;
2. we eliminate discrete increments $\neq 1$ by splitting them into consecutive $+1$ updates in 0-time, using an extra clock z_0;
3. we add after each update (or series of updates in 0-time) a last transition in 0-time via a fresh action f;
4. we relabel each transition of the form $\eta_i : +1$ with action inc_i, and make any other action silent;
5. for each final location ℓ_f, we add a new location ℓ'_f (the only ones to be final) reachable from ℓ_f via action t if ℓ_f is reachable on an integer time, and via ε otherwise.

We exemplify our transformation (applied to Fig. 3a) in Fig. 9.

Now, there is a one-to-one correspondence between the buffered discrete energy observation of the runs of $\mathcal{A}$, and the untimed language of $\mathcal{A}'$. For example, given the private run $\rho_1 = (\ell_0, 0, 0) \overset{(0.8,a)}{\longmapsto} (\ell_{priv}, 0.8, 0) \overset{(0.3,b)}{\longmapsto} (\ell_{priv}, 1.1, 1) \overset{(0,b)}{\longmapsto} (\ell_{priv}, 1.1, 2) \overset{(0,b)}{\longmapsto} (\ell_f, 1.1, 2)$ of $\mathcal{A}$ in Fig. 3a, the untimed word will be $tinc_1 finc_1 f$. The public run $\rho_2 = (\ell_0, 0, 0) \overset{(1.1,c)}{\longmapsto} (\ell_f, 1.1, 2)$ gives in contrast $tinc_1 inc_1 f$.

Solving δ-bDE-opacity therefore amounts to checking inclusion and intersection over the NFAs $\mathcal{RA}(\mathcal{A}'_{pub})$ and $\mathcal{RA}(\mathcal{A}'_{priv})$.

Again, the complexity for discrete positive *guarded* METAs follows from Proposition 1 and the fact that the transformation in the proof of Proposition 1 for discrete positive guarded METAs leads to an exponential blowup of the META. □

F.5 Proof of Theorem 8

Theorem 8. *For each $\delta \in \{\exists, weak, full\}$, δ-bDE-opacity can be decided in* 2EXPSPACE *for discrete ETAs.*

Proof. Given a discrete ETA $\mathcal{A}$, we build $\mathcal{RA}(\mathcal{A}'_{pub})$ and $\mathcal{RA}(\mathcal{A}'_{priv})$ the same way as in Theorem 7 (see Section F.4), where increments are encoded by action *inc* and decrements by *dec*. These regular languages recognize some words that are not valid, typically leading to the energy dropping below 0, i.e., not in $\mathcal{L}_{\geq 0} = \{\omega = (dec + inc + f + t)^* \mid \text{for each prefix } \omega' \text{ of } \omega, \#_{dec}(\omega') \leq \#_{inc}(\omega')\}$. This language is context-free: see Fig. 19 for a pushdown automaton recognizing the language $\mathcal{L}_{\geq 0}$.

To verify $\exists$-bDE-Opacity, verifying $((\mathcal{RA}(\mathcal{A}'_{pub}) \cap \mathcal{RA}(\mathcal{A}'_{priv})) \cap \mathcal{L}_{\geq 0})$ emptiness is sufficient: the intersection of both region automata is quadratic, but determinising it adds an exponential factor. Intersection between a regular language and a context-free language expressed by a PDA is polynomial. This gives a 2EXPSPACE procedure.

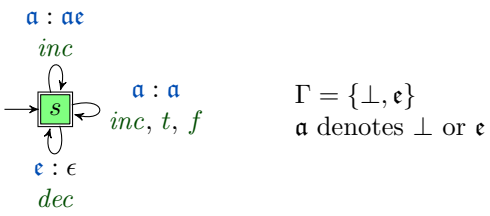


Figure 19: Non-deterministic pushdown automaton recognizing the language $\mathcal{L}_{\geq 0}$

For weak and full bDE-opacity, verifying the inclusions $(\mathcal{RA}(\mathcal{A}'_{pub}) \cap \mathcal{L}_{\geq 0}) \subseteq (\mathcal{RA}(\mathcal{A}'_{priv}) \cap \mathcal{L}_{\geq 0})$ and $(\mathcal{RA}(\mathcal{A}'_{priv}) \cap \mathcal{L}_{\geq 0}) \subseteq (\mathcal{RA}(\mathcal{A}'_{pub}) \cap \mathcal{L}_{\geq 0})$ cannot be done directly, as the inclusion of two context-free languages is not decidable in general [CM14]. However, in our case, the only non-regular language is $\mathcal{L}_{\geq 0}$. Therefore $((\mathcal{RA}(\mathcal{A}'_{pub}) \cap (\neg\mathcal{RA}(\mathcal{A}'_{priv})) \cap \mathcal{L}_{\geq 0})$ and $((\mathcal{RA}(\mathcal{A}'_{priv}) \cap (\neg\mathcal{RA}(\mathcal{A}'_{pub})) \cap \mathcal{L}_{\geq 0})$ are computable, and the emptiness of both languages gives us the inclusion that we seek. Computing these intersections can be done in polynomial time. The automaton to be intersected with $\mathcal{L}_{\geq 0}$ is doubly exponential in the original ETA (one exponential for the region automaton, and another one for the determinisation). This gives a 2EXPSPACE procedure. □