Time-bounded Reachability for Hybrid Automata: Complexity and Fixpoints

Thomas Brihaye[⋆] Laurent Doyen[+] Gilles Geeraerts[%]
Joël Ouaknine[†] Jean-François Raskin[%] James Worrell[†]

⋆: Université de Mons, Belgium
+: LSV, ENS Cachan & CNRS, France
%: Université Libre de Bruxelles, Belgium
†: Oxford University Computing Lab., UK

Abstract

In this paper, we study the *time-bounded reachability problem* for rectangular hybrid automata with non-negative rates ($\text{RHA}^{\geq 0}$). This problem was recently shown to be decidable [5] (even though the *unbounded* reachability problem for even very simple classes of hybrid automata is well-known to be undecidable). However, [5] does not provide a precise characterisation of the complexity of the time-bounded reachability problem. The contribution of the present paper is threefold. First, we provide a new NEXPTIME algorithm to solve the timed-bounded reachability problem on $\text{RHA}^{\geq 0}$. This algorithm improves on the one of [5] by at least one exponential. Second, we show that this new algorithm is optimal, by establishing a matching lower bound: time-bounded reachability for $\text{RHA}^{\geq 0}$ is therefore NEXPTIME-complete. Third, we extend these results in a practical direction, by showing that we can effectively compute fixpoints that characterise the sets of states that are reachable (resp. co-reachable) within T time units from a given starting state.

1 Introduction

Hybrid systems form a general class of systems that mix *continuous* and *discrete* behaviors. Examples of hybrid systems abound in our everyday life, particularly in applications where an (inherently discrete) computer system must interact with a continuous environment. The need for modeling hybrid systems is obvious, together with methods to analyse those systems.

Hybrid automata are arguably among the most prominent families of models for hybrid systems [7]. Syntactically, a hybrid automaton is a finite automaton (to model the discrete part of the system) augmented with a finite set of real-valued variables (to model the continuous part of the system). Those variables evolve with time elapsing, at a rate which is given by a flow function that depends on the current location of the automaton. The theory of hybrid automata has been well developed for about 20 years,

and tools to analyse them are readily available, see for instance HYTECH [8, 9] and PHAVER [6].

Hybrid automata are thus a class of powerful models, yet their high expressiveness comes at a price, in the sense that the undecidability barrier is rapidly hit. Simple *reachability properties* are undecidable even for the restricted subclass of *stopwatch automata*, where the rate of growth of each variable stays constant in all locations and is restricted to either 0 or 1 (see [10] for a survey).

On the other hand, a recent and successful line of research in the setting of *timed automata* has outlined the benefits of investigating *timed-bounded variants* of classical properties [12, 14]. For instance, while *language inclusion* is, in general undecidable for timed automata, it becomes decidable when considering only executions of *bounded duration* [14].

In a recent work [5] we have investigated the decidability of *time-bounded reachability* for rectangular hybrid automata (i.e., is a given state reachable by an execution of duration at most T ? for a given T). We have shown that *time-bounded* reachability is *decidable* for *rectangular hybrid automata with non-negative rates* (RHA$^{\geq 0}$), while it is well-known [10] that (plain, time unbounded) reachability is not for this class. We have also shown that the decidability frontier is quite sharp in the sense that time-bounded reachability becomes *undecidable* once we allow either *diagonal constraints* in the guards or *negative rates*.

To obtain decidability of time-bounded reachability for RHA$^{\geq 0}$, we rely, in [5], on a *contraction operator* that applies to runs, and allows to derive, from any run of duration at most T of an RHA$^{\geq 0}$ $\mathcal{H}$, an equivalent run that reaches the *same state*, but whose length (in terms of number of discrete transitions) is *uniformly bounded* by a function F of the size of $\mathcal{H}$ and T. Hence, deciding reachability within T time units reduces to exploring runs of bounded lengths only, which is feasible algorithmically (see [5] for the details). However, this previous work does not contain a precise characterisation of the complexity of time-bounded reachability. Clearly, an upper bound on the complexity depends on the bound F on the length of the runs that need to be explored.

In the present work, we revisit and extend our previous results [5] in several directions, both from the theoretical and the practical point of view. *First*, we completely revisit the definition of the *contraction operator* and obtain a new operator that allows to derive a *singly exponential upper bound* on the lengths of the runs that need to be considered, while the operator in [5] yields an upper bound that is at least doubly exponential. Our new contraction operator thus provides us with an NEXPTIME algorithm that improves on the algorithm of [5] by at least one exponential. *Second*, we show that this new algorithm is optimal, by establishing a matching lower bound. Hence, *time-bounded reachability for RHA$^{\geq 0}$ is* NEXPTIME*-complete*. *Third*, we extend those results towards more practical concerns, by showing that we *can effectively compute fixpoints* that characterise the set of states that are reachable (resp. co-reachable) within T time units, from a given state. The time needed to compute them is at most doubly exponential in the size of the RHA$^{\geq 0}$ and the bound T. *Fourth*, *we apply those ideas to two examples* of RHA$^{\geq 0}$ for which the classical (time-unbounded) forward and backward fixpoints do not terminate. We show that, in those examples, the sets of states that are time-bounded reachable is computable in practice, for values of the time bound that allow us to derive *meaningful properties*.

This brief summary of the results outlines the structure of the paper. Remark that, by lack of space, some more technical proofs have been moved to the appendix.

2 Definitions

Let $\mathcal{I}$ be the set of intervals of real numbers with endpoints in $\mathbb{Z} \cup \{-\infty, +\infty\}$. Let X be a set of continuous variables, and let $\dot{X} = \{\dot{x} \mid x \in X\}$ be the set dotted variables, corresponding to variable first derivatives. A *rectangular constraint* over X is an expression of the form $x \in I$ where x belongs to X and I to $\mathcal{I}$. A *diagonal constraint* over X is a constraint of the form $x - y \sim c$ where x, y belong to X, c to $\mathbb{Z}$, and $\sim$ is in $\{<, \leq, =, \geq, >\}$. Finite conjunctions of diagonal and rectangular constraints over X are called *guards*, over $\dot{X}$ they are called *rate constraints*. A guard or rate constraint is *rectangular* if all its constraints are rectangular. We denote by $\mathcal{G}(X)$ and $\mathcal{R}(X)$ respectively the sets of guards and rate constraints over X.

Linear, rectangular and singular hybrid automata A *linear hybrid automaton* (LHA) is a tuple $\mathcal{H} = (X, \mathrm{Loc}, \mathrm{Edges}, \mathrm{Rates}, \mathrm{Inv}, \mathrm{Init})$ where $X = \{x_1, \ldots, x_{|X|}\}$ is a finite set of continuous *variables* ; Loc is a finite set of *locations*; $\mathrm{Edges} \subseteq \mathrm{Loc} \times \mathcal{G}(X) \times 2^X \times \mathrm{Loc}$ is a finite set of *edges*; $\mathrm{Rates} : \mathrm{Loc} \mapsto \mathcal{R}(X)$ assigns to each location a constraint on the *possible variable rates*; $\mathrm{Inv} : \mathrm{Loc} \mapsto \mathcal{G}(X)$ assigns an *invariant* to each location; and $\mathrm{Init} \subseteq \mathrm{Loc}$ is a *set of initial locations*. For an edge $e = (\ell, g, Y, \ell')$, we denote by $\mathsf{src}(e)$ and $\mathsf{trg}(e)$ the location ℓ and ℓ' respectively, g is called the *guard* of e and Y is the *reset* set of e. In the sequel, we denote by rmax and cmax the maximal constant occurring respectively in the constraints of $\{\mathrm{Rates}(\ell) \mid \ell \in \mathrm{Loc}\}$ and of $\{\mathrm{Rates}(\ell) \mid \ell \in \mathrm{Loc}\} \cup \{g \mid \exists (\ell, g, Y, \ell') \in \mathrm{Edges}\}$.

An LHA is *non-negative rate* if for all variables x, for all locations ℓ, the constraint $\mathrm{Rates}(\ell)$ implies that $\dot{x}$ must be non-negative. A *rectangular hybrid automaton* (RHA) is a linear hybrid automaton in which all guards, rates, and invariants are rectangular. In the case of RHA, we view rate constraints as functions $\mathrm{Rates} : \mathrm{Loc} \times X \to \mathcal{I}$ that associate with each location ℓ and each variable x an interval of possible rates $\mathrm{Rates}(\ell)(x)$. A *singular hybrid automaton* (SHA) is an RHA s.t. for all locations ℓ and for all variables x: $\mathrm{Rates}(\ell)(x)$ is a singleton. We use the shorthands RHA$^{\geq 0}$ and SHA$^{\geq 0}$ for *non-negative rates* RHA and SHA respectively.

LHA semantics A *valuation* of a set of variables X is a function $\nu : X \mapsto \mathbb{R}$. We denote by $\mathbf{0}$ the valuation that assigns 0 to each variable.

Given an LHA $\mathcal{H} = (X, \mathrm{Loc}, \mathrm{Edges}, \mathrm{Rates}, \mathrm{Inv}, \mathrm{Init}, X)$, a *state* of $\mathcal{H}$ is a pair (ℓ, ν), where $\ell \in \mathrm{Loc}$ and ν is a valuation of X. The semantics of $\mathcal{H}$ is defined as follows. Given a state $s = (\ell, \nu)$ of $\mathcal{H}$, an *edge step* $(\ell, \nu) \xrightarrow{e} (\ell', \nu')$ can occur and change the state to (ℓ', ν') if $e = (\ell, g, Y, \ell') \in \mathrm{Edges}$, $\nu \models g$, $\nu'(x) = \nu(x)$ for all $x \notin Y$, and $\nu'(x) = 0$ for all $x \in Y$; given a time delay $t \in \mathbb{R}^+$, a *continuous time step* $(\ell, \nu) \xrightarrow{t} (\ell, \nu')$ can occur and change the state to (ℓ, ν') if there exists a vector $r = (r_1, \ldots r_{|X|})$ such that $r \models \mathrm{Rates}(\ell)$, $\nu' = \nu + (r \cdot t)$, and $\nu + (r \cdot t') \models \mathrm{Inv}(\ell)$ for all $0 \leq t' \leq t$.

A *path* in $\mathcal{H}$ is a finite sequence $e_1, e_2, \ldots, e_n$ of edges such that $\mathsf{trg}(e_i) = \mathsf{src}(e_{i+1})$ for all $1 \leq i \leq n-1$. A *timed path* of $\mathcal{H}$ is a finite sequence of the form $\pi = (t_1, e_1), (t_2, e_2), \ldots, (t_n, e_n)$, such that $e_1, \ldots, e_n$ is a path in $\mathcal{H}$ and $t_i \in \mathbb{R}^+$ for all $0 \leq i \leq n$. For all k, ℓ, we denote by $\pi[k : \ell]$ the maximal portion $(t_i, e_i), (t_{i+1}, e_{i+1}), \ldots, (t_j, e_j)$ of π s.t. $\{i, i+1, \ldots, j\} \subseteq [k, \ell]$ (remark that the interval $[k, \ell]$ could be empty, then $\pi[k : \ell]$ is empty too). Given a timed path $\pi = (t_1, e_1), (t_2, e_2), \ldots, (t_n, e_n)$ of an SHA, we let $\mathsf{Effect}(\pi) = \sum_{i=1}^{n} \mathrm{Rates}(\ell_{i-1}) \cdot t_i$ be the *effect of* π (where $\ell_i = \mathsf{src}(e_i)$ for $1 \leq i \leq n$).

A *run* in $\mathcal{H}$ is a sequence $s_0, (t_1, e_1), s_1, (t_2, e_2), \ldots, (t_n, e_n), s_n$ such that:

- $(t_1, e_1), (t_2, e_2), \ldots, (t_n, e_n)$ is a timed path in $\mathcal{H}$, and
- for all $0 \leq i < n$, there exists a state s'_i of $\mathcal{H}$ with $s_i \xrightarrow{t_{i+1}} s'_i \xrightarrow{e_{i+1}} s_{i+1}$.

Given a run $\rho = s_0, (t_1, e_1), \ldots, s_n$, let $\mathsf{first}(\rho) = s_0 = (\ell_0, \nu_0)$, $\mathsf{last}(\rho) = s_n$, $\mathsf{duration}(\rho) = \sum_{i=1}^{n} t_i$, and $|\rho| = n + 1$. We say that ρ is $\mathbf{T}$-*time-bounded* (for $\mathbf{T} \in \mathbb{N}$) if $\mathsf{duration}(\rho) \leq \mathbf{T}$. Given two runs $\rho = s_0, (t_1, e_1), \ldots, (t_n, e_n), s_n$ and $\rho' = s'_0, (t'_1, e'_1), \ldots, (t'_k, e'_k), s'_k$ with $s_n = s'_0$, we let $\rho \cdot \rho'$ denote the run $s_0, (t_1, e_1), \ldots, (t_n, e_n), s_n, (t'_1, e'_1), \ldots, (t'_k, e'_k), s'_k$.

Note that a unique timed path $\mathsf{TPath}(\rho) = (t_1, e_1), (t_2, e_2), \ldots, (t_n, e_n)$, is associated with each run $\rho = s_0, (t_1, e_1), s_1, \ldots, (t_n, e_n), s_n$. Hence, we sometimes abuse notation and denote a run ρ with $\mathsf{first}(\rho) = s_0$, $\mathsf{last}(\rho) = s$ and $\mathsf{TPath}(\rho) = \pi$ by $s_0 \xrightarrow{\pi} s$. The converse however is not true: given a timed path π and an initial state s_0, it could be impossible to build a run starting from s_0 and following π because some guards or invariants along π might be violated. However, if such a run exists it is necessarily unique *when the automaton is singular*. In that case, we denote by $\mathsf{Run}(s_0, \pi)$ the function that returns the unique run ρ such that $\mathsf{first}(\rho) = s_0$ and $\mathsf{TPath}(\rho) = \pi$ if it exists, and $\perp$ otherwise. Remark that, when consider an SHA: if $\rho = (\ell_0, \nu_0) \xrightarrow{\pi} (\ell_n, \nu_n)$ is a run, then for all x that is *not reset* along ρ: $\nu_n(x) = \nu_0(x) + \mathsf{Effect}(\pi)(x)$.

Time-bounded reachability problem for LHA While the reachability problem asks whether there exists a run reaching a given goal location, we are only interested in runs having *bounded duration*.

Problem 1 (Time-bounded reachability problem) *Given an LHA* $\mathcal{H} = (X, \mathrm{Loc}, \mathrm{Edges}, \mathrm{Rates}, \mathrm{Inv}, \mathrm{Init})$*, a location* $\mathrm{Goal} \in \mathrm{Loc}$ *and a time bound* $\mathbf{T} \in \mathbb{N}$*, the* time-bounded reachability problem *is to decide whether there exists a finite run* $\rho = (\ell_0, \mathbf{0}) \xrightarrow{\pi} (\mathrm{Goal}, \cdot)$ *of* $\mathcal{H}$ *with* $\ell_0 \in \mathrm{Init}$ *and* $\mathsf{duration}(\rho) \leq \mathbf{T}$.

This problem is known to be decidable [5] for RHA$^{\geq 0}$, but its exact complexity is, so far, unknown. We prove in Section 4 (thanks to the results of Section 3) that it is NEXPTIME-complete. This problem is known to become undecidable once we allow either diagonal constraints in the guards, or negative and positive rates to occur in the LHA [5].

A more general problem that is relevant in practice, is to compute a symbolic representation of all the states that are reachable in at most $\mathbf{T}$ time units. Here, by 'symbolic representation' we mean a finite representation of the set of states that can be manipulated algorithmically. This problem, together with the definition of such a such a symbolic representation, will be addressed in Section 5.

Let us illustrate, by means of the RHA$^{\geq 0}$ $\mathcal{H}$ in Fig. 1, the difficulties encountered when computing the reachable states of a RHA$^{\geq 0}$. Let us characterise the set $\mathsf{Reach}_{\ell_1}(s_0)$ of all states of the form (ℓ_1, ν) that are reachable from s_0. It is easy to see that $\mathsf{Reach}_{\ell_1}(s_0) = \{(\ell_1, (0, \frac{1}{2^n})) \mid n \in \mathbb{N}_0\}$. Moreover, observe that, for all $n \in \mathbb{N}_0$, $(\ell_1, (0, \frac{1}{2^n}))$ is reachable from s_0 by one and only one run, of duration $(n-1) + \frac{1}{2^n}$, and that the number of bits necessary to encode those states grows *linearly* with the length of the run. This examples shows that finding an adequate, compact and effective representation (such as regions in the case of Timed Automata [2]) for the set of reachable of an RHA$^{\geq 0}$ is not trivial (and, in full generality, impossible because reachability

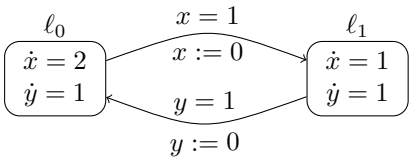

Figure 1: A simple hybrid automaton.

is undecidable for this class). Nevertheless, in Section 5, we show that, in an RHA$^{\geq 0}$, an effective representation of the set of states that are reachable *within* $\mathbf{T}$ *time units* can be computed.

3 Contracting runs

In this section, we describe a *contraction operator*. Given an RHA $\mathcal{H}$, and one of its timed paths π of arbitrary length but of duration $\leq \mathbf{T}$, the contraction operator builds a timed path $\mathsf{Cnt}^*(\pi)$ that reaches the same state as π, but whose size is uniformly bounded by a function of $\mathbf{T}$, and of the size of $\mathcal{H}$. This operator is central to prove correctness of the algorithms for time-bounded reachability in sections 4 and 5. Since Problem 1 is undecidable if both positive and negative rates are allowed [5], we restrict our attention to RHA with non negative rates. Moreover, for the sake of clarity, all the results presented in this section are limited to *singular hybrid automata*, but they extend easily to RHA$^{\geq 0}$ as we will see later. Thus, from now one, we fix an SHA$^{\geq 0}$ $\mathcal{H} = (X, \mathrm{Loc}, \mathrm{Edges}, \mathrm{Rates}, \mathrm{Inv}, \mathrm{Init})$.

Self loops The first step of our construction consists in adding, on each location ℓ of $\mathcal{H}$, a self-loop $(\ell, \mathbf{true}, \emptyset, \ell)$. The resulting SHA$^{\geq 0}$ is called $\mathcal{H}'$. Those self-loops allow to split runs of $\mathcal{H}'$ into portions of arbitrary small delays, because if $\mathcal{H}'$ admits a run of the form $(\ell, \nu), (t_1 + t_2, e), s$, it also admits the run $(\ell, \nu), (t_1, e'), (\ell, \nu'), (t_2, e), s$, where e' is the self loop on ℓ. Yet, this construction preserves (time-bounded) reachability:

Lemma 1 *Every run of $\mathcal{H}$ is also a run of $\mathcal{H}'$. Conversely, if $\mathcal{H}'$ admits a run ρ' with* $\mathsf{first}(\rho') = s_1$ *and* $\mathsf{last}(\rho') = s_2$*, then $\mathcal{H}$ admits a run ρ with ρ with* $\mathsf{first}(\rho) = s_1$ $\mathsf{last}(\rho) = s_2$*,* $\mathsf{duration}(\rho) = \mathsf{duration}(\rho')$ *and $|\rho| \leq |\rho'|$. Moreover, for each run ρ of $\mathcal{H}'$, there exists a run $\rho' = \rho_1 \cdot \rho_2 \cdots \rho_n$ of $\mathcal{H}'$ s.t. $n \leq \mathsf{duration}(\rho) \times \mathrm{rmax} + 1$,* $\mathsf{first}(\rho) = \mathsf{first}(\rho')$*,* $\mathsf{last}(\rho) = \mathsf{last}(\rho')$*,* $\mathsf{duration}(\rho) = \mathsf{duration}(\rho')$ *and, for all $1 \leq i \leq n$:* $\mathsf{duration}(\rho_i) < \frac{1}{\mathrm{rmax}}$.

Hybrid automaton with regions Let us describe a second construction that applies to the syntax of the hybrid automaton, and consists, roughly speaking, in encoding the integral part of the variable valuations in the locations. Let $\mathsf{Reg}(\mathrm{cmax}) = \left(\{[a,a],\ (a-1,a) \mid a \in \{1,\ldots,\mathrm{cmax}\}\} \cup \{\mathbf{0}^{=}, \mathbf{0}^{+}, (\mathrm{cmax}, +\infty)\}\right)^{X}$ be the set of *regions*, and further let $\mathsf{Reg}(\mathrm{cmax}, X)$ denote the set of all functions $r : X \mapsto \mathsf{Reg}(\mathrm{cmax})$ that assign a region to each variable. By abuse of language, we sometimes call *regions* elements of $\mathsf{Reg}(\mathrm{cmax}, X)$ too. Remark that the definition of $\mathsf{Reg}(\mathrm{cmax}, X)$ differs from the classical regions [2] by the absence of $[0,0]$ which is replaced by two

symbols: $\mathbf{0}^=$ and $\mathbf{0}^+$, and by the fact that no information is retained about the relative values of the fractional parts of the variables. The difference between $\mathbf{0}^=$ and $\mathbf{0}^+$ is elucidated later (see Lemma 3). When testing for membership to a region, $\mathbf{0}^+$ and $\mathbf{0}^=$ should be interpreted as $[0,0]$, i.e., $v \in \mathbf{0}^+$ and $v \in \mathbf{0}^=$ hold iff $v = 0$. Given a valuation ν of the set of variable X, and $r \in \mathsf{Reg}\,(\mathrm{cmax}, X)$, we let $\nu \in r$ iff $\nu(x) \in r(x)$ for all x, and, provided that $\nu > \mathbf{0}$, we denote by $[\nu]$ the (unique) element from $\mathsf{Reg}\,(\mathrm{cmax}, X)$ s.t. $\nu \in [\nu]$. Remark that for all sets of variable X and all maximal constants cmax: $|\mathsf{Reg}\,(\mathrm{cmax}, X)\,| \leq (2 \times (\mathrm{cmax}+1))^{|X|}$. Let r_1 and r_2 be two regions in $\mathsf{Reg}\,(\mathrm{cmax}, X)$, and let $v : X \mapsto \mathbb{R}$ be a function assigning a rate $v(x)$ to each variable x. Then, we say that r_2 *is a time successor of* r_1 *under* v (written $r_1 \leq^v_{\mathrm{ts}} r_2$) iff there are $\nu_1 \in r_1$, $\nu_2 \in r_2$ and a time delay t s.t. $\nu_2 = \nu_1 + t \cdot v$. Remark that, by this definition, we can have $r_1 \leq^v_{\mathrm{ts}} r_2$, $r_1(x) = \mathbf{0}^=$ and $r_2(x) = \mathbf{0}^+$ for some clock x (for instance, if $v(x) = 0$).

Let us now explain how we label the locations of $\mathcal{H}'$ by regions. We let $\mathsf{R}\,(\mathcal{H}') = (X, \mathrm{Loc}', \mathrm{Edges}', \mathrm{Rates}', \mathrm{Inv}', \mathrm{Init}')$ be the $\mathrm{SHA}^{\geq 0}$ where:

- $\mathrm{Loc}' = \mathrm{Loc} \times \mathsf{Reg}\,(\mathrm{cmax}, X)$
- for all $(\ell, r) \in \mathrm{Loc}'$: $\mathrm{Rates}'(\ell, r) = \mathrm{Rates}(\ell)$
- for all $(\ell, r) \in \mathrm{Loc}'$: $\mathrm{Inv}(\ell, r) = \mathrm{Inv}(\ell) \wedge \bigwedge_{x : r(x) = \mathbf{0}^=} x = 0$
- There is an edge $e' = \big((\ell, r), g \wedge x \in r'' \wedge g_0, Y, (\ell', r')\big)$ in Edges' iff there are an edge $e = (\ell, g, Y, \ell')$ in Edges and a region r'' s.t.: $r \leq^{\mathrm{Rates}(\ell)}_{\mathrm{ts}} r''$, for all $x \notin Y$: $r'(x) = r''(x)$, for all $x \in Y$: $r'(x) \in \{\mathbf{0}^=, \mathbf{0}^+\}$ and $g_0 = \bigwedge_{x \in X} g_0(x)$ where:

$$\forall x \in X \quad : \quad g_0(x) = \begin{cases} x = 0 & \text{if } r(x) = \mathbf{0}^= \\ x > 0 & \text{if } r(x) = \mathbf{0}^+ \\ \mathbf{true} & \text{otherwise} \end{cases} \tag{1}$$

  in this case, we say that e is the (unique) edge of $\mathcal{H}'$ *corresponding* to e'. Symmetrically, e' is the only edge corresponding to e between locations (ℓ, r) and (ℓ', r').
- $\mathrm{Init}' = \mathrm{Init} \times \{\mathbf{0}^=, \mathbf{0}^+\}^X$

It is easy to see that this construction incurs an exponential blow up in the number of locations. More precisely:

$$\begin{aligned} |\mathrm{Loc}'| &\leq |\mathrm{Loc}| \times |\mathsf{Reg}\,(\mathrm{cmax}, X)\,| \\ &= |\mathrm{Loc}| \times (2 \times (\mathrm{cmax}+1))^{|X|} \end{aligned} \tag{2}$$

Let us prove that this construction preserves reachability of states:

Lemma 2 *Let $s = (\ell, \nu)$ and $s' = (\ell', \nu')$ be two states of $\mathcal{H}'$. Then, $\mathcal{H}'$ admits a run ρ with $\mathsf{first}\,(\rho) = s$ and $\mathsf{last}\,(\rho) = s'$ iff there are r and r' s.t. $\mathsf{R}\,(\mathcal{H}')$ admits a run ρ' with $\mathsf{first}\,(\rho') = ((\ell, r), \nu)$, $\mathsf{last}\,(\rho') = ((\ell', r'), \nu')$, $\mathsf{duration}\,(\rho) = \mathsf{duration}\,(\rho')$ and $|\rho| = |\rho'|$.*

Intuitively, the regions that label locations in $\mathsf{R}\,(\mathcal{H}')$ are intended to track the region to which each variable belongs when entering the location. However, in the case where a variable x enters a location with value 0, we also need to remember whether x is

still null when crossing the next edge (for reasons that will be made clear later). This explains why we have two regions, $\mathbf{0}^=$ and $\mathbf{0}^+$, corresponding to value 0. They encode respectively the fact that the variable is null (strictly positive) when leaving the location.

Formally, we say that a run $\rho = ((\ell_0, r_0), \nu_0), (t_1, e_1), ((\ell_1, r_1), \nu_1), \ldots, (t_n, e_n), ((\ell_n, r_n), \nu_n)$ of $\mathsf{R}(\mathcal{H}')$ is *region consistent* iff (i) for all $0 \leq i \leq n$: $\nu_i \in r_i$ and (ii) for all $0 \leq i \leq n-1$, for all $x \in X$: $r_i(x) = \mathbf{0}^=$ implies $\nu_i(x) + t_{i+1} \times \mathrm{Rates}(\ell_i)(x) = 0$ and $r_i(x) = \mathbf{0}^+$ implies $\nu_i(x) + t_{i+1} \times \mathrm{Rates}(\ell_i)(x) > 0$. Then, it is easy to see that the construction of $\mathsf{R}(\mathcal{H})$ guarantees that all runs are region consistent:

Lemma 3 *All runs of* $\mathsf{R}(\mathcal{H}')$ *are region consistent.*

The contraction operator we are about to describe preserves reachability of states when applied to carefully selected run portions only. Those portions are obtained by splitting several times a complete run into sub-runs, that we categorise in 4 different *types*.

Type-0 and type-1 runs The notion of *type-0* run relies on the fact that each $\mathbf{T}$-time bounded run of $\mathcal{H}'$ (hence of $\mathsf{R}(\mathcal{H}')$) corresponds to a run ρ' that can be split into at most $\mathbf{T} \times \mathrm{rmax} + 1$ portions of duration $< \frac{1}{\mathrm{rmax}}$ (see Lemma 1). A run ρ of $\mathsf{R}(\mathcal{H}')$ is called a *type-0 run* iff there are $\rho_0, \rho_1, \ldots, \rho_k$ s.t. $\rho = \rho_0 \cdot \rho_1 \cdots \rho_k$, and for all $0 \leq i \leq k$: $\mathsf{duration}(\rho_i) < \frac{1}{\mathrm{rmax}}$. Then, each ρ_i making up the type-0 run is called a *type-1 run*.

Type-2 runs Type-1 runs are further split into type-2 runs as follows. Let $\rho = s_0, (t_1, e_1), s_1, \ldots, (t_n, e_n), s_n$ be a type-1 run of $\mathsf{R}(\mathcal{H}')$, s.t. $\mathsf{duration}(\rho) \leq \mathbf{T}$. Let S_ρ be the set of positions $0 < i \leq n$ s.t:

$$\exists x \in X : \left(\begin{array}{c} \lfloor \nu_{i-1}(x) \rfloor \neq \lfloor \nu_i(x) \rfloor \\ \text{or} \\ \lfloor \nu_{i-1}(x) \rfloor > 0 \text{ and } 0 = \langle \nu_{i-1}(x) \rangle < \langle \nu_i(x) \rangle \end{array} \right)$$

where $\lfloor x \rfloor$ and $\langle x \rangle$ denote respectively the integral and fractional parts of x. Roughly speaking, each transition (t_i, e_i) with $i \in S_\rho$ corresponds to the fact that a variable changes its region, except in the case where the variable moves from $\mathbf{0}^+$ to $(0,1)$: such transitions are not recorded in S_ρ. Since ρ is a type-1 run, its duration is at most $\frac{1}{\mathrm{rmax}}$. Hence, each variable can cross an integer value at most once along ρ, because all rates are positive. Thus, the size of S_ρ can be bounded, by a value *that does not depend on* $|\rho|$:

Lemma 4 *Let* ρ *be a type-1 run. Then* $|S_\rho| \leq 3 \times |X|$.

Proof. As the duration of a type-1 run is $< \frac{1}{\mathrm{rmax}}$, each variable can, in the worst case, follow a trajectory that will be split into 4 parts. This happens when it starts in $(b, b+1)$, moves to $[b+1, b+1]$, then $(b+1, b+2)$, then gets reset and stays in $[0, 1)$. □ Remark that if we had recorded in S_ρ the indices of the transitions from (ℓ, ν) to (ℓ', ν') s.t. $\nu(x) = 0$ and $\nu(x) \in (0,1)$ for some variable x, Lemma 4 would not hold, and we could not bound the size of S_ρ by a value independent from $|\rho|$. Indeed, in any time interval, the density of time allows a variable to be reset and to reach a strictly positive value an arbitrary number of times.

Let us now split a type-1 run ρ according to S_ρ. Assume $\rho = s_0, (t_1, e_1), s_1, \ldots, (t_n, e_n), s_n$, and that $S_\rho = \{p(1), \ldots, p(k)\}$, with $p(1) \leq p(2) \leq \cdots \leq p(k)$. Then, we let $\rho_0, \rho_1, \ldots, \rho_k$ be the runs s.t.:

$$\begin{aligned}\rho = \rho_0 \cdot s_{p(1)-1}, (t_{p(1)}, e_{p(1)}), s_{p(1)} \cdot \rho_1 \cdot s_{p(2)-1}, (t_{p(2)}, e_{p(2)}),\\ s_{p(2)}, \ldots, s_{p(k)-1}, (t_{p(k)}, e_{p(k)}), s_{p(k)} \cdot \rho_k \end{aligned} \tag{3}$$

Each ρ_i is called a *type-2 run*, and can be empty. The next lemma summarises the properties of this construction:

Lemma 5 *Let ρ be a type-1 run of $\mathsf{R}(\mathcal{H}')$ with $\mathsf{duration}(\rho) \leq \mathbf{T}$. Then, ρ is split into: $\rho_0 \cdot \rho'_1 \cdot \rho_1 \cdot \rho'_2 \cdot \rho_2 \cdots \rho'_k \cdot \rho_k$ where each ρ_i is a type-2 run; $k \leq 3 \times |X|$; $|\rho'_i| = 1$ for all $1 \leq i \leq k$; and for all $1 \leq i \leq k$: $\rho_i = (\ell_0, \nu_0), (t_1, e_1), \ldots, (t_n, e_n), (\ell_n, \nu_n)$ implies that, for all $x \in X$:*

- either *there is $a \in \mathbb{N}^{>0}$ s.t. for all $0 \leq j \leq n$: $\nu_j(x) = a$ and x is not reset along ρ_i;*
- or *for all $0 \leq j \leq n$: $\nu_j(x) \in (a, a+1)$ with $a \in \mathbb{N}^{>0}$ and x is not reset along ρ_i;*
- or *for all $0 \leq j \leq n$: $\nu_j(x) \in [0, 1)$.*

Remark that in the last case (i.e., x is in $[0, 1)$ along a type-2 run), the number of resets cannot be bounded *a priori*. For the sake of clarity, we summarise the construction so far by the following lemma:

Lemma 6 *Each type-0 run of $\mathsf{R}(\mathcal{H}')$ can be decomposed into k type-2 runs with $k \leq 3 \times (T \times \mathrm{rmax} + 1) \times |X|$.*

Type-3 runs Finally, we obtain type-3 runs by splitting type-2 runs according to the first and last resets (if they exist) of each clock. Formally, let $s_0, (t_1, e_1), s_1, \ldots, (t_n, e_n), s_n$ be a type-2 run. Assume Y_i is the reset set of e_i, for all $1 \leq i \leq n$. We let $FR_\rho = \{i \mid x \in Y_i \text{ and } \forall 0 \leq j < i : x \notin Y_j\}$ and $LR_\rho = \{i \mid x \in Y_i \text{ and } \forall i < j \leq n : x \notin Y_j\}$ be respectively the set of edge indices where a variable is reset for the first (last) in ρ. Let $R_\rho = FR_\rho \cup LR_\rho$ and assume $R_\rho = \{p(1), p(2), \ldots, p(k)\}$ with $p(1) \leq p(2) \leq \cdots \leq p(k)$. Then, we let $\rho_0, \rho_1, \ldots, \rho_k$ be the *type 3 runs* making up ρ s.t. $\rho = \rho_0 \cdot s_{p(1)-1}, (t_{p(1)}, e_{p(1)}), s_{p(1)} \cdot \rho_1 \cdots s_{p(k)-1}, (t_{p(k)}, e_{p(k)}), s_{p(k)} \cdot \rho_k$. Remark that each type-2 is split into at most $2 \times |X| + 1$ type-3 runs (i.e., $k \leq 2 \times |X|$).

Contraction operator So far, we have defined a procedure that splits any time-bounded run of $\mathsf{R}(\mathcal{H})$ into a bounded number of type-3 runs. However, the construction does not allow us to bound the length of type-3 runs, because the density of time allows to perform an arbitrary number of actions in every possible time delay. Let us now define a contraction operator that turns type-3 runs into runs with the same effect but whose lengths can be uniformly bounded (thanks to the properties of type-3 runs established below).

Intuitively, the contraction operator works as follows. Let $\rho = (\ell_0, \nu_0), (t_1, e_1), (\ell_1, \nu_1), \ldots, (t_n, e_n), (\ell_n, \nu_n)$ be a run, and let π be its timed path. We *contract* π by looking for a pair of positions $i < j$ s.t. $\ell_i = \ell_j$ (i.e., $\pi[i+1 : j]$ forms a loop) *and* s.t. all locations $\ell_{i+1}, \ell_{i+2}, \ldots, \ell_j$ occur in the prefix $\pi[1 : i]$. This situation is depicted in

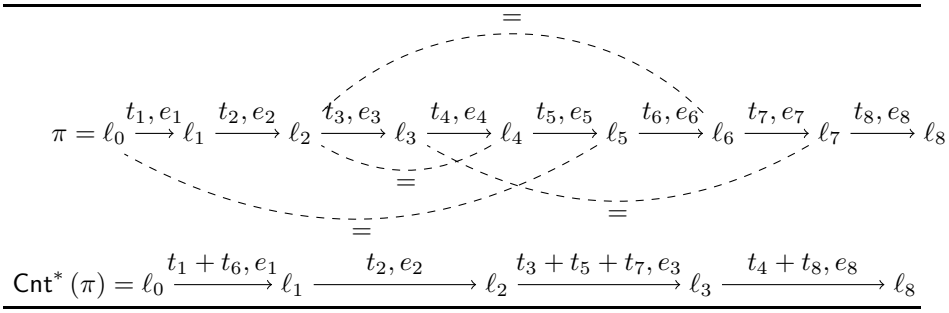

Figure 2: Illustrating the contraction operator. Here, $i = 3$, $j = 7$, $h(4) = 2$, $h(5) = 0$ and $h(6) = 2$.

Fig. 2 (top). Then, the contraction consists, roughly speaking, in *deleting* the portion $\pi[i+1:j]$ from π, and in *reporting* the delays t_{i+1},..., t_{j-1} to the other occurrences of $\ell_i, \ldots, \ell_{j-1}$ in π (that exist by hypothesis), see Fig. 2 (bottom). Clearly, in general, the resulting timed path might not yield a run as some guards could fail because of the additional delays. Yet, we prove (see Proposition 1) that, when carefully applied to type-2 runs, the contraction operator produces a *genuine run* with a bounded length, and that reaches the same state as the original run. Remark that the proof of soundness of the contraction operator relies on the fact that we have encoded the regions of the variable valuations in the locations. This information will be particularly critical when a variable is in $[0, 1)$ and reset.

The contraction operator is first defined on timed paths (we will later lift it to type-2 runs). Let us consider a timed path $\pi = (t_1, e_1), (t_2, e_2), \ldots, (t_n, e_n)$. Let $\ell_0 = \mathsf{src}(e_1)$, and, for all $1 \leq i \leq n$: $\ell_i = \mathsf{trg}(e_i)$. Assume there are $0 \leq i < j < n$ and a function $h : \{i+1, \ldots, j-1\} \mapsto \{0, \ldots, i-1\}$ s.t. (i) $\ell_i = \ell_j$ and (ii) for all $i < p < j$: $\ell_p = \ell_{h(p)}$. Then, we let $\mathsf{Cnt}(\pi) = \ell'_0, (t'_1, e'_1), \ldots, \ell'_m$ where:

1. $m = n - (j - i)$.
2. for all $0 \leq p \leq i$: $\ell'_p = \ell_p$.
3. for all $1 \leq p \leq i$: $e'_p = e_p$ and $t'_p = t_p + \sum_{k \in h^{-1}(p-1)} t_{k+1}$.
4. $e'_{i+1} = e_{j+1}$ and $t'_{i+1} = t_{i+1} + t_{j+1}$
5. for all $i + 1 < p \leq m$: $\ell'_p = \ell_{p+j-i}$ and $(t'_p, e'_p) = (t_{p+j-i}, e_{p+j-i})$.

Then, given a timed path π, we let $\mathsf{Cnt}^0(\pi) = \pi$, $\mathsf{Cnt}^i(\pi) = \mathsf{Cnt}\left(\mathsf{Cnt}^{i-1}(\pi)\right)$ for any $i \geq 1$, and $\mathsf{Cnt}^*(\pi) = \mathsf{Cnt}^n(\pi)$ where n is the least value such that $\mathsf{Cnt}^n(\pi) = \mathsf{Cnt}^{n+1}(\pi)$. Clearly, since π is finite, and since $|\mathsf{Cnt}(\pi)| < |\pi|$ or $\mathsf{Cnt}(\pi) = \pi$ for any π, $\mathsf{Cnt}^*(\pi)$ always exists. Moreover, we can always bound the length of $\mathsf{Cnt}^*(\pi)$ by a value *that does not depend on* $|\pi|$.

Lemma 7 *For all timed path* π*:* $|\mathsf{Cnt}^*(\pi)| \leq |\mathrm{Loc}|^2 + 1$.

Proof. Assume $\pi' = \mathsf{Cnt}^*(\pi) = (t_1, e_1), (t_2, e_2), \ldots, (t_n, e_n)$. Let $\ell_0 = \mathsf{src}(e_1)$, and $\ell_i = \mathsf{trg}(e_i)$ for all $1 \leq i \leq n$. Let $\mathrm{Loc}' = \{L_0, \ldots, L_m\} \subseteq \mathrm{Loc}$ be the set of locations that appear in π'. For all $L_i \in \mathrm{Loc}'$, let k_i denote the least index s.t. $\ell_{k_i} = L_i$ (i.e., the first occurrence of L_i in π'). Wlog,

we assume that $k_0 \leq k_1 \leq \cdots \leq k_m$. Then, clearly, $k_0 = 0$. Observe that each portion of the form $\pi'[k_i : k_{i+1} - 1]$ (with $0 \leq i \leq m-1$) is of length at most $|\mathrm{Loc}'|$. Otherwise, the contraction operation can be applied in this portion, as there must be two positions $k_i \leq \alpha < \beta \leq k_{i+1} - 1$ s.t. $\ell_\alpha = \ell_\beta$, and all the locations occurring along $\pi'[\alpha : \beta - 1]$ have occurred before, by definition of k_i and k_{i+1}. By the same arguments, $|\pi'[k_m : n-1]| \leq |\mathrm{Loc}'|$ (remark that, by definition of the contraction operator, the last location ℓ_n will never be considered for contraction). As $\pi'[0 : n-1]$ is made up of all those portions, and as there are $|\mathrm{Loc}'|$ portions, $|\pi'|$ is bounded by $|\mathrm{Loc}'|^2 + 1 \leq |\mathrm{Loc}|^2 + 1$. □

We can now lift the definition of the contraction operator to *runs* of type-2. Let ρ be a type-2 run and let us consider its (unique) decomposition into type-3 runs, as in (3), above. Then, we let $\mathsf{Cnt}(\rho) = \mathsf{Run}\left(\mathsf{first}(\rho), \pi_{\mathsf{Cnt}(\rho)}\right)$, where:

$$\pi_{\mathsf{Cnt}(\rho)} = \mathsf{Cnt}^*(\mathsf{TPath}(\rho_0)), (t_{p(1)}, e_{p(1)}), \mathsf{Cnt}^*(\mathsf{TPath}(\rho_1)), (t_{p(2)}, e_{p(2)}), \ldots, (t_{p(k)}, e_{p(k)}), \mathsf{Cnt}^*(\mathsf{TPath}(\rho_k))$$

By definition of Cnt^*, and by definition of Cnt on type-2 runs, it is easy to see that:

Lemma 8 *For all type-3 runs ρ:* $\mathsf{duration}(\mathsf{Cnt}^*(\mathsf{TPath}(\rho))) = \mathsf{duration}(\rho)$ *and for all variables x:* $\mathsf{Effect}(\mathsf{Cnt}^*(\mathsf{TPath}(\rho)))(x) = \mathsf{Effect}(\mathsf{TPath}(\rho))(x)$. *Similarly, for all type-2 runs ρ:* $\mathsf{duration}\left(\pi_{\mathsf{Cnt}(\rho)}\right) = \mathsf{duration}(\rho)$ *and for all variables x:* $\mathsf{Effect}\left(\pi_{\mathsf{Cnt}(\rho)}\right)(x) = \mathsf{Effect}(\mathsf{TPath}(\rho))(x)$.

Let us show that the contraction of type-2 runs is sound:

Proposition 1 *For all type-2 runs ρ,* $\mathsf{Cnt}(\rho) \neq \bot$, $\mathsf{first}(\mathsf{Cnt}(\rho)) = \mathsf{first}(\rho)$ *and* $\mathsf{last}(\mathsf{Cnt}(\rho)) = \mathsf{last}(\rho)$.

Proof. Let $\rho = (\ell_0, \nu_0), (t_1, e_1), \ldots (t_n, e_n), (\ell_n, \nu_n)$. Let π denote $\mathsf{TPath}(\rho)$, and let $\pi_{\mathsf{Cnt}(\rho)} = (t'_1, e'_1), \ldots, (t'_k, e'_k)$. For all $1 \leq i \leq k$, let $\ell'_i = \mathsf{dest}(e'_i)$ by ℓ'_i; and let $\ell'_0 = \mathsf{src}(e_1) = \ell_0$.

First, observe that, by definition of the contraction operator, $\ell_n = \ell'_k$. Let us show that $\mathsf{Cnt}(\rho) \neq \bot$. Assume that, for all $0 \leq i \leq k$: $\ell'_i = (\overline{\ell}_i, r_i)$ and let ν'_i be the valuation s.t. for all x:

$$\nu'_i(x) = \begin{cases} \nu'_{i-1}(x) + \mathrm{Rates}(\ell'_{i-1})(x) \times t'_i & \text{If } e'_i \text{ does not reset } x \\ 0 & \text{Otherwise} \end{cases}$$

Finally, let $\nu'_0 = \nu_0$. Remark that $\nu'_0(x) \leq \nu'_1(x) \leq \cdots \leq \nu'_k(x)$ because rates are non-negative. Clearly, to show that $\rho' \neq \bot$, it is sufficient to show, for all i, that $\nu'_i \models g'_i$ (where g'_i is the guard of e'_i); and that both ν_i and ν'_i satisfy[1] $\mathrm{Inv}(\ell_i)$. For the sake of clarity, we prove that all the guards are satisfied; the arguments can be easily adapted to show that the invariants are satisfied too.

First, consider a variable x that is not reset along π (hence along $\pi_{\mathsf{Cnt}(\rho)}$) and s.t. $\nu_0(x) = \nu'_0(x) > 0$. By definition of type-2 runs, and since x is not reset and not null initially, $\nu_0(x)$, $\nu_1(x)$,..., $\nu_n(x)$ all belong to the same interval I which is either $(a-1, a)$ or $[a, a]$ for some $a \geq 1$. Thus, in particular, $\nu_0(x) = \nu'_0(x) \in I$. Moreover, since $\mathsf{Effect}\left(\pi_{\mathsf{Cnt}(\rho)}\right)(x) = \mathsf{Effect}(\pi)(x)$ (Lemma 8), we have $\nu'_k(x) = \nu_n(x) \in I$ too. Hence, since $\nu'_0(x) \leq \nu'_1(x) \leq \cdots \leq \nu'_k(x)$, we conclude that $\nu'_i(x) \in I$ for all

[1] Remember that we consider $\mathrm{RHA}^{\geq 0}$, so the invariants are convex.

$0 \leq i \leq k$. Since all the $\nu_i(x)$ are also in I, since ρ is a genuine run, and since all edges e'_i in π' are also present in π, we conclude that $\nu \in I$ *implies* $\nu \models g'_i$, for all valuation ν and all guards g'_i of some edge e'_i in π. Hence, $\nu'_i \models g'_i$ for all i.

Thus, we can, from now on, safely ignore all variables x that are not reset along π (hence along $\pi_{\mathsf{Cnt}(\rho)}$) and s.t. $\nu_0(x) = \nu'_0(x) > 0$, and focus on variables x that are either reset along π or s.t. $\nu_0(x) = \nu'_0(x) = 0$. By definition of type-2 runs, in both cases, these variables take values in $[0, 1)$ in each state along ρ. Hence, since ρ is region consistent (Lemma 3), all locations in ρ are of the form (ℓ, r) with $r(x) \in \{\mathbf{0}^=, \mathbf{0}^+, (0, 1)\}$, and so are all locations in $\pi_{\mathsf{Cnt}(\rho)}$: for all $0 \leq i \leq k$: $\overline{\ell}_i \in \{\mathbf{0}^=, \mathbf{0}^+, (0, 1)\}$. Let us denote, by ρ'_j the value $\mathsf{Run}\left(\mathsf{first}\left(\rho\right), \pi_{\mathsf{Cnt}(\rho)}[1:j]\right)$ for all $m \geq 1$. We further denote by ρ'_0 the run of null length (ℓ'_0, ν'_0). Let us show that, for all $0 \leq j \leq k$, $\rho'_j \neq \bot$, by induction on j.

The base case is $j = 0$ and is trivial since $(\ell_0, \nu_0) = (\ell'_0, \nu'_0)$. For the inductive case, we assume that $\rho'_{m-1} \neq \bot$ (for some $m \geq 1$) and ends in $((\overline{\ell}, r), \nu)$, and we show that we can extend it by firing (t'_m, e'_m) (i.e., that $\rho'_m \neq \bot$). Observe that, by definition of Cnt, the edge e'_m occurs in $\pi_{\mathsf{Cnt}(\rho)}$ because it was already present in π (say, at position α, hence $e_\alpha = e'_m$ and $(\overline{\ell}, r) = \ell_{\alpha-1}$). Moreover, still by definition of Cnt, the delay t'_m is equal to $t_\alpha + \sum_{i=1}^{\beta} t_{p(i)}$, where for all $1 \leq i \leq \beta$: $\mathsf{src}\left(e_{p(i)}\right) = (\overline{\ell}, r)$. We consider three cases:

1. Either $r(x) = \mathbf{0}^=$. In this case, since ρ and ρ'_{m-1} are region consistent (Lemma 3), and since the region $r(x)$ is $\mathbf{0}^=$ (and not $\mathbf{0}^+$), we know that $\nu_{\alpha-1}(x) = 0$ (x is null when entering $(\overline{\ell}, r)$ at position $\alpha-1$ in ρ), that $\nu(x) = 0$ (x is null at the end of ρ'_{m-1}), and that $\nu_{\alpha-1}(x) + t_\alpha \times \mathrm{Rates}(\overline{\ell}, r)(x) = t_\alpha \times \mathrm{Rates}(\overline{\ell}, r)(x) = 0$ (x is null when leaving $(\overline{\ell}, r)$ at position $\alpha - 1$ in ρ). This means, in particular that it is sufficient for x to be null to satisfy the guard of $e'_m = e_\alpha$. Moreover, for all $1 \leq i \leq \beta$: $\nu_{p(i-1)}(x) = 0 = t_{p(i)} \times \mathrm{Rates}(\overline{\ell}, r)(x)$ (x is null when entering and leaving the locations at all positions $p(i)$ that have yielded the contraction in ρ). Thus, the value that x takes after letting t'_m t.u. elapse the last state or ρ'_{m-1} is $\nu'(x) = \nu(x) + t'_m \times \mathrm{Rates}(\overline{\ell}, r)(x) = (t_\alpha + \sum_{i=1}^{\beta} t_{p(i)}) \times \mathrm{Rates}(\overline{\ell}, r)(x) = 0$. Hence $\nu'(x)$ satisfies the guard of e'_m, and we can extend ρ'_{m-1} by (t'_m, e'_m).

2. Or $r(x) = \mathbf{0}^+$. In this case, we know that $\nu_{\alpha-1}(x) = \nu(x) = 0$, that $t_\alpha \times \mathrm{Rates}(\overline{\ell}, r)(x) > 0$, and that for all $1 \leq i \leq \beta$: $\nu_{p(i-1)}(x) = 0$ and $t_{p(i)} \times \mathrm{Rates}(\overline{\ell}, r)(x) > 0$. Moreover, since $\mathsf{duration}\left(\rho\right) < \frac{1}{\mathrm{rmax}}$, we can precise this information and conclude that $t_\alpha \times \mathrm{Rates}(\overline{\ell}, r)(x) \in (0, 1)$ and that for all $1 \leq i \leq \beta$: $t_{p(i)} \times \mathrm{Rates}(\overline{\ell}, r)(x) \in (0, 1)$. Thus, it is sufficient, to satisfy the constraints on x in the guard of e'_m, that $x \in (0, 1)$. Let us show that $\nu'(x) = (t_\alpha + \sum_{i=1}^{\beta} t_{p(i)}) \times \mathrm{Rates}(\overline{\ell}, r)(x)$ is in $(0, 1)$ too. We have $\nu'(x) > 0$ because $t_\alpha \times \mathrm{Rates}(\overline{\ell}, r)(x) > 0$, as shown above. Moreover, $\nu'(x) < 1$ because $t_\alpha + \sum_{i=1}^{\beta} t_{p(i)} \leq \mathsf{duration}\left(\rho\right) < \frac{1}{\mathrm{rmax}}$, by def. of type-2 runs. Thus, $\nu'(x)$ satisfies the guard of e'_m and we can extend ρ'_{m-1} by (t'_m, e'_m).

3. Or $r(x) = (0, 1)$. In this case, we can rely on the same arguments as above to show that $\nu'(x) > 0$, and that $\nu'(x)$ should be in $(0, 1)$ to satisfy the guard of e'_m. The difference with the previous case is that $\nu(x) \neq 0$ here, and we have to make sure that the additional delay accumulated on $(\overline{\ell}, r)$ by the contraction operator does not increase x above 1. This property holds because of the split of type-2 runs in type-3 runs, according to the first reset of each variable. More precisely, we consider two cases. Either $\ell_{\alpha-1}$ occurs, in ρ in a type-3 run that

takes place *after* the first reset of x. In this case, $\nu'(x) = \nu(x)(t_\alpha + \sum_{i=1}^{\beta} t_{p(i)}) \times \mathrm{Rates}(\overline{\ell}, r)(x) < 1$, because all the $t_{p(i)}$ also occur $\pi[\alpha : n]$ (i.e., after the first reset of x), and $\mathsf{duration}(\pi[\alpha : n]) < \frac{1}{\mathrm{rmax}}$. Or $\ell_{\alpha-1}$ occurs, in ρ in a type-3 run that takes place *before* the first reset of x. In this case, $\nu'(x) = \nu(x)(t_\alpha + \sum_{i=1}^{\beta} t_{p(i)}) \times \mathrm{Rates}(\overline{\ell}, r)(x) \geq 1$ implies that, in ρ: $\nu_{p(i)} \geq 1$, which contradicts the definition of type-2 runs. Hence, $\nu'(x) \in (0, 1)$ and we can extend ρ'_{m-1} by (t'_m, e'_m).

Let us conclude the proof by showing that $\nu'_k = \nu_n$. We consider three cases. First, x is a variable that is not reset along ρ. Since $\mathsf{Effect}(\mathsf{Cnt}^*(\pi))(x) = \mathsf{Effect}(\pi)(x)$ (Lemma 8), and since $\nu_0 = \nu'_0$, we conclude that $\nu'_k(x) = \nu_n(x)$. Second, x is a variable that is reset along ρ. Since the duration of a type-2 is at most $\frac{1}{\mathrm{rmax}}$, $\nu_n(x) \in [0, 1)$. Thus, we consider two further cases. *Either* $\nu_n(x) = 0$. Since ρ is region-consistent (Lemma 3), ℓ_n is of the form $(\overline{\ell}, r)$ with $r(x) \in \{\mathbf{0}^+, \mathbf{0}^=\}$. However, $\ell_n = \ell'_k$, and since $\mathsf{Cnt}(\rho)$ is a run and hence region-consistent, we conclude that $\nu'_k(x) = 0$ too. *Or* $\nu_n(x) \in (0, 1)$. In this case, it is easy to observe that $\nu_n(x)$ depends only on the portion of ρ that occurs after the last reset of x, i.e., $\nu_n(x) = \mathsf{Effect}(\pi[i + 1 : n])(x)$, where i is the largest position in ρ s.t. e_i resets x. By definition of the contraction operator on type 2 runs, e_i occurs at some position α of $\pi_{\mathsf{Cnt}(\rho)}$, i.e. $e_i = e'_\alpha$ and e'_α is the last edge of $\pi_{\mathsf{Cnt}(\rho)}$ to reset x. Thus, $\nu'_k(x) = \mathsf{Effect}\left(\pi_{\mathsf{Cnt}(\rho)}[\alpha + 1 : k]\right)(x)$. However, by Lemma 8, and by definition of the contraction of type 2 runs: $\mathsf{Effect}\left(\pi_{\mathsf{Cnt}(\rho)}[\alpha + 1 : k]\right)(x) = \mathsf{Effect}(\pi[i + 1 : n])(x)$. Hence, $\nu_n(x) = \nu'_k(x)$. □

Then, observe that, by the above definition, and by Lemma 7, we can bound the length of $\mathsf{Cnt}(\rho)$ for type-2 runs ρ:

Lemma 9 *For all type-2 runs:* $|\mathsf{Cnt}(\rho)| \leq 8 \times |\mathrm{Loc}|^2 \times |X|$.

Proof. By definition of type-2 runs, and by Lemma 7, $|\mathsf{Cnt}(\rho)|$ is at most $(2 \times |X| + 1) \times (|\mathrm{Loc}|^2 + 1) + 2 \times |X| = 2 \times (|X| + 1) \times (|\mathrm{Loc}|^2 + 1)$. However, wlog, $|\mathrm{Loc}| \geq 1$ and $|X| \geq 1$. Hence $|X| + 1 \leq 2 \times |X|$, $|\mathrm{Loc}|^2 + 1 \leq 2 \times |\mathrm{Loc}|^2$. Hence the lemma. □

We can now explain more intuitively why we need two different regions ($\mathbf{0}^=$ and $\mathbf{0}^+$) for variables that are null, and cannot use $[0, 0]$ instead. Consider the example given in Fig. 3. Run ρ_1 depicts a run of an automaton with a single variable x, where we have used only region $[0, 0]$ in the construction of $\mathsf{R}(\mathcal{H}')$. In this run, x is null in all four states. The two locations of $\mathsf{R}(\mathcal{H}')$ that are met are $(\ell_1, [0, 0])$ and $(\ell_2, [0, 0])$ (and in both locations, the rate of x is strictly positive). Hence, the contraction operator 'merges' the two occurrences of both locations, an produces ρ_2. However, ρ_2 fails to satisfy Proposition 1, as x is null in the last state of ρ_1 but not in the last state of ρ_2. This comes from the fact that region $[0, 0]$ does not allow to distinguish between locations that are left with a strictly positive delay or a null delay. With our definition of $\mathsf{R}(\mathcal{H}')$, however, the first state of the run is $\big((\ell_1, \mathbf{0}^=), 0\big)$, as x is null when crossing the first edge, but the *third* state is $\big((\ell_1, \mathbf{0}^+), 0\big)$, as x is *not null* when crossing the last edge, which avoids the problem illustrated in Fig. 3.

Thus, summing up the properties of the contraction operator, and the splitting procedure we obtain, as a corollary of Proposition 1 and Lemma 6:

Corollary 1 *Let s and s' be two states of $\mathsf{R}(\mathcal{H}')$. Then, $\mathsf{R}(\mathcal{H}')$ admits a $\mathbf{T}$-time-bounded type-0 run ρ with $\mathsf{first}(\rho) = s$ and $\mathsf{last}(\rho) = s'$ iff it admits a $\mathbf{T}$-time bounded type-0 run ρ' with $\mathsf{first}(\rho') = s$, $\mathsf{last}(\rho') = s'$ and $|\rho'| \leq 48 \times \mathbf{T} \times \mathrm{rmax} \times |\mathrm{Loc}'|^2 \times$*

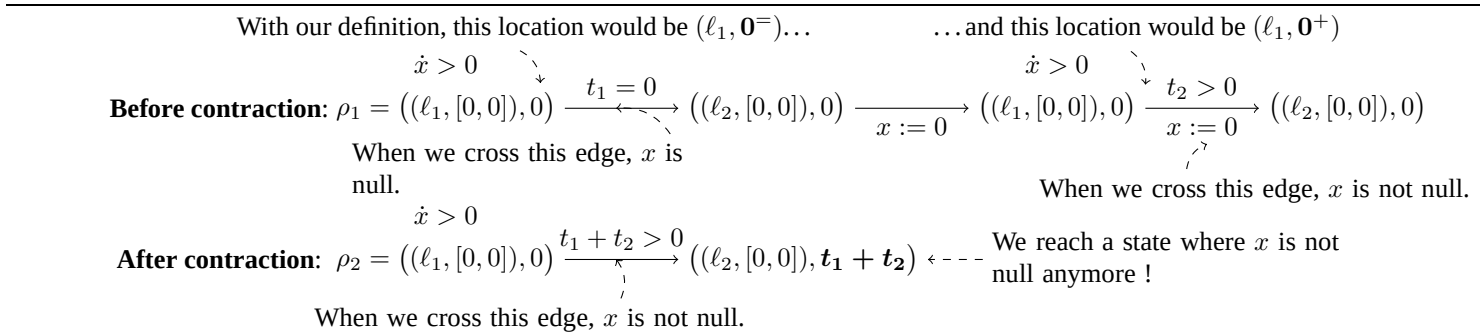

Figure 3: An example that shows why the contraction operator fails if we use $[0, 0]$ to characterise the variables that are null.

$|X|^2$, *where* X, Loc' *and* rmax *are resp. the set of variable, set of locations and maximal rate of* $\mathsf{R}(\mathcal{H}')$.

Finally, for all SHA$^{\geq 0}$ $\mathcal{H} = (X, \mathrm{Loc}, \mathrm{Edges}, \mathrm{Rates}, \mathrm{Inv}, \mathrm{Init})$ and all time bound $\mathbf{T} \in \mathbb{N}$, we let:

$$F(\mathcal{H}, \mathbf{T}) = \\ 24 \times (\mathbf{T} \times \mathrm{rmax} + 1) \times |X|^2 \times |\mathrm{Loc}|^2 \times (2 \times \mathrm{cmax} + 1)^{2 \times |X|}$$

This value $F(\mathcal{H}, \mathbf{T})$ is actually a bound on the length of the runs we need to consider to decide $\mathbf{T}$-time-bounded reachability:

Theorem 1 *Let* $\mathcal{H}$ *be a SHA*$^{\geq 0}$, $\mathbf{T}$ *be a time bound and let* s_1 *and* s_2 *be two states of* $\mathcal{H}$. *Then* $\mathcal{H}$ *admits a* $\mathbf{T}$*-time-bounded run* ρ *with* $\mathsf{first}(\rho) = s_1$ *and* $\mathsf{last}(\rho) = s_2$ *iff it admits a* $\mathbf{T}$*-time-bounded run* ρ' *with* $|\rho'| \leq F(\mathcal{H}, \mathbf{T})$, $\mathsf{first}(\rho') = s_1$ *and* $\mathsf{last}(\rho') = s_2$.

Proof. The *if* direction is trivial, let us prove the *only if*, by proving the contraposition, i.e., that if $\mathcal{H}$ admits no $\mathbf{T}$-time-bounded run of length at most $F(\mathcal{H}, \mathbf{T})$ from s_1 to s_2, then it admits no $\mathbf{T}$-time-bounded run from s_1 to s_2. By Lemma 1, if $\mathcal{H}$ admits no $\mathbf{T}$-time bounded run of length at most $F(\mathcal{H}, \mathbf{T})$ from $s_1 = (\ell_1, \nu_1)$ to $s_2 = (\ell_2, \nu_2)$, then, $\mathcal{H}'$ admits no $\mathbf{T}$-time-bounded run of length at most $F(\mathcal{H}, \mathbf{T})$ from s_1 to s_2. Then, by Lemma 2, then, for all pair of regions r_1, r_2: $\mathsf{R}(\mathcal{H})$ admits no type-0 $\mathbf{T}$-time-bounded run of length at most $F(\mathcal{H}, \mathbf{T})$ from $s_1' = ((\ell_1, r_1), \nu_1)$ to $s_2' = ((\ell_2, r_2), \nu_2)$. By Corollary 1, and by (2), $\mathsf{R}(\mathcal{H})$ admits no type-0 $\mathbf{T}$-time-bounded run from s_1' to s_2', regardless of the length of the run. Hence, by Lemma 2, $\mathcal{H}'$ admits no $\mathbf{T}$-time-bounded run ρ from s_1 to s_2, and neither does $\mathcal{H}$, by Lemma 1 again. □

Remark that $F(\mathcal{H}, \mathbf{T}) = \mathcal{O}\left(\mathbf{T} \times 2^{|\mathcal{H}|}\right)$, where $|\mathcal{H}|$ is the number of bits necessary to encode $\mathcal{H}$, using standard encoding techniques and binary encoding for the constants. Hence, Theorem 1 tells us that, to decide $\mathbf{T}$-time-bounded reachability, we only need to consider runs whose length is singly exponential in the size of the instance $(\mathcal{H}, \mathbf{T})$.

Let us now briefly explain how we can adapt the previous construction to cope with non-singular rates. Let us first notice that given $\mathcal{H}$ a RHA$^{\geq 0}$, the construction of $\mathsf{R}(\mathcal{H}')$ still makes perfect sense and still satisfies Lemma 3. Then, we need to adapt the definition of timed path. A timed path is now of the form $(t_1, R_1, e_1) \cdots (t_n, R_n, e_n)$, where each $R_i : X \mapsto \mathbb{R}$ gives the actual rate that was chosen for each variable at the i-th continuous step. It is then straightforward to extend the definitions of Cnt, Effect and $\mathsf{Contraction}$ to take those rates into account and still keep the properties needed to prove Theorem 1. More precisely, the contraction of a set of transitions $(t_1, R_1, e_1), \ldots, (t_n, R_n, e_n)$ yields a transition (t, R, e) with $t = \sum_{i=1}^{n} t_i$ and, $R = \frac{\sum_{i=1}^{n} t_i \times R_i}{t}$. Note that we need to rely on the convexity of the invariants and rates in an RHA to ensure that this construction is correct. Thus, we can extend Theorem 1 to the case of RHA with positive rates (RHA$^{\geq 0}$):

Corollary 2 *Let* $\mathcal{H}$ *be a RHA*$^{\geq 0}$, $\mathbf{T}$ *be a time bound and let* s_1 *and* s_2 *be two states of* $\mathcal{H}$. *Then* $\mathcal{H}$ *admits a* $\mathbf{T}$*-time-bounded run* ρ *with* $\mathsf{first}(\rho) = s_1$ *and* $\mathsf{last}(\rho) = s_2$ *iff it admits a* $\mathbf{T}$*-time-bounded run* ρ' *with* $|\rho'| \leq F(\mathcal{H}, \mathbf{T})$, $\mathsf{first}(\rho') = s_1$ *and* $\mathsf{last}(\rho') = s_2$.

4 Time-bounded reachability is NEXPTIME-c

In this section, we establish the exact computational complexity of the time-bounded reachability problem for RHA$^{\geq 0}$.

Theorem 2 *The time-bounded reachability problem for RHA$^{\geq 0}$ is complete for* NEXPTIME.

To prove this theorem, we exhibit an NEXPTIME algorithm for time-bounded reachability and we reduce this problem from the reachability problem of exponential time Turing machine.

An NEXPTIME **algorithm** Recall that an instance of the time-bounded reachability problem is of the form $(\mathcal{H}, \ell, \mathbf{T})$, where $\mathcal{H}$ is an RHA$^{\geq 0}$, ℓ is a location, and $\mathbf{T}$ is a time bound (expressed in binary). We establish membership to NEXPTIME by giving a non-deterministic algorithm that runs in exponential time in the size of $(\mathcal{H}, \ell, \mathbf{T})$ in the worst case. The algorithm first *guesses* a sequence of edges $\mathcal{E} = e_0 e_1 \dots e_n$ of $\mathcal{H}$ s.t. $n + 1 \leq F(\mathcal{H}, \mathbf{T})$ and $\mathsf{trg}\,(e_n) = \ell$. Then the algorithm builds from $\mathcal{E}$ a linear constraint $\Phi(\mathcal{E})$, that expresses all the properties that must be satisfied by a run that follows the sequence of edges in $\mathcal{E}$ (see [13] for a detailed explanation on how to build such a constraint). This constraint uses $n + 1$ copies of the variables in X and $n + 1$ variables t_i to model the time elapsing between two consecutive edges, and imposes that the valuations of the variables along the run are consistent with the rates, guards and resets of $\mathcal{H}$. Finally, the algorithm checks whether $\Phi(\mathcal{E})$ is satisfiable and returns 'yes' iff it is the case.

The number of computation steps necessary to build $\Phi(\mathcal{E})$ is, in the worst case, exponential in $|\mathcal{H}|$ and $\mathbf{T}$. Moreover, checking satisfiability of $\Phi(\mathcal{E})$ can be done in polynomial time (in the size of the constraint) using classical algorithms to solve linear programs. Clearly this procedure is an NEXPTIME algorithm for solving the time-bounded reachability problem for RHA$^{\geq 0}$.

NEXPTIME**-hardness** To establish the NEXPTIME-hardness, we show how to reduce the membership problem for non-deterministic exponential time Turing machines to time-bounded reachability for SHA$^{\geq 0}$.

A non-deterministic exponential time Turing machine (NExpTM) is a tuple $M = (Q, \Sigma, \Gamma, \sharp, q_0, \delta, F, \xi)$ where Q is the (nonempty and finite) set of control states, Σ is the (finite) input alphabet, $\Gamma \supseteq \Sigma$ is the (finite) alphabet of the tape, $\sharp \in \Gamma$ is the blank symbol, $q_0 \in Q$ is the initial control state, $\delta \subseteq Q \times \Gamma \times \Gamma \times \{L, R\} \times Q$ is the transition relation, $F \subseteq Q$ is the set of accepting states, and $\xi = \mathcal{O}\left(2^{p(n)}\right)$ (for some polynomial p), is an exponential function that bounds the execution time of the machine on input w by $\xi(|w|)$.

As usual, a state of M is a triple (q, w_1, w_2) where $q \in Q$ is a control state, $w_1 \in \Gamma^*$ a word that represents the content of the tape on the left of the reading head (this word is empty when the head is on the leftmost cell of the tape), and $w_2 \in \Gamma^*$ is the content of the tape on the right of the reading head excluding the sequence of blank symbols ($\sharp$) at the end of the tape, (in particular the first letter in w_2 is the content of the cell below the reading head).

A transition of the Turing machine is a tuple of the form $(q_1, \gamma_1, \gamma_2, D, q_2)$ with the usual semantics: it is enabled iff the current control state is q_1, the content of the cell below the reading head is equal to γ_1, and the head should not

be above the left most cell when $D = L$. The execution of the transition modifies the content of the tape below the reading head to γ_2, moves the reading head one cell to the right if $D = R$, or one cell to the left if $D = L$, and finally, changes the control state to q_2. We write $(q, w_1, w_2) \triangleright (q', w_1', w_2')$ if there exists a transition in δ from state (q, w_1, w_2) to state (q', w_1', w_2').

An (exponentially bounded) execution of M on input w is a finite sequence of states $c_0 c_1 \dots c_n$ such that: (i) $n \leq \xi(|w|)$ (the execution is exponentially bounded); (ii) $c_0 = (q_0, \epsilon, w \cdot \sharp^{\xi(|w|)-|w|})$, (the initial control state is q_0 and the tape contains w followed by the adequate number of blank symbols); and (iii) for all $0 \leq i < n$ $c_i \triangleright c_{i+1}$, (the transition relation is enforced). The execution is *accepting* iff $c_n = (q, w_1, w_2)$ with $q \in F$. W.l.o.g., we make the assumption that $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \sharp\}$, and transitions only write letters in Σ. This ensures that in all reachable states (q, w_1, w_2) we have that $w_1, w_2 \in \{0, 1\}^*$.

The *membership problem* for an NExpTM M and a word w asks whether there exists an accepting execution of the Turing Machine M that uses at most $\xi(|w|)$ steps.

Let us show how we can encode all executions of M into the executions of an SHA$^{\geq 0}$ $\mathcal{H}_M$. We encode the words w_1 and w_2 as pairs of rational values (l_1, c_1) and (l_2, c_2) where $l_i = \frac{1}{2^{|w_i|}}$ encodes the length of the word w_i by a rational number in $[0, 1]$, and c_i encodes w_i as follows. Assume $w_1 = \sigma_0 \sigma_1 \dots \sigma_n$. Then, we let $c_1 = \mathsf{Val}^{\leftarrow}(w_1) = \sigma_n \cdot \frac{1}{2} + \sigma_{n-1} \cdot \frac{1}{4} + \dots + \sigma_0 \cdot \frac{1}{2^{n+1}}$. Intuitively, c_1 is the value which is represented in binary by $0.\sigma_n \sigma_{n-1} \cdots \sigma_0$, i.e., w_1 is the binary encoding of the fractional part of c_1 where the most significant bit in the rightmost position. For instance, if $w_1 = 001010$ then $\mathsf{Val}^{\leftarrow}(w_1) = 0 \cdot \frac{1}{2} + 1 \cdot \frac{1}{4} + 0 \cdot \frac{1}{8} + 1 \cdot \frac{1}{16} + 0 \cdot \frac{1}{32} + 0 \cdot \frac{1}{64} = 0.3125$, and so w_1 is encoded as the pair $(\frac{1}{64}, 0.3125)$. Remark that we need to remember the actual length of the word w_1 because the function $\mathsf{Val}^{\leftarrow}(\cdot)$ ignores the leading 0's (for instance, $\mathsf{Val}^{\leftarrow}(001010) = \mathsf{Val}^{\leftarrow}(1010)$). Symmetrically, if $w_2 = \sigma_0 \sigma_1 \dots \sigma_n$, we let $c_2 = \mathsf{Val}^{\rightarrow}(w_2) = \sigma_0 \cdot \frac{1}{2} + \sigma_1 \cdot \frac{1}{4} + \dots + \sigma_n \cdot \frac{1}{2^{n+1}}$ (i.e., σ_0 is now the most significant bit). Then, a state (q, w_1, w_2) of the TM is encoded as follows: the control state q is remembered in the locations of the automaton, and the words w_1, w_2 are stored, using the encoding described above using four variables to store the values (l_1, c_1) and (l_2, c_2).

With this encoding in mind, let us list the operations that we must be able to perform to simulate the transitions of the TM. Assume $w_1 = w_0^1 w_2^1 \cdots w_n^1$ and $w_2 = w_0^2 w_2^2 \cdots w_k^2$. We first describe the operations that are necessary to *read the tape*:

- *Read the letter under the head.* Following our encoding, we need to test the value of the bit w_0^2. Clearly, $w_0^2 = 1$ iff $l_2 \leq 1/2$, and $c_2 \geq \frac{1}{2}$; $w_0^2 = 0$ iff $l_2 \leq 1/2$, and $c_2 < \frac{1}{2}$ and $w_0^2 = \sharp$ iff $l_2 = 1$ (which corresponds to $w_2 = \epsilon$).
- *Test whether the head is in the leftmost cell of the tape.* This happens if and only if $w_1 = \epsilon$, and so if and only if $l_1 = 1$.
- *Read the letter at the left of the head* (assuming that $w_1 \neq \epsilon$). Following our encoding, this amounts to testing the value of the bit w_n^1. Clearly, $w_n^1 = 1$ iff $c_1 \geq \frac{1}{2}$ and $w_n^1 = 0$ iff $c_1 < \frac{1}{2}$.

Then, let us describe the operations that are necessary to update the values on the tape. Clearly, they can be carried out by appending and removing 0 or 1's to the right of w_1 or to the left of w_2. Let us describe how we update c_1 and l_1 to simulate these operations on w_1 (the operations on w_2 can be deduced from this description). We denote by c_1' (resp. l_1') the value of c_1 (l_1) after the simulation of the TM transition.

- To append a 1 to the right of w_1, we let $l'_1 = \frac{1}{2} \times l_1$. We let $c'_1 = \frac{1}{2}$ if $l_1 = 1$ (i.e. w_1 was empty) and $c'_1 = \frac{1}{2} \times c_1 + \frac{1}{2}$.
- To append a 0 to the right of w_1, we let $l'_1 = \frac{1}{2} \times l_1$ and $c'_1 = \frac{1}{2} \times c_1$.
- To delete a 0 from the rightmost position of w_1, we $l'_1 = 2 \times l_1$, $c'_1 = 2 \times c_1$.
- To delete a 1 from the rightmost position of w_1, $l'_1 = 2 \times l_1$, and $c'_1 = (c_1 - \frac{1}{2}) \times 2$.

In addition, remark that we can flip the leftmost bit of w_2 by adding or subtracting $1/2$ from c_2 (this is necessary when updating the value under the head).

Thus, the operations that we need to be able to perform on c_1, l_1, c_2 and l_2 are: to multiply by 2, divide by 2, increase by $\frac{1}{2}$ and decrease by $\frac{1}{2}$, while keeping untouched the value of all the other variables. Fig. 4 exhibits four gadgets to perform these operations. Remark that these gadgets can be constructed in polynomial time, execute in exactly 1 time unit time and that all the rates in the gadgets are singular.

We claim that all transitions of M can be simulated by combining the gadgets in Fig. 4 and the tests described above. As an example, consider the transition: $(q_1, 1, 0, L, q_2)$. It is simulated in our encoding as follows. First, we check that the reading head is not at the leftmost position of the tape by checking that $l_1 < 1$. Second, we check that the value below the reading head is equal to 1 by testing that $l_2 < 1$ and $c_2 \geq \frac{1}{2}$. Third, we change the value below the reading head from 1 to 0 by subtracting $\frac{1}{2}$ from c_2 using an instance of gadget (ii) in Fig. 4. And finally, we move the head one cell to the left. This is performed by testing the bit on the left of the head, deleting it from w_1 and appending it to the left of w_2, by the operations described above. All other transitions can be simulated similarly. Remark that, to simulate one TM transition, we need to perform several tests (that carry out in 0 t.u.) and to: (i) update the bit under the reading head, which takes 1 t.u. with our gadgets; (ii) remove one bit from the right of w_1 (resp. left of w_2), which takes at most 3 t.u. and (iii) append this bit to the left of w_2 (right of w_1), which takes at most 3 t.u. We conclude that each TM transition can be simulate in at most 7 time units.

Thus M has an accepting execution on word w (of length at most $\xi(|w|)$ iff $\mathcal{H}_M$ has an execution of duration at most $\mathbf{T} = 7 \cdot \xi(|w|)$ that reaches a location encoding an accepting control state of M. This sets the reduction.

5 Computing fixpoints

In this section, we show that Corollary 2 implies that we can effectively compute the set of states that are reachable within $\mathbf{T}$ time units in an RHA with non-negative rates (using formulas of the first-order logic $(\mathbb{R}, 0, 1, +, \leq)$ over the reals as a symbolic representation for such sets). We demonstrate, by means of two examples, that this information can be useful in practice, in particular when the regular (not time-bounded) fixed points do not terminate.

Post and Pre Let s be state of an RHA with set of edges Edges. Then, we let $\mathsf{Post}(s)\{s' \mid \exists e \in \mathrm{Edges}, t \in \mathbb{R}^+ : s \xrightarrow{t,e} s'\}$ and $\mathsf{Pre}(s)\{s' \mid \exists e \in \mathrm{Edges}, t \in \mathbb{R}^+ : s' \xrightarrow{t,e} s\}$. We further let $\mathsf{Reach}^{\leq \mathbf{T}}(s) = \{s' \mid \exists \pi : s \xrightarrow{\pi} s' \wedge \mathsf{duration}(\pi) \leq \mathbf{T}\}$, and $\mathsf{coReach}^{\leq \mathbf{T}}(s) = \{s' \mid \exists \pi : s' \xrightarrow{\pi} s \wedge \mathsf{duration}(\pi) \leq \mathbf{T}\}$ be respectively the set of states that are reachable from s (that can reach s) within $\mathbf{T}$ time units. We extend all those operators to sets of states in the obvious way.

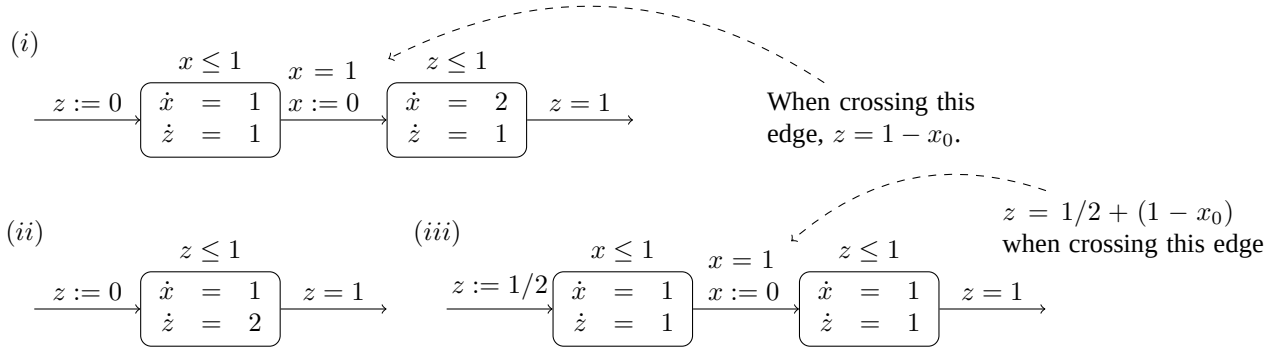

Figure 4: Gadgets (i) for multiplication by 2, (ii) adding $\frac{1}{2}$ and (iii) subtracting $\frac{1}{2}$. The rates of the $y \notin \{x, z\}$ is 0. Gadget (i) can be modified to divide by 2, by swapping the rates of x and z in the second location. x_0 is the value of x when entering the gadget.

Region algebra To symbolically manipulate sets of states, it is well known that we can use formulas of $(\mathbb{R}, 0, 1, +, \leq)$, i.e. the first-order logic of the reals, with the constants 0 and 1, the usual order $\leq$ and addition $+$ (see [11] for the details). Recall that the satisfiability problem for that logic is decidable [4] and that it admits effective quantifier elimination. Further remark that, in a RHA, all guards can be characterised by a formula of $(\mathbb{R}, 0, 1, +, \leq)$ ranging over X. Let Ψ be a formula of $(\mathbb{R}, 0, 1, +, \leq)$, and let ν be a valuation of the free variables of Ψ. Then, we write $\nu \models \Psi$ iff ν satisfies Ψ, and we let $[\![\Psi]\!]$ be the set off all valuations ν s.t. $\nu \models \Psi$. To emphasise the fact that a formula Ψ ranges over the set of variables X, we sometimes denote it by $\Psi(X)$.

Based on $(\mathbb{R}, 0, 1, +, \leq)$, we can defined a so-called *algebra of regions* [11] to effectively represent sets of states. The regions[2] in that algebra can be seen as functions R from the set of locations Loc to quantifier free formula of $(\mathbb{R}, 0, 1, +, \leq)$ with free variables in X, representing sets of valuations for the variables of the RHA. More precisely, any region R represents the set of states $[\![R]\!] = \{(\ell, \nu) \mid \nu \in [\![R(\ell)]\!]\}$. As $(\mathbb{R}, 0, 1, +, \leq)$ is closed under all Boolean operations, so is the region algebra. Since the logic is decidable, testing whether $s \in [\![R]\!]$ or whether $[\![R]\!] = \emptyset$ are both decidable problems.

In order to obtain fixpoint expressions that characterise $\mathsf{Reach}^{\leq \mathbf{T}}(s)$ and $\mathsf{coReach}^{\leq \mathbf{T}}(s)$ using the region algebra, we introduce $\mathsf{post}^\sharp$ and $\mathsf{pre}^\sharp$ operators ranging over regions. Let R be a region. We let $\mathsf{post}^\sharp(R)$, be the region s.t. for all $\ell \in \mathrm{Loc}$, $\mathsf{post}^\sharp(R)(\ell)$ is obtained by eliminating quantifiers in $\Psi^E_\ell(X) \vee \Psi^t_\ell(X)$, where $\Psi^E_\ell(X)$ characterises all the successors of $R(\ell)$ by an edge with source ℓ, and $\Psi^t_\ell(X)$ represents all the successors of $R(\ell)$ by a flow transition in ℓ (time elapsing). The following equations define Ψ^E_ℓ and Ψ^t_ℓ, both ranging on the set of free variables X:

$$\Psi^E_\ell = \bigvee_{e \in \mathrm{Edges}} \psi^e_\ell$$

$$\psi^{(\ell,g,Y,\ell')}_\ell = \exists X' : \begin{pmatrix} & R(\ell)(X') \wedge g(X') \\ \wedge & \bigwedge_{x \in X \setminus Y} x = x' \wedge \bigwedge_{x \in Y} x = 0 \\ \wedge & \mathrm{Inv}(\ell)(X') \wedge \mathrm{Inv}(\ell')(X) \end{pmatrix}$$

$$\Psi^t_\ell = \exists t : \exists X' : \begin{pmatrix} & t \geq 0 \wedge R(\ell)(X') \\ \wedge & \mathrm{Inv}(\ell)(X) \wedge \mathrm{Inv}(\ell)(X') \\ \wedge & \bigwedge_{x \in X} x' + t \cdot \min(\mathrm{Rates}(\ell, x)) \leq x \\ \wedge & \bigwedge_{x \in X} x \leq x' + t \cdot \max(\mathrm{Rates}(\ell, x)) \end{pmatrix}$$

Symmetrically, we let $\mathsf{pre}^\sharp(R)$ be the region s.t. for all $\ell \in \mathrm{Loc}$, $\mathsf{post}^\sharp(R)(\ell)$ is obtained by eliminating quantifiers in $\Phi^E_\ell(X) \vee \Phi^t_\ell(X)$, where $\Phi^E_\ell(X)$ represents all the predecessors of $R(\ell)$ by an edge whose target is ℓ, and $\Phi^t_\ell(X)$ represents all the

[2] The notion of region used in this section differs from the notion of region given by $\mathsf{Reg}\,(\mathrm{cmax}, X)$ and used to define $\mathsf{R}\,(\mathcal{H}')$. Notice however that any region from $\mathsf{R}\,(\mathcal{H}')$ can be expressed via a quantifier free formula of $(\mathbb{R}, 0, 1, +, \leq)$ with free variables in X. The converse is obviously not true.

predecessors of $R(\ell)$ by a flow transition in ℓ:

$$\Phi^E_\ell = \bigvee_{e \in \mathrm{Edges}} \varphi^e_\ell$$

$$\varphi^{(\ell,g,Y,\ell')}_\ell = \exists X' : \left(\begin{array}{ll} & R(\ell')(X') \wedge g(X) \\ \wedge & \bigwedge_{x \in X \setminus Y} x = x' \wedge \bigwedge_{x \in Y} x' = 0 \\ \wedge & \mathrm{Inv}(\ell)(X) \wedge \mathrm{Inv}(\ell')(X') \end{array} \right)$$

$$\Phi^t_\ell = \exists t : \exists X' : \left(\begin{array}{ll} & t \geq 0 \wedge R(\ell)(X') \\ \wedge & \mathrm{Inv}(\ell)(X) \wedge \mathrm{Inv}(\ell)(X') \\ \wedge & \bigwedge_{x \in X} x + t \cdot \min(\mathrm{Rates}(\ell, x)) \leq x' \\ \wedge & x' \leq x + t \cdot \max(\mathrm{Rates}(\ell, x)) \end{array} \right)$$

To keep the above definitions compact, we have implicitly assumed that the rates are given as *closed* rectangles. The definitions of Φ^t_ℓ and Ψ^t_ℓ can be adapted to cope with intervals that are left (respectively right) open by substituting $<$ ($>$) for $\leq$ ($\geq$).

In practice formulas in $(\mathbb{R}, 0, 1, +, \leq)$ can be represented and manipulated as finite union of convex polyhedra for which there exist efficient implementations, see [3] for example. Those techniques have been implemented in HYTECH [8] and PHAVER [6]. Unfortunately, termination of the symbolic model-checking algorithms is not ensured for linear hybrid automata. While in the literature, it is known that forward reachability and backward reachability fixpoint algorithms terminate for initialised rectangular hybrid automata [10], we show here that termination is also guaranteed for *time-bounded fixpoint formulas* over the class of $\mathrm{RHA}^{\geq 0}$ (that are not necessarily initialised).

Time-bounded forward and backward fixpoints Let $\mathcal{H}$ be an $\mathrm{RHA}^{\geq 0}$ with set of variables X, and let $\mathbf{T} \in \mathbb{N}$ be a time bound. Let us augment $\mathcal{H}$ with a fresh variable t to measure time (hence the rate of t is 1 in all locations, and t is never reset). Let R be region over the variables X. Then, it is easy to see that the following fixpoint equations characterise respectively $\mathsf{Reach}^{\leq \mathbf{T}}(\llbracket R \rrbracket)$ and $\mathsf{coReach}^{\leq \mathbf{T}}(\llbracket R \rrbracket)$:

$$\mathsf{Reach}^{\leq \mathbf{T}}(\llbracket R \rrbracket) = \mu Y \cdot ((\llbracket R(X) \rrbracket \cup \mathsf{Post}(Y)) \cap \llbracket 0 \leq t \leq \mathbf{T} \rrbracket) \tag{4}$$

$$\mathsf{coReach}^{\leq \mathbf{T}}(\llbracket R \rrbracket) = \mu Y \cdot ((\llbracket R(X) \rrbracket \cup \mathsf{Pre}(Y)) \cap \llbracket 0 \leq t \leq \mathbf{T} \rrbracket) \tag{5}$$

The next lemma ensures that these fixpoints can be effectively computed. The proof rely on Corollary 2.

Lemma 10 *For all RHA$^{\geq 0}$ $\mathcal{H}$, all region R and all time bound $\mathbf{T}$, the least fix points (4) and (5) are respectively equal to the limit of $F_0, F_1, F_2, \ldots$ and $B_0, B_1, B_2, \ldots$ where:*

$$\begin{array}{lll} F_0 = \llbracket R(X) \wedge 1 \leq t \leq \mathbf{T} \rrbracket & & \\ F_i = (\mathsf{Post}(F_{i-1}) \cap \llbracket 0 \leq t \leq \mathbf{T} \rrbracket) \cup F_{i-1} & & \textit{for all } i > 0 \\ B_0 = \llbracket R(X) \wedge 1 \leq t \leq \mathbf{T} \rrbracket & & \\ B_i = B_i = (\mathsf{Pre}(B_{i-1}) \cap \llbracket 0 \leq t \leq \mathbf{T} \rrbracket) \cup B_{i-1} & & \textit{for all } i > 0 \end{array}$$

Furthermore, both sequences stabilize after at most $F(\mathcal{H}, \mathbf{T})$ iterations, and both fixpoints can be computed in worst-case doubly exponential time.

Proof. We justify the result for the least fixpoint equation (4), the result for the least fixpoint equation (5) is justified similarly.

By induction, it is easy to prove that, for all $i \geq 0$, F_i contains all the states that are reachable within $\mathbf{T}$ time units *and* by at most i transitions. By Corollary. 2, we know that all states that reachable within $\mathbf{T}$ time units are reachable by a run of length at most $F(\mathcal{H}, \mathbf{T})$. We conclude that $F_j = F_{j+1} = \mathsf{Reach}^{\leq \mathbf{T}}([\![R]\!])$ for $j = F(\mathcal{H}, \mathbf{T})$. All the operations for computing F_i from F_{i-1} take polynomial time in the size of F_{i-1}, and so the size of F_i is also guaranteed to be polynomial in F_{i-1}, the overall doubly-exponential time bound follows. □ Note that by our NEXPTIME-hardness result, this deterministic algorithm can be considered optimal (unless NEXPTIME=EXPTIME.) Let us now consider two examples to demonstrate that this approach can be applied in practice.

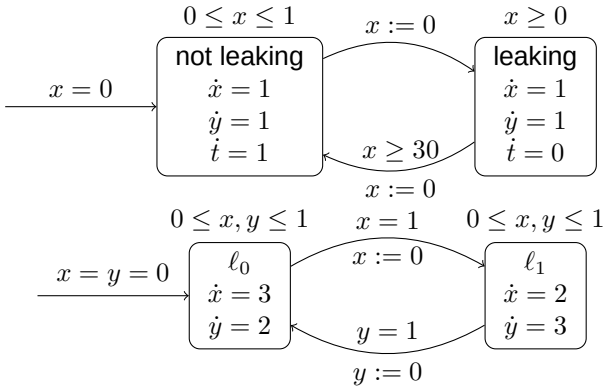

Figure 5: A stopwatch automaton for the leaking gas burner (top) and an SHA with bounded invariants (bottom).

Example 1: Leaking gas burner We present an example of a system where the classical fixpoint computation for reachability analysis does not terminate, while the time-bounded analysis does terminate. Consider the example of a leaking gas burner [1]. The gas burner can be either *leaking* or *not leaking*. Leakages are repaired within 1 second, and no leakage can happen in the next 30 seconds after a repair. In Fig. 5 (top), an automaton with two locations and the clock x is a model of the gas burner. In order to measure the leakage time and the total elapsed time, the stopwatch t and clock y are used as monitors of the system. It was shown using backward reachability analysis that in any time interval of at least 60 seconds, the time of leakage is at most one twentieth of the elapsed time [8]. The fixpoint is computed after 7 iterations of the backward reachability algorithm. However, the forward reachability analysis does not terminate.

Using forward time-bounded reachability analysis we can prove the property that in all time intervals of fixed length $T \geq 60$, the leakage time is at most $\frac{T}{20}$. In order to prove that this property holds in *all* time intervals, we perform the reachability analysis from all possible states of the system (i.e., from location leaking with $0 \leq x \leq 1$, and location not_leaking with $x \geq 0$) and starting with $t = y = 0$. For a fixed time bound T, we compute the set of reachable states satisfying $y \leq T$ and check that $t \leq \frac{T}{20}$ when $y = T$. The results of this paper guarantees that the analysis terminates. Using

HyTech, the property is established for $T = 60$ after 5 iterations of the forward time-bounded fixpoint algorithm. Thus for all time intervals of $T = 60$ seconds, the leakage time is at most $\frac{T}{20}$.

Example 2: bounded invariant In Fig. 5 (bottom), we consider a rectangular automaton with positive rates where all variables have a bounded invariant $[0, 1]$. In this example, the forward reachability analysis of HyTech does not terminate because the set of reachable states is not a finite union of polyhedra (see Fig. 6). On the other hand, the time-bounded forward fixpoint terminates by Lemma 10. This example shows that it is not sufficient to bound the variables in the automaton to get termination, but it is necessary to bound the time horizon of the analysis.

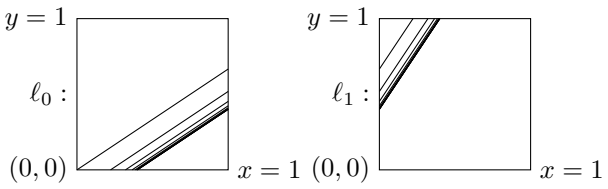

Figure 6: **Reachable states for the automaton of Fig. 5 (bottom).**

References

[1] R. Alur, C. Courcoubetis, T. A. Henzinger, and P.-H. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In *Hybrid Systems*, LNCS 736, pages 209–229. Springer, 1993.

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

[3] R. Bagnara, P. M. Hill, and E. Zaffanella. The parma polyhedra library: Toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems. *Sci. Comput. Program.*, 72(1-2):3–21, 2008.

[4] S. Basu. New results on quantifier elimination over real closed fields and applications to constraint databases. *J. ACM*, 46(4):537–555, 1999.

[5] T. Brihaye, L. Doyen, G. Geeraerts, J. Ouaknine, J.-F. Raskin, and J. Worrell. On reachability for hybrid automata over bounded time. In *ICALP (2)*, volume 6756 of *Lecture Notes in Computer Science*, pages 416–427. Springer, 2011.

[6] G. Frehse. Phaver: algorithmic verification of hybrid systems past hytech. *STTT*, 10(3):263–279, 2008.

[7] T. A. Henzinger. The theory of hybrid automata. In *LICS*, pages 278–292. IEEE Computer Society, 1996.

[8] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi. A user guide to HYTECH. In *TACAS'95: Tools and Algorithms for the Construction and Analysis of Systems*, volume 1019 of *Lecture Notes in Computer Science*, pages 41–71. Springer-Verlag, 1995.

[9] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi. Hytech: A model checker for hybrid systems. In *CAV*, volume 1254 of *Lecture Notes in Computer Science*, pages 460–463. Springer, 1997.

[10] T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata. *Journal of Computer and System Sciences*, 57(1):94–124, 1998.

[11] T. A. Henzinger, R. Majumdar, and J.-F. Raskin. A classification of symbolic transition systems. *ACM Trans. Comput. Log.*, 6(1):1–32, 2005.

[12] M. Jenkins, J. Ouaknine, A. Rabinovich, and J. Worrell. Alternating timed automata over bounded time. In *LICS*, pages 60–69. IEEE Computer Society, 2010.

[13] S. K. Jha, B. H. Krogh, J. E. Weimer, and E. M. Clarke. Reachability for linear hybrid automata using iterative relaxation abstraction. In *HSCC*, volume 4416 of *Lecture Notes in Computer Science*, pages 287–300. Springer, 2007.

[14] J. Ouaknine, A. Rabinovich, and J. Worrell. Time-bounded verification. In *CONCUR*, volume 5710 of *Lecture Notes in Computer Science*, pages 496–510. Springer, 2009.

Proof of Lemma 2

Proof. With each run $\rho = (\ell_0, \nu_0), (t_1, e_1), (\ell_1, \nu_1), \ldots, (t_n, e_n), (\ell_n, \nu_n)$ of $\mathcal{H}'$, we associate a run $\rho' = ((\ell_0, r_0), \nu_0), (t_0, e'_0), ((\ell_1, r_1), \nu_1), \ldots, (t_n, e'_n), ((\ell_n, r_n), \nu_n)$ of $\mathsf{R}(\mathcal{H}')$ (hence with $\mathsf{duration}(\rho) = \mathsf{duration}(\rho')$) s.t.:

- for all $0 \leq i \leq n$, for all $x \in X$ (assuming $t_{n+1} = 0$):

$$r_i(x) = \begin{cases} \mathbf{0}^+ & \text{if } \nu_i(x) = 0 \text{ and } (t_{i+1} > 0 \text{ and } \dot{x} \neq 0) \\ \mathbf{0}^= & \text{if } \nu_i(x) = 0 \text{ and } (t_{i+1} = 0 \text{ or } \dot{x} = 0) \\ [\nu_i(x)] & \text{otherwise} \end{cases}$$

- e'_i is the unique edge between (ℓ_i, r_i) and (ℓ_{i+1}, r_{i+1}), corresponding to e_i.

We prove by (backward) induction (on n) that the run ρ' is a genuine run of $\mathsf{R}(\mathcal{H}')$. When $n = 0$, there is nothing to prove. Let us now assume that given $\rho = (\ell_0, \nu_0), (t_1, e_1), (\ell_1, \nu_1), \ldots, (t_n, e_n), (\ell_n, \nu_n)$ run of $\mathcal{H}'$, we have proved that $((\ell_1, r_1), \nu_1), \ldots, (t_n, e'_n), ((\ell_n, r_n), \nu_n)$ is a genuine of $\mathsf{R}(\mathcal{H}')$. To obtain the desired result, it remains to prove that $((\ell_0, r_0), \nu_0), (t_0, e'_0), ((\ell_1, r_1), \nu_1)$ is a genuine run of $\mathsf{R}(\mathcal{H}')$. For this, we have to prove that for all $x \in X$: *(i)* $\nu_0(x) + t \times \mathrm{Rates}(\ell_0, r_0)(x) \models \mathrm{Inv}(\ell_0, r_0)$, for all $0 \leq t \leq t_1$, *(ii)* $\nu_0(x) + t_1 \times \mathrm{Rates}(\ell_0, r_0)(x) \models g'$ (where g' is the guard of the transition e'_1), and *(iii)* $\nu_1(x) = 0$ (resp. $\nu_1(x) = \nu_0(x)$) if $x \in Y'$ (resp. $x \notin Y'$) (where Y' is the reset of the transition e'_1). Let us distinguish three cases:

1. Case 1: $r(x) = \mathbf{0}^+$. In this case, by construction of ρ', we know that $\nu_0(x) = 0$, $t_1 > 0$ and $\dot{x} \neq 0$. In particular, we have that $\nu_0(x) + t_1 \times \mathrm{Rates}(\ell_0, r_0)(x) > 0$. By construction of $\mathsf{R}(\mathcal{H}')$, we know that $g'(x) = g(x) \wedge r''(x) \wedge (x > 0)$, where

$g(x)$ (resp. $g'(x)$) represents the constraints[3] on x in the guard of the transition e_1 (resp. e'_1), and $r \leq_{\text{ts}}^{\text{Rates}(\ell_0,r_0)} r''$. Moreover we have that $\text{Inv}(\ell_0, r_0)(x) = \text{Inv}(\ell_0)(x)$. Since ρ is a genuine run of $\mathcal{H}'$, we clearly have that *(i)* and *(iii)* are satisfied. Point *(ii)* follows from the facts that ρ is a genuine run of $\mathcal{H}'$ and that $\nu_0(x) + t_1 \times \text{Rates}(\ell_0, r_0)(x) > 0$.

2. Case 2: $r(x) = \mathbf{0}^=$. In this case, by construction of ρ', we know that $\nu_0(x) = 0$, $t_1 = 0$ or $\dot{x} = 0$. In particular, we have that $\nu_0(x) + t_1 \times \text{Rates}(\ell_0, r_0)(x) = 0$. By construction of $\mathsf{R}(\mathcal{H}')$, we know that $g'(x) = g(x) \wedge r''(x) \wedge (x = 0)$, where $g(x)$ (resp. $g'(x)$) represents the constraints on x in the guard of the transition e_1 (resp. e'_1), and $r \leq_{\text{ts}}^{\text{Rates}(\ell_0,r_0)} r''$. Moreover we have that $\text{Inv}(\ell_0, r_0)(x) = \text{Inv}(\ell_0)(x) \wedge (x = 0)$. Since ρ is a genuine run of $\mathcal{H}'$, we clearly have that *(iii)* is satisfied. Points *(i)* and *(ii)* follow from the facts that ρ is a genuine run of $\mathcal{H}'$ and that $\nu_0(x) + t \times \text{Rates}(\ell_0, r_0)(x) = 0$, for all $0 \leq t \leq t_1$.

3. Case 3: $r(x) \notin \{\mathbf{0}^=, \mathbf{0}^+\}$. This case is simpler than the two previous ones. The three points *(i)*, *(ii)* and *(iii)* follow from the facts that ρ is a genuine run of $\mathcal{H}'$.

Then, with each run $\rho' = ((\ell_0, r_0), \nu_0), (t_0, e'_0), ((\ell_1, r_1), \nu_1), \ldots, (t_n, e'_n), ((\ell_n, r_n), \nu_n)$ of $\mathsf{R}(\mathcal{H}')$ we associate the run $\rho = (\ell_0, \nu_0), (t_1, e_1), (\ell_1, \nu_1), \ldots, (t_n, e_n), (\ell_n, \nu_n)$ where, for all $1 \leq i \leq n$, e_i is the unique edge of $\mathcal{H}'$ that corresponds to e'_i. Since the guards and invariant of $\mathsf{R}(\mathcal{H}')$ are more constraining than those of $\mathcal{H}'$, the fact that ρ' is a genuine run of $\mathsf{R}(\mathcal{H}')$ implies that ρ is a genuine run of $\mathcal{H}'$. □

Proof of Lemma 3

Proof. Let us first prove that, for all $0 \leq i \leq n$: $\nu_i \in r_i$. The proof is by induction on i. For $i = 0$, the property holds by definition of $\mathsf{R}(\mathcal{H}')$ and because $((\ell_0, r_0), \nu_0)$ is an initial state. Assume $\nu_i \in r_i$ for some $i \geq 0$, and let us show that $\nu_{i+1} \in r_{i+1}$. Let g and Y denote respectively the guard and the reset set of e_{i+1}. Let $\nu' = \nu_i + t_{i+1} \times \text{Rates}(\ell_i)$ be the valuation of the variables when crossing e_{i+1}. By construction, we know that there is a region r'' which is a conjunct of g (hence $\nu' \in r''$) s.t. for all $x \notin Y$: $r''(x) = r_{i+1}(x)$. Thus, for all $x \notin Y$: $\nu'(x) = \nu_{i+1}(x) \in r_{i+1}(x)$. Moreover, still by construction, for all $x \in Y$: $r_{i+1}(x) \in \{\mathbf{0}^=, \mathbf{0}^+\}$. Hence, for all $x \in Y$: $\nu_{i+1}(x) = 0 \in r_{i+1}$ too.

To conclude, the two last points of the lemma follow immediately from the construction of $\mathsf{R}(\mathcal{H}')$, as for all edge e and all variable x s.t. $\mathsf{src}(e) = (\ell, r)$ and $r(x) = \mathbf{0}^=$ (resp. $r(x) = \mathbf{0}^+$), the constraint $x = 0$ ($x > 0$) appears as a conjunct of e's guard. □

[3]Notice that it makes sense to decouple guard according to variables since there are no diagonal constraints.