$1\frac{1}{2}$-Player Stochastic StopWatch Games

Sparsa Roychowdhury
Indian Institute of Technology Bombay, Mumbai, India

Abstract

Stochastic timed games (STGs), introduced by Bouyer and Forejt, generalize continuous-time Markov chains and timed automata. Depending on the number of players – 2, 1, or 0 – subclasses of stochastic timed games are classified as $2\frac{1}{2}$-player, $1\frac{1}{2}$-player, and $\frac{1}{2}$-player games where the $\frac{1}{2}$ symbolizes the presence of the stochastic player. The qualitative and quantitative reachability problem for STGs was studied in [10] and [1]. In this paper, we introduce stochastic stopwatch games (SSG), an extension of (STG) from clocks to stopwatches. We focus on $1\frac{1}{2}$-player SSGs and prove that with two variables which can be either a clock or a stopwatch, qualitative reachability is decidable, whereas, if we increase the number of variables to three, with at least one stopwatch, the problem becomes undecidable.

2012 ACM Subject Classification Theory of computation → Timed and hybrid models

Keywords and phrases Timed Automata, Stopwatches, Stochastic Timed Games

Digital Object Identifier 10.4230/LIPIcs.TIME.2021.17

Acknowledgements The author thanks Prof. Krishna S. of IIT Bombay, India for insightful discussions, suggestions and encouragement towards this work. The open access publication of this article was supported by the Alpen-Adria-Universität Klagenfurt, Austria.

1 Introduction

Two-player zero-sum games are well studied for controller synthesis of discrete event systems. But, they are not sufficient to model real-time probabilistic systems. To model real-time systems, one needs to capture the semantics of time. Timed automata [4] are a well-known and extensively studied formalism widely used to model real timed systems. Timed automata models time with a finite set of real valued variables known as clocks. The reachability problem of timed automata is showed to be PSPACE-Complete using a special abstraction known as region automata [4]. But, only clocks are not sufficient to model stochastic behaviors of a system. In [7] probabilistic semantics were added in timed automata where choices of time and transitions are randomized. The probabilistic notion was mostly used to solve the "almost sure model checking" [8] of timed automata i.e., to check if a property is satisfied with some certainty or not. Different formalisms like probabilistic timed automata [18], continuous probabilistic timed automata [17], continuous timed Markov chains [6], and stochastic timed automata [9] have been proposed that capture both of the real-time and stochastic nature of the system. Timed games [5] are a natural extension of timed automata to model interactive systems in a more robust manner. Stochastic timed games (STG in short) was proposed in [10], which extends timed games with probabilities. Just like timed games, the locations are partitioned among players but in STG there is a special player known as the "environment". A player can only process their move if she is in a location that belongs to her. The player "environment" is different from other players in the sense that it can choose delays and transitions stochastically based on a distribution. Hybrid automata [3] is a powerful formalism that uses more generalized real valued variables to model hybrid systems. Unlike clocks, the value of the variables in a hybrid automaton changes depending on a function defined on the locations. These functions can be linear as well as non-linear. But, most of the interesting problems of hybrid automata like reachability are

28th International Symposium on Temporal Representation and Reasoning (TIME 2021).
Editors: Carlo Combi, Johann Eder, and Mark Reynolds; Article No. 17; pp. 17:1–17:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

undecidable [14]. Hybrid automata have been studied extensively under different constraints like time-bounded hybrid automata [12], initialized hybrid automata [20], singular hybrid automata [16], etc. A hybrid automaton is linear if its constraints can be expressed as linear expressions over the set of variables [2]. A stopwatch [13] is a real-valued variable that can either track time like clocks or can choose to stay in the current value depending on the current location. It is shown in [13] that a minor upgrade from timed automata to stopwatch automata immediately yields the full expressive power of linear hybrid automata.

In case of stochastic systems reachability, one can associate a probability parameter p with reachability query to ask, “is a state reachable from another state with probability p?”. The parameter p is called threshold probability. Depending on the value of threshold probability, we can have two different types of reachability queries,

Quantitative Reachability: When the constraint on the threshold is $0 < p < 1$,

Qualitative Reachability: When the constraint on the threshold is $p \in \{0, 1\}$.

It is known that [10], the qualitative reachability problem is decidable for $1\frac{1}{2}$-player stochastic timed games with one clock, and quantitative reachability is undecidable for $2\frac{1}{2}$-player stochastic timed games with ≥ 3 clocks. These results were further refined in [1], where it was shown that the qualitative reachability problem is undecidable for $1\frac{1}{2}$-STGs with four or more clocks, and the same problem is undecidable for $2\frac{1}{2}$-STGs for three or more clocks.

Just as stopwatches generalizes clocks, we generalize stochastic timed games to stochastic stopwatch games (SSG in short) by replacing clocks with stopwatches. We solve the qualitative reachability problem on this extended model. Our focus in this paper is only on qualitative reachability and $1\frac{1}{2}$ player games on SSG. We keep the case of two and a half player qualitative reachability for future work. Our main results are,

(1) The qualitative reachability problem is EXPTIME-Complete for $1\frac{1}{2}$ player stochastic stopwatch games with two stopwatches.

(2) The qualitative reachability problem for $1\frac{1}{2}$ player stochastic stopwatch games is undecidable (Π_1^0 hard) with three stopwatches.

Our results give a tight demarcation between decidability and undecidability in the case of $1\frac{1}{2}$ player SSGs.

2 Preliminaries

We use standard notations for the set of reals ($\mathbb{R}$), rationals ($\mathbb{Q}$), and natural numbers ($\mathbb{N}$), and add subscripts to indicate additional constraints (for instance $\mathbb{R}_{\geq 0}$ is for the set of non-negative reals). Let $\mathcal{X}$ be a finite set of real-valued variables called *clocks*. A *valuation* on $\mathcal{X}$ is a function $\nu : \mathcal{X} \to \mathbb{R}_{\geq 0}$. We assume an arbitrary but fixed ordering on the clocks and write x_i for the clock with order i. This allows us to treat a valuation ν as a point $(\nu(x_1), \nu(x_2), \ldots, \nu(x_n)) \in \mathbb{R}_{\geq 0}^{|\mathcal{X}|}$. For a subset of clocks $R \subseteq \mathcal{X}$ and valuation $\nu \in \mathbb{R}^{|\mathcal{X}|}$, we write $\nu[R]$ for the valuation where $\nu[R](x) = 0$ if $x \in R$, and $\nu[R](x) = \nu(x)$ otherwise. For $t \in \mathbb{R}_{\geq 0}$, write $\nu + t$ for the valuation defined by $\nu(x) + t$ for all $x \in \mathcal{X}$. The valuation $\mathbf{0} \in \mathbb{R}^{|\mathcal{X}|}$ is a special valuation such that $\mathbf{0}(x) = 0$ for all $x \in \mathcal{X}$. A constraint (or guard) over $\mathcal{X}$ is a subset of $\mathbb{R}_{\geq 0}^{|\mathcal{X}|}$ defined by a (finite) conjunction of constraints of the form $x \bowtie k$, where $k \in \mathbb{N}$, $x \in \mathcal{X}$, and $\bowtie \in \{<, \leq, =, >, \geq\}$. We write $\mathsf{rect}(\mathcal{X})$ for the set of constraints on $\mathcal{X}$. For a constraint $\varphi \in \mathsf{rect}(\mathcal{X})$, and a valuation ν, we write $\nu \models \varphi$ to represent the fact that valuation ν satisfies constraint φ (defined in a natural way).

▶ **Definition 1** (Timed Automata [4]). *A timed automaton is a tuple $\mathcal{A} = (\mathcal{Q}, \mathcal{Q}_0, \mathcal{X}, \Delta, F)$ where $\mathcal{Q}$ is a finite set of locations, $\mathcal{Q}_0 \subseteq \mathcal{Q}$ is a set of initial locations, $\mathcal{X}$ is a finite set of clocks, $F \subseteq \mathcal{Q}$ is a set of accepting locations and Δ is a set of transitions of the form (l_1, φ, R, l_2) where, $l_1, l_2 \in \mathcal{Q}$, $R \subseteq \mathcal{X}$ is known as the set of reset clocks, and $\varphi \in \mathsf{rect}(\mathcal{X})$.*

A state s of timed automata is a pair $s = (l, \nu) \in (\mathcal{Q} \times \mathbb{R}^{|\mathcal{X}|}_{\geq 0})$ consists of a location and valuation. A transition (t, e) from a state $s = (l, \nu)$ to a state $s' = (l', \nu')$ is written as $s \xrightarrow{t,e} s'$ if $e = (l, \varphi, R, l') \in \Delta$, such that $\nu + t \models \varphi$, and $\nu' = (\nu + t)[R](x)$.

A run is a finite or infinite sequence of transitions $\rho = s_0 \xrightarrow{t_1,e_1} s_1 \xrightarrow{t_2,e_2} s_2 \ldots$ of states and transitions. An edge e is enabled from s whenever there is a state s' such that $s \xrightarrow{0,e} s'$. Given a state s of $\mathcal{A}$ and an edge e, we define $\mathcal{E}(s, e) = \{t \in \mathbb{R}_{\geq 0} \mid s \xrightarrow{t,e} s'\}$ for some s' and $\mathcal{E}(s) = \bigcup_{e \in \Delta} \mathcal{E}(s, e)$. We say that $\mathcal{A}$ is non-blocking if and only if for all states s, $\mathcal{E}(s) \neq \emptyset$.

▶ **Definition 2** (Singular Stopwatch Automata). *A Singular Stopwatch automaton is a tuple* $\mathcal{H} = (\mathcal{Q}, \mathcal{Q}_0, V, \Delta, \mathfrak{R}, F)$ *where,*

- $\mathcal{Q}$ *is a finite set of locations including a distinguished initial set of locations* $\mathcal{Q}_0 \subseteq \mathcal{Q}$,
- V *is an (ordered) set of variables called stopwatches,*
- $\Delta \subseteq \mathcal{Q} \times \mathsf{rect}(V) \times 2^V \times \mathcal{Q}$ *is the set of* transitions *of the form* (l, φ, R, l') *such that,* $l, l' \in \mathcal{Q}$, $\varphi \in \mathsf{rect}(V)$, *and* $R \subseteq V$ *subset of variables that are reset during the transition.*
- $\mathfrak{R} : \mathcal{Q} \to \{0, 1\}^{|V|}$ *is the location dependent* flow function *characterizing the rate of each variable in each location.*
- $F \subseteq \mathcal{Q}$ *is the set of final locations.*

A variable $x \in V$ is a clock when for all locations $l \in \mathcal{Q}$, $\mathfrak{R}(l)[x] = 1$, and a variable is a stopwatch when, for all locations $l \in \mathcal{Q}$, $\mathfrak{R}(l)[x] \in \{0, 1\}$. Just like timed automata the map $\nu : V \to \mathbb{R}_{\geq 0}$ represents the current valuation of the variables, we define a state of $\mathcal{H}$ as a pair of locations and valuations $(l, \nu) \in (\mathcal{Q} \times \mathbb{R}^{|V|}_{\geq 0})$. For a state $s = (l, \nu)$ of $\mathcal{H}$ and $t \in \mathbb{R}_{\geq 0}$ we define $s + t = (l, \nu + t)$ where, $(\nu + t)(x) = \nu(x) + \mathfrak{R}(l)[x] \cdot t, \forall x \in V$. A transition (t, e) of $\mathcal{H}$ from a state $s = (l, \nu)$ to a state $s' = (l', \nu')$ is written as $s \xrightarrow{t,e} s'$ if $e = (l, \varphi, R, l') \in \Delta$, such that $\nu + t \models \varphi$, and $\nu' = (\nu + t)[R](x)$. A run of $\mathcal{H}$ is $\rho = (l_0, \nu_0) \xrightarrow{t_1,e_1} (l_1, \nu_1) \xrightarrow{t_2,e_2} \ldots$ of states and transitions. The notions of non blocking states, and $\mathcal{E}(s) = \bigcup_{e \in \Delta} \mathcal{E}(s, e)$ for all states carry over as in timed automata.

Singular Stopwatch automata are a special case of singular hybrid automata [16], when the variables are stopwatches. We now formally define the stochastic stopwatch games in the same line of stochastic timed games as defined in [10].

▶ **Definition 3** ($1\frac{1}{2}$-Stochastic Stopwatch Games ($1\frac{1}{2}$-SSG)). *A* $1\frac{1}{2}$*-player stochastic stopwatch game is a tuple* $\mathcal{SG} = (\mathcal{H}, \mathcal{Q}_\Diamond, \mathcal{Q}_\bigcirc, w, \mu)$ *where* $\mathcal{H} = (\mathcal{Q}, \mathcal{Q}_0, V, \Delta, \mathfrak{R}, F)$ *is a singular stopwatch automaton,* $(\mathcal{Q}_\Diamond, \mathcal{Q}_\bigcirc)$ *is a partition of* $\mathcal{Q}$ *such that they are controlled by players* $\Diamond$ *and* $\bigcirc$ *respectively,* w *is a map that assigns weight to each transition leaving* $\mathcal{Q}_\bigcirc$*, and* μ *is a function that assigns a measure over* $\mathcal{E}(s)$ *for every state* $s \in \mathcal{Q}_\bigcirc \times \mathbb{R}^{|\mathcal{X}|}_{\geq 0}$. *The function* $\mu(s)$ *satisfies the following properties,*

(1) $\mu(s)(\mathcal{E}(s)) = 1$ *(Law of total probability)*
(2) *Let* λ *be the Lebesgue measure, if* $\lambda(\mathcal{E}(s)) > 0$ *then for each measurable set* $B \subseteq \mathcal{E}(s)$ *we have* $\lambda(B) = 0$ *if and only if* $\mu(s)(B) = 0$. *The choice of measures is such that the measures evolve smoothly while moving from one state to another.*

The singular stopwatch automaton $\mathcal{H}$ is equipped with uniform distributions over delays if for every state s, $\mathcal{E}(s)$ is bounded, and $\mu(s)$ is the uniform distribution over $\mathcal{E}(s)$. $\mathcal{H}$ is equipped with exponential distributions over delays whenever, for every state s, either $\mathcal{E}(s)$ has Lebesgue measure zero, or $\mathcal{E}(s) = \mathbb{R}_{\geq 0}$ and for every location l, there is a positive rational α_l such that $\mu(s)(\mathcal{E}) = \int_{t \in \mathcal{E}} \alpha_l e^{-\alpha_l t} dt$. We assume $\alpha_l = 1$ for all locations l. The locations in $\mathcal{Q}_\Diamond$ are controlled by player $\Diamond$. The locations in $\mathcal{Q}_\bigcirc$ are governed by probabilistic laws. For $s \in \mathcal{Q}_\bigcirc \times \mathbb{R}^{|V|}_{\geq 0}$, both delays and discrete moves will be chosen probabilistically: from s, a delay d is chosen following the probability distribution over delays $\mu(s)$. Then, from state

$s+d$, an enabled edge is selected following a discrete probability distribution that is given in a usual way with the weight function w: in state $s+t$, the probability of edge e (if enabled), denoted $p(s+t)(e)$ is $\frac{w(e)}{\sum_{e'}\{w(e')|e' \text{ is enabled in } s+t\}}$. This way of probabilizing behaviours in timed automata has been presented in [10]. We refer to $\ell \in \mathcal{Q}_{\bigcirc}$ as stochastic nodes and $\ell \in \mathcal{Q}_{\Diamond}$ as diamond ($\Diamond$) nodes.

Strategies. Let $\rho = (l_0, \nu_0) \xrightarrow{t_1,e_1} (l_1, \nu_1) \dots \xrightarrow{t_n,e_n} (l_n, \nu_n)$ be a finite run of SSG $\mathcal{SG}$. A strategy (for $\Diamond$) is a function that maps a finite run $\rho = (l_0, \nu_0) \xrightarrow{t_1,e_1} (l_1, \nu_1) \dots \xrightarrow{t_n,e_n} (l_n, \nu_n)$ to a pair (t, e) such that $(l_n, \nu_n) \xrightarrow{t,e} (l', \nu')$ for some (l', ν'), whenever $l_n \in \mathcal{Q}_{\Diamond}$. In order to measure probabilities of certain sets of runs, the following measurability condition is imposed on strategy $\lambda_{\Diamond}$: for every finite sequence of edges $e_1, \dots, e_n$ and every state s, the function $\chi : (t_1, \dots, t_n) \to (t, e)$ is such that $\chi(t_1, \dots, t_n) = (d, e)$ if and only if $\lambda_{\Diamond}(s_0 \overset{t_1,e_1}{\to} s_1 \overset{t_2,e_2}{\to} s_2 \dots \overset{t_n,e_n}{\to} s_n) = (t, e)$ is measurable.

Given SSG $\mathcal{SG}$, a finite run ρ ending in state s_0, and a strategy $\lambda_{\Diamond}$, we define $Runs(\mathcal{SG}, \rho, \lambda_{\Diamond})$ to be the set of all runs generated by $\lambda_{\Diamond}$ after prefix ρ; that is, the set of all runs of the automaton satisfying the following condition: If $s_i = (l_i, \nu)$ and $l \in \mathcal{Q}_{\Diamond}$, then $\lambda_{\Diamond}$ returns (t_{i+1}, e_{i+1}) when applied to $\rho \overset{t_1,e_1}{\to} s_1 \overset{t_2,e_2}{\to} \dots \overset{t_i,e_i}{\to} s_i$. Given a finite sequence $e_1, \dots, e_n$ of edges, a symbolic path $\pi_{\lambda_{\Diamond}}(\rho, e_1 \dots e_n)$ is defined as $\pi_{\lambda_{\Diamond}}(\rho, e_1 \dots e_n) = \{\rho' \in Runs(\mathcal{SG}, \rho, \lambda_{\Diamond}) \mid \rho' = s_0 \overset{t_1,e_1}{\to} s_1 \overset{t_2,e_2}{\to} s_2 \dots \overset{t_n,e_n}{\to} s_n, \text{ with } t_i \in \mathbb{R}_{\geq 0}\}$. When $\lambda_{\Diamond}$ is clear, we simply write $\pi(\rho, e_1 \dots e_n)$.

Given a strategy $\lambda_{\Diamond}$, and a finite run ρ ending in state $s = (l, \nu)$, a probability measure $\mathcal{P}_{\lambda_{\Diamond}}$ can be defined on the set $Runs(\mathcal{G}, \rho, \lambda_{\Diamond})$, following [10]. First, define $\mathcal{P}_{\lambda_{\Diamond}}$ on symbolic paths starting with ρ, $\mathcal{P}_{\lambda_{\Diamond}}(\pi(\rho)) = 1$. Then:

If $\ell \in \mathcal{Q}_{\Diamond}$, and $\lambda_{\Diamond}(\rho) = (t, e)$,

$$\mathcal{P}_{\lambda_{\Diamond}}(\pi(\rho, e_1 \dots e_n)) = \begin{cases} 0 & \text{if } e_1 \neq e \\ \mathcal{P}_{\lambda_{\Diamond}}(\pi(\rho \xrightarrow{t,e} s', e_2 \dots e_n)) & \text{otherwise} \end{cases}$$

If $\ell \in \mathcal{Q}_{\bigcirc}$

$$\mathcal{P}_{\lambda_{\Diamond}}(\pi(\rho, e_1 \dots e_n)) = \int_{t \in \mathcal{E}(s,e_1)} p(s+t)(e_1) \cdot \mathcal{P}_{\lambda_{\Diamond}}(\pi(\rho \xrightarrow{t,e_1} s', e_2 \dots e_n)) \, t\mu(s)(t)$$

where $s \xrightarrow{t,e_1} s'$ for every $d \in \mathcal{E}(s, e_1)$.

The correctness of these integrals is done as in [10], assuming some measurability conditions. When $\mathcal{Q}_{\Diamond} = \emptyset$ we call this game a $\frac{1}{2}$-player stochastic stopwatch games.

▶ **Example 4.** We give here an example to explain the method of computing probabilities in the ($\frac{1}{2}$-SSG) $\mathcal{SG}$ in the Figure 1. There is a single stopwatch x.

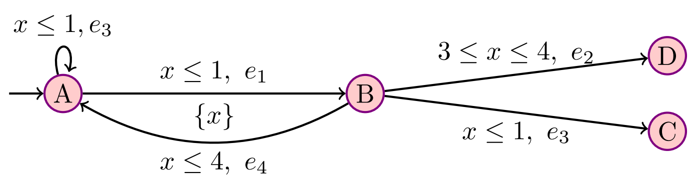

Figure 1 An example of $\frac{1}{2}$ SSG.

We consider two cases. First, assume the rate of x $\mathfrak{R}[l](x) = 0$ when, $l = B$, and $\mathfrak{R}[l](x) = 1$ when $l \neq B$. The initial state is $s_0 = (A, 0)$. Then:

$$\mathcal{P}(\pi((A,0),e_1e_3)) = \int_0^1 \frac{\mathcal{P}(\pi((B,0),e_3))}{2} d\mu_{(A,0)}(t) = \int_0^1 \frac{1}{2}\int_0^\infty \frac{1}{3}\cdot d\mu_{(B,0)}(t_1)\cdot d\mu_{(A,0)}(t)$$
$$= \int_0^1 \frac{1}{2}\frac{1}{3}\int_0^\infty e^{-t_1}dt_1 dt = \frac{1}{6}$$

$d\mu_{(A,0)}$ is the uniform distribution over [0,1]. Note that since the rate of x is 0 at B, an unbounded time can be spent, enabling transition e_3. Hence, $d\mu_{(B,0)}$ is the exponential distribution. Second, if we assume the rate of x is 1 in all locations (x is a clock), then we have the uniform distribution of all delays.

$$\mathcal{P}(\pi((A,0),e_1e_2)) = \int_0^1 \frac{\mathcal{P}(\pi((B,0),e_2))}{2} d\mu_{(A,0)}(t) = \int_0^1 \frac{1}{2}\int_3^4 \frac{1}{3}\cdot d\mu_{(B,0)}(t_1)\cdot d\mu_{(A,0)}(t)$$
$$= \int_0^1 \frac{1}{2}\frac{1}{3}\int_3^4 \frac{1}{4-0}dt_1 dt = \frac{1}{24}$$

Note that SSGs are defined on top of a stopwatch automata. In general, reachability is undecidable for stopwatch automata [13]. On restricting the number of stopwatches, we find the fine boundary between decidability and undecidability for $1\frac{1}{2}$-SSG.

Qualitative Reachability. We study the qualitative reachability problem for SSGs, stated as follows. Given a SSG $\mathcal{SG}$ with a set T of target locations, an initial state s_0 and $p \in \{0,1\}$, decide whether there is a strategy $\lambda_\Diamond$ for Player $\Diamond$ such that $\mathcal{P}_{\lambda_\Diamond}(\{\rho \in \mathit{Runs}(\mathcal{SG}, s_0, \lambda_\Diamond) \mid \rho \text{ visits } T\}) \bowtie p$. Now we state our main theorem,

▶ **Theorem 5.** *The qualitative reachability problem of $1\frac{1}{2}$-SSG is*
(1) *decidable with two stopwatches moreover, it is* EXPTIME *complete.*
(2) *undecidable with three variables (with at least one stopwatch, and other two are clocks).*

3 Qualitative Reachability of $1\frac{1}{2}$ SSG : the two stopwatches case

The qualitative reachability of $1\frac{1}{2}$ SSG $\mathcal{SG}$ consists of two major objectives, reaching a desired set of locations $\mathcal{F}$ of $\mathcal{SG}$, with probability greater than 0, and equal to 1, represented by Prob_Reach$_{>0}(\mathcal{F})$ and Prob_Reach$_{=1}(\mathcal{F})$ respectively. All other objectives can be achieved using these two. All the edges of $\mathcal{SG}$ have some positive probability greater than 0 because we can remove the negligible edges effectively [10]. Since objectives under consideration are Prob_Reach$_{>0}(\mathcal{F})$ and Prob_Reach$_{=1}(\mathcal{F})$, exact probability does not matter. Thus, we only need to check if there is some clock valuation which allows us to make a move, or if all valuations are good. Hence, it is sufficient to work with regions.

▶ **Lemma 6.** *Given a 2 variable $1\frac{1}{2}$ SSG $\mathcal{SG}$, and desired set of target locations $\mathcal{F}$, we can compute the set of states from which player $\Diamond$ has a strategy to attain the objective of* Prob_Reach$_{>0}(\mathcal{F})$.

Proof. The first thing to do is to work with the underlying singular stopwatch automaton $\mathcal{A}$ of the $\mathcal{SG}$. Since $\mathcal{A}$ has only 2 stopwatches, with some care, the region construction applies to $\mathcal{A}$ (Appendix A.1), and we can construct the region automaton $\mathcal{R}(\mathcal{A})$ corresponding to $\mathcal{A}$, such that there is a run ρ in $\mathcal{A}$ reaching a target location $T \in \mathcal{F}$ if and only if there is a run ρ' in $\mathcal{R}(\mathcal{A})$ reaching a corresponding target location T'. $\mathcal{R}(\mathcal{A})$ is an untimed automaton, and the locations of $\mathcal{R}(\mathcal{A})$ have the form (l, α) where l is a location of $\mathcal{A}$ and $\alpha \in \mathsf{Reg}(V)$ is a region over the variables V of $\mathcal{A}$. Thus, $T' = \{(T, \alpha) \mid \alpha \in \mathsf{Reg}(V), T \in \mathcal{F})\}$ is the set of target locations in $\mathcal{R}(\mathcal{A})$. Note that the region construction for singular stopwatch automata does not extend when there are 3 variables (see Appendix A.2).

Given a location $(l, \alpha) \in \mathcal{R}(\mathcal{A})$, we say that it belongs to $\mathcal{Q}_\Diamond$ if $l \in \mathcal{Q}_\Diamond$; likewise, $(l, \alpha) \in \mathcal{Q}_\bigcirc$ if $l \in \mathcal{Q}_\bigcirc$.

For brevity, in the following, we use ℓ to denote locations of $\mathcal{R}(\mathcal{A})$. Likewise we use $\mathcal{F}$ to denote target locations in $\mathcal{R}(\mathcal{A})$. Our reachability algorithm operates on the region automaton $\mathcal{R}(\mathcal{A})$. We do backward reachability, starting from the target $\mathcal{F}$. We construct a set Y as follows.

1. Initialize: $Y_0 = \mathcal{F}$
2. Repeatedly add locations ℓ to Y_i, to construct Y_{i+1} as follows, $Y_{i+1} = Y_i \cup \{\ell\}$,
   a. If $\ell \in \mathcal{Q}_\bigcirc$, and has at least one enabled edge going into the set Y_i.
   b. If $\ell \in \mathcal{Q}_\Diamond$, and has at least one enabled edge going into the set Y_i.

We will repeat Step-2 until a fixpoint is reached.

Now, we claim that: a location $\ell \in Y$ if and only if there exists a strategy of player $\Diamond$ from ℓ to attain the objective $\mathsf{Prob\_Reach}_{>0}(\mathcal{F})$.

($\Rightarrow$) Let $\ell \in Y$. The rank of a location ℓ is i if it is added to Y_i. We prove that if $\ell \in Y$, then player $\Diamond$ has a strategy such that $\mathcal{F}$ is reached with positive probability by inducting on the rank of locations. The base case is trivial when $i = 0$ since target locations have rank 0. Assume the result holds for ranks $\leqslant i$. Now, we will prove for rank $i + 1$. There can be two different cases depending on the type of location.

Case $\ell \in \mathcal{Q}_\Diamond$: If location ℓ belongs to player $\Diamond$, then the probability of reaching $\mathcal{F}$ from ℓ is equal to the probability of reaching some location ℓ' of rank i (because of which ℓ was added to Y) which can be reached from ℓ according to the strategy of player $\Diamond$.

Case $\ell \in \mathcal{Q}_\bigcirc$: If location ℓ is probabilistic, then there exists an enabled out-going edge from ℓ to $\ell' \in Y_i$ whose probability is greater than 0 (say p_1). Since $\ell \in Y_i$, it reaches $\mathcal{F}$ with some probability (say p_2). Then the probability of reaching $\mathcal{F}$ from ℓ is $p_1.p_2$, which is greater than 0.

($\Leftarrow$) If $\ell \notin Y$, then the probability of reaching $\mathcal{F}$ from ℓ is equal to 0. To prove this we will consider the following,

Case $\ell \in \mathcal{Q}_\bigcirc \cup \mathcal{Q}_\Diamond$: If ℓ belongs to player $\Diamond$, then player $\Diamond$ has no strategy to reach set Y (because of the way Y is constructed). The case of probabilistic location is also the same. Hence, the probability of reaching Y from ℓ is equal to zero. ◀

▶ **Lemma 7.** *Given a 2 variable $1\frac{1}{2}$ SSG $\mathcal{SG}$, and a desired set of target locations $\mathcal{F}$, we can compute the set of states from which player $\Diamond$ has a strategy to attain objective $\mathsf{Prob\_Reach}_{=1}(\mathcal{F})$.*

Proof. First, we construct the set Y from which player $\Diamond$ has a strategy to reach $\mathcal{F}$ with probability greater than 0 (using algorithm given in Lemma 6). From set Y, we will construct a set Z as follows,

1. Initialize: $Z_0 = Y$
2. Repeatedly remove locations ℓ from Z depending on their type, $Z_{i+1} = Z_i \setminus \ell$ until a fix-point is reached.
   a. If $\ell \in \mathcal{Q}_\bigcirc$, and has any enabled edge going out of the set Z_i.
   b. If $\ell \in \mathcal{Q}_\Diamond$, and has no enabled edge going into the set Z_i.

Now, we claim: a location $\ell \in Z$ if and only if player $\Diamond$ has a strategy to attain the objective $\mathsf{Prob\_Reach}_{=1}(\mathcal{F})$ from ℓ.

Let us consider the case that $\ell \notin Z$. We show that player $\Diamond$ does not have a strategy to reach a location in $\mathcal{F}$ with probability 1 from ℓ.

1. If $\ell \notin Y$. Then by Lemma 6, the probability of reaching $\mathcal{F}$ from ℓ is 0.
2. If $\ell \in Y \setminus Z$. Since $\ell \notin Z$, there exists some $i \geq 0$ such that $\ell \in Z_i \subseteq Y$, but $\ell \notin Z_{i+1} \subset Z_i$, and as $\ell \in Y$, atleast one outgoing edge of ℓ must be in Y.
   a. If $\ell \in \mathcal{Q}_{\bigcirc}$, then there exists an outgoing edge from ℓ to a node $\ell' \notin Z_i$.
   b. If $\ell \in \mathcal{Q}_{\Diamond}$, then all outgoing edges from ℓ are to nodes $\ell' \notin Z_i$.

Continuing backward from these nodes ℓ', we eventually reach nodes ℓ_{bad} such that $\ell_{bad} \notin Z_0 = Y$. $\ell_{bad} \notin Y$ implies that the probability of reaching $\mathcal{F}$ is 0. Since ℓ_{bad} is reachable from $\ell \in Y \setminus Z$, we conclude that the probability of reaching $\mathcal{F}$ from ℓ is not 1. The converse case, that is, if ℓ is a location such that player $\Diamond$ has no strategy to reach a location in $\mathcal{F}$ with probability 1, then $\ell \notin Z$ can be proved in a similar way. ◀

▶ **Theorem 8.** *The qualitative reachability problem is decidable for $1\frac{1}{2}$ SSG with two variables. Moreover, it is* EXPTIME *complete.*

Proof. We give the EXPTIME membership and hardness.

Membership. First, construct the region game graph from the given SSG (Appendix A.1). As seen in Lemma 6, a location (l, α) in the region graph is in $\mathcal{Q}_{\bigcirc}$ if and only if $l \in \mathcal{Q}_{\bigcirc}$; likewise, it is in $\mathcal{Q}_{\Diamond}$ if $l \in \mathcal{Q}_{\Diamond}$. Given the region game graph, we solve the qualitative reachability question using a backward fixpoint algorithm that iteratively refines the probability computation starting from the target locations. We know that the fixed point computation is polynomial time with respect to the size of the underlying graph (as this can be solved using BFS on the underlying graph). The size of the region graph (Appendix A.1) is exponential in the number of the variables V, when $|V| > 1$. Given the polytime algorithm for the fixed point, this problem is in EXPTIME.

Hardness. For the hardness we use the qualitative reachability problem of probabilistic timed automata (PTA) with two clocks [15]. A probabilistic timed automata (PTA) [17] is defined as $\mathsf{T} = (\mathcal{Q}, \ell_0, \mathcal{X}, \Delta, \Delta_{prob})$ where, $\mathcal{Q}$ is a finite set of locations, $\ell_0 \in \mathcal{Q}$ is the initial location, $\mathcal{X}$ is the finite set of real valued variables called clocks, Δ is a set of transitions of the form $(\ell, \varphi, R, \ell')$ with the usual semantics as timed automata transitions, and Δ_{prob} is the set of probabilistic transitions of the form $\big(\ell, \varphi, \mathsf{Dist}(2^{\mathcal{X}} \times \mathcal{Q})\big)$ where $\ell \in \mathcal{Q}$, φ is a clock constraint in the outgoing transitions from ℓ and $\mathsf{Dist}(2^{\mathcal{X}} \times \mathcal{Q})$ is a probability distribution which assigns probabilities to (reset set, target location) pairs (R, ℓ') on outgoing transitions $(\ell, \varphi, R, \ell')$ from ℓ. W.l.o.g, in the PTA, we replace probabilities with weights (and calculate probabilities from weights in the usual way). See Figure 2 for a PTA (on the left).

A $1\frac{1}{2}$ SSG with two variables can simulate a PTA $\mathsf{T} = (\mathcal{Q}, \ell_0, \{x, y\}, \Delta, \Delta_{prob})$ with two clocks x, y. We construct a $1\frac{1}{2}$ SSG $\mathcal{SG}$ with two variables x', y' corresponding to the two clocks x, y of the T. For each location $\ell \in \mathcal{Q}$ of the T we create location $\ell_{\Diamond} \in \mathcal{Q}_{\Diamond}$. The two clocks of the T are simulated using the two variables of the $1\frac{1}{2}$ SSG whose rate is 1 $\mathfrak{R}(\ell_{\Diamond})[x'] = \mathfrak{R}(\ell_{\Diamond})[y'] = 1$ in all locations from $\mathcal{Q}_{\Diamond}$. The transitions Δ of the T are added in the $1\frac{1}{2}$-SHG (by replacing ℓ with $\ell_{\Diamond}$, and replacing x, y with x', y'). It remains to add the probabilistic transitions Δ_{prob} to $\mathcal{SG}$. Consider $t = \big(\ell^i, \varphi, \mathsf{Dist}_i\big) \in \Delta_{prob}$. We add a new stochastic location $\ell_t \in \mathcal{Q}_{\bigcirc}$ such that, the rate of x', y' are zero at ℓ_t and add the following transitions in the $\mathcal{SG}$.

- $(\ell^i_{\Diamond}, \varphi, \emptyset, \ell_t)$ i.e., a transition from $\ell^i_{\Diamond}$ to the stochastic location ℓ_t with the same constraints of the transition t.
- For each pair $(R_j, \ell^j) \in \mathsf{Dist}(2^{\mathcal{X}} \times \mathcal{Q})$ in the probability distribution $(\ell^i, \varphi, \mathsf{Dist}(2^{\mathcal{X}} \times \mathcal{Q}))$ with weight w_j, we add the transition $(\ell_t, \top, R_j, \ell^j_{\Diamond})$ with the same weight w_j.

The probability incurred to go from ℓ_t to $\ell_\Diamond^j$ is given by $\frac{w_j}{\sum w_j}\int_0^\infty e^{-t}dt$, since an unbounded delay is allowed at ℓ_t. Hence, the probability in the SSG $\mathcal{SG}$ to go from $\ell_\Diamond^i$ to $\ell_\Diamond^j$ is $\frac{w_j}{\sum w_j}$ which is the same as the probability given by the PTA T. Since we preserve all probabilities, it is easy to check that, by solving the qualitative reachability of the constructed $1\frac{1}{2}$ SSG we solve the qualitative reachability of the T with two clocks. Hence the qualitative reachability of $1\frac{1}{2}$ SSG is EXPTIME-Hard. ◀

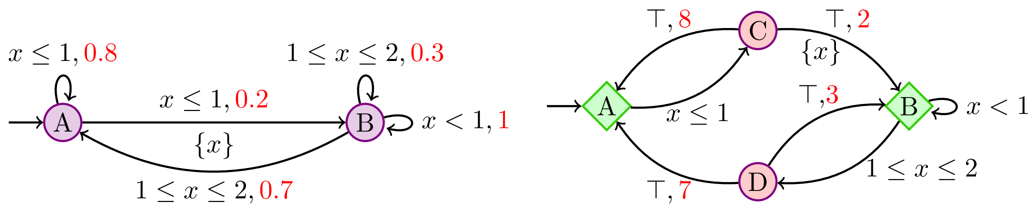

Figure 2 PTA to $1\frac{1}{2}$ SSG reduction (PTA is on left and $1\frac{1}{2}$-SSG on right). Probabilities (and weights for SSG) are in red color. From A, on $x \leq 1$, there is a reset free edge and a reset edge with probabilities 0.8 and 0.2. Likewise, from B, on $1 \leq x \leq 2$, there is a distribution (0.3, 0.7), while for $x \leq 1$, there is just one transition. The green diamond shaped nodes in the right are of Player $\Diamond$ and other nodes are stochastic.

We next show that the qualitative reachability problem becomes undecidable as soon as we have three variables. The proof goes via a reduction from the non-halting problem for Minsky two counter machines to the qualitative reachability problem for $1\frac{1}{2}$ SSG with three variables, where we have one stopwatch and two clocks.

Two-counter machine. A counter machine can be defined as a tuple $(L, \mathcal{C})$, where L is the finite state of instructions including the special instruction "HALT", and $\mathcal{C} = \{C_1, C_2\}$. The instruction can be any one of the following types,

1) (Increment the counter) $\ell_p : C_i := C_i + 1;$ goto ℓ_q; $\forall i \in \{1, 2\}$,
2) (Decrement the counter) $\ell_p : C_i := C_i - 1;$ goto ℓ_q; $\forall i \in \{1, 2\}$,
3) (Checking zero) $\ell_p :$ if $(C_i = 0)$ then goto ℓ_q else goto ℓ_r; $\forall i \in \{1, 2\}$,
4) (Halting instruction) $\ell_q :$ HALT;

Where, $C_i \in \mathcal{C}$, $\ell_p, \ell_q, \ell_r \in L$. A configuration of a two-counter machine is a tuple (ℓ, m, n) where, $\ell \in L$ and $m, n \in \mathbb{N} \cup \{0\}$ represents the current value of the counters c_1, c_2 respectively. A two-counter machine starts from the initial configuration $(\ell_0, 0, 0)$. A run of a two-counter machine is a sequence of configurations $(\ell_0, 0, 0) \to (\ell_1, m_1, n_1) \to (\ell_2, m_2, n_2) \cdots$. The transition between two configurations depends on the instruction of the first configuration. We say a run is halting if it is finite and ends with an HALT instruction, in fact the two-counter machine never progresses beyond a HALT instruction. The halting problem of a two-counter machine is checking if a given two-counter machine has a halting run or not. It is well-known that two-counter machine is Turing complete and the halting problem for two-counter machine is undecidable [19].

▶ **Theorem 9.** *The qualitative reachability problem for $1\frac{1}{2}$ SSG is $\prod_1^0$ hard with one stopwatch variable and two clocks.*

We prove the $\prod_1^0$ hardness of qualitative reachability for $1\frac{1}{2}$ SSG with one stopwatch and two clocks by reducing it to the non-halting problem for two counter machines.

Reduction to reachability in $1\frac{1}{2}$ SSG

Given a two counter machine $\mathcal{M}$, we construct a one and half player SSG $G(\mathcal{M})$, where $\Diamond$ has a strategy to reach some desired location (denoted as pink square nodes labeled T) with probability =1 if and only if $\mathcal{M}$ does not halt.

The game graph $G(\mathcal{M})$ uses 3 variables : a stopwatch x and two clocks y, z. The values i, j of the counters C_1, C_2 are encoded in the variable x as $\frac{1}{2^i 3^j}$. y, z are used as auxiliary clock variables. The stopwatch x has rate $r_x = 1$ in all the stochastic nodes. The $\Diamond$ nodes where x has rate $r_x = 0$ are colored blue. The graph $G(\mathcal{M})$ has one gadget per instruction of the two counter machine. By adjoining the gadgets appropriately, depending on the instructions of $\mathcal{M}$, we obtain the complete game graph $G(\mathcal{M})$.

Decrement C_1

Let us begin with the gadget for the instruction ℓ_i: $C_1 := C_1 - 1$; goto ℓ_j. Figure 3 depicts the gadget for decrementing counter C_1.

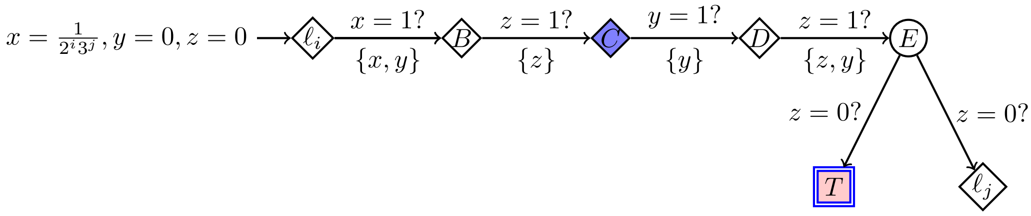

Figure 3 Gadget Dec C_1.

▶ **Lemma 10.** *On entering the gadget Dec C_1 in Figure 3, in node ℓ_i with values $x = \frac{1}{2^i 3^j}$, $y = z = 0$, the node ℓ_j is reached with probability $\frac{1}{2}$ with $x = \frac{1}{2^{i-1} 3^j}, y = z = 0$ and the target node T is reached with probability $\frac{1}{2}$.*

Proof. The proof of correctness of Lemma 10 can be seen by examining the functioning of the gadget: On entry to node ℓ_i, we have the values $x = \frac{1}{2^i 3^j}$, $y = z = 0$. A time $1 - \frac{1}{2^i 3^j}$ is spent at location ℓ_i, obtaining $x = y = 0$,$z = 1 - \frac{1}{2^i 3^j}$ on entry into node B.

Time equal to $\frac{1}{2^i 3^j}$ is spent at node B, obtaining valuations $x = y = \frac{1}{2^i 3^j}$,$z = 0$ on entry to node C. Subsequently time equal to $1 - \frac{1}{2^i 3^j}$ is spent at node C, obtaining valuations $x = \frac{1}{2^i 3^j}$,$y = 0$, $z = 1 - \frac{1}{2^i 3^j}$ on entry to node D. The value of x remains constant during this transition to node D as the node is shaded blue. Then the constraint forces to spend time equal to $\frac{1}{2^i 3^j}$ at node D and reaches the stochastic node E with valuation $x = \frac{2}{2^i 3^j} = \frac{1}{2^{i-1} 3^j}$,$y = z = 0$. Through the stochastic node E, ℓ_j and T can be reached with probability $= \frac{1}{2}$ each, with the clock valuation $x = \frac{1}{2^{i-1} 3^j}$,$y = z = 0$ as required. ◀

Increment C_1

Next, let us look at the instruction for incrementing. Figure 4 depicts the gadget for incrementing counter C_1 simulating the instruction $\ell_i : C_1 := C_1 + 1$, goto ℓ_j.

▶ **Lemma 11.** *On entering the gadget Inc C_1 in Figure 4, in node ℓ_i with values $x = \frac{1}{2^i 3^j}$, $y = z = 0$, the node ℓ_j is reached with probability $\frac{1}{2}$ and $x = \frac{1}{2^{i+1} 3^j}, y = z = 0$. The target node is reached with probability $\frac{1}{2}$.*

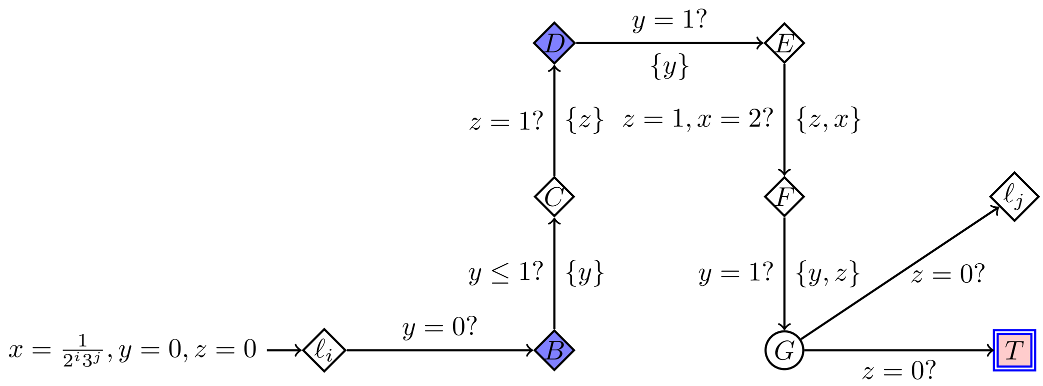

Figure 4 Gadget Inc C_1.

Proof. The proof of Lemma 11 can be seen by examining the functioning of the gadget in Figure 4. The node ℓ_i is entered with $x = \frac{1}{2^i 3^j}, y = z = 0$. The node B is entered with the same values but the stopwatch x is now stopped because the node is blue shaded. An amount of time $t \leq 1$ is spent at B obtaining $x = \frac{1}{2^i 3^j}, y = 0, z = t$ on entering node C. We spend $1 - t$ time at node C entering node D with valuations $x = 1 - t + \frac{1}{2^i 3^j}, y = 1 - t, z = 0$. Now t time is spent at node D with x paused (the node D is blue shaded) and hence we enter node E with valuations $x = 1 - t + \frac{1}{2^i 3^j}, y = 0, z = t$. Now to satisfy the first guard i.e., $z = 1$ for leaving E we need to spend $1 - t$ time at E which would mean x would now be $x = 2 - 2t + \frac{1}{2^i 3^j}$. To move to F we also need $x = 2$ which would imply that $2t = \frac{1}{2^i 3^j}$. Then we enter node F with $x = z = 0, y = 1 - t$ where $t = \frac{1}{2^{i+1} 3^j}$. We spend t time at F and through the stochastic node G. We enter node ℓ_j and target node T with probability $\frac{1}{2}$ each, with clock values $x = \frac{1}{2^{i+1} 3^j}, y = 0, z = 0$ as required. ◀

Zero Check for C_1

Next, we look at the zero check instruction, ℓ_i : if $C_1 = 0$, then goto ℓ_j, else goto ℓ_k. Figure 5 depicts the gadget for zero check of counter C_1.

▶ **Lemma 12.** *On entering the gadget Zero Check C_1 in Figure 5, in node ℓ_i with values $x = \frac{1}{2^i 3^j}$, $y = z = 0$, player $\Diamond$ has a strategy to reach the nodes l_j and the target T with probability $\frac{1}{2}$ each if and only if $i = 0$, that is, the value of $x = \frac{1}{3^j}$ on entry at ℓ_i. Similarly, player $\Diamond$ has a strategy to reach the nodes l_k and the target node T with probability $\frac{1}{2}$ each if and only if $i \neq 0$, that is, the value of $x = \frac{1}{2^i 3^j}$, $i > 0$ on entry at ℓ_i.*

Proof. The proof of Lemma 12 follows by examining the functioning of the gadget in Figure 5. The $\Diamond$ player has two possible strategies at node ℓ_i: to goto Z or NZ. The correct strategy is to go to Z when $x = \frac{1}{3^j}$ and to go to NZ when $x = \frac{1}{2^i 3^j}$, $i > 0$ at node ℓ_i. No time is elapsed at node ℓ_i. Assume the strategy is to go to Z, no time is elapsed at Z, and the stochastic node SZ is entered. Then from the stochastic node SZ, with probability $\frac{1}{2}$ each, the node ℓ_j (corresponding to the next instruction) and the gadget *Make* C_2 0 is entered. This gadget is in Figure 6. From the $\Diamond$ node SZ, we enter the starting location A of in Figure 6, and from the same node A, we can move to the location T of Figure 5 if $x = 1, z = 0$. If the strategy of choosing Z was indeed the correct one, then, decrementing C_2 some number of times would give $x = 1$. In this case, after some number of iterations of the gadget in Figure 6, we reach T in Figure 5 with probability $\frac{1}{2}$. Note that the gadget in Figure 6 has no stochastic nodes, so on successful completion we reach T with $x = 1$ incurring probability $\frac{1}{2}$.

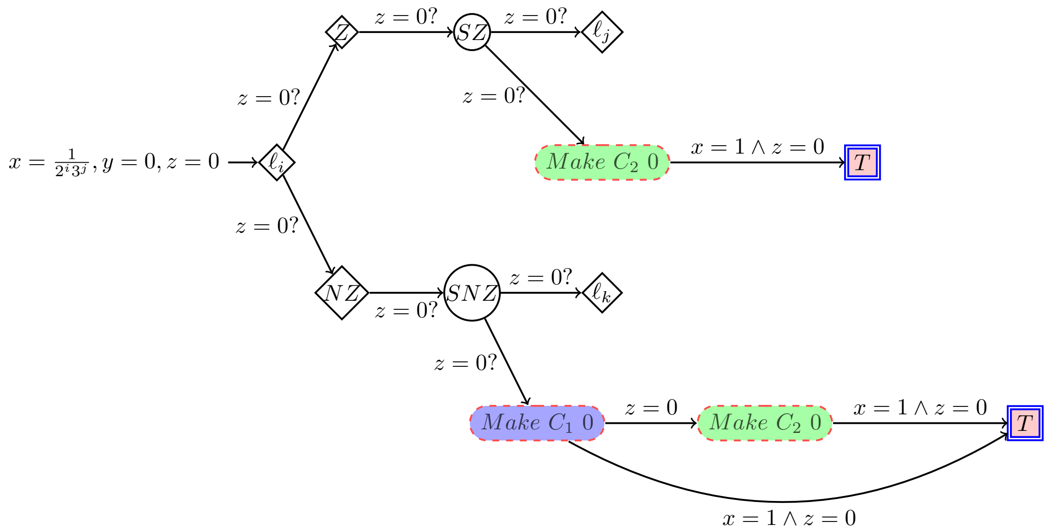

Figure 5 Gadget Zero Check C_1.

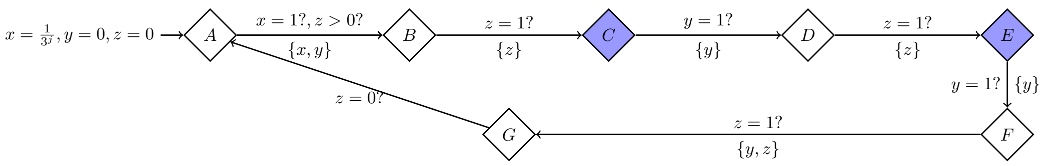

Figure 6 Make C_2 0.

Note that if the decision of choosing node Z is incorrect, that is, if $x = \frac{1}{2^i 3^j}$, $i > 0$, then the value of x will exceed 1 and we will be stuck in the gadget $Make\ C_2\ 0$.

The decision of choosing NZ from ℓ_i is similar : in this case, the gadget $Make\ C_1\ 0$ has to be visited at least once, before the target T is reached. $Make\ C_1\ 0$ can be obtained similar to $Make\ C_2\ 0$, and it also has no stochastic nodes. In particular, if we have $x = \frac{1}{2^i 3^j}$, $i > 0$, then we must iterate $Make\ C_1\ 0$ i times exactly, and in case $x < 1$ we iterate $Make\ C_2\ 0$ j times. If we iterate $Make\ C_1\ 0 < i$ ($> i$) times, we will get stuck in $Make\ C_1\ 0$ ($Make\ C_2\ 0$). Otherwise, we will reach T in Figure 5 with probability $\frac{1}{2}$. ◀

Gadgets for incrementing, decrementing and zero check gadget for C_2 are similar to the seen gadgets. We now argue that $\Diamond$ has a strategy to reach a target location T with probability 1 if and only if the two counter machine does not halt.

▶ **Theorem 13.** *Given a two counter machine $\mathcal{M}$, Player $\Diamond$ has a strategy to reach a target node T with probability 1 in the constructed game graph $G(\mathcal{M})$ if and only if the two counter machine $\mathcal{M}$ does not halt.*

Proof. The proof of Theorem 13 is obtained by putting together the lemmas above. Since HALT is also an instruction in the two counter machine, we have a gadget with location labeled *Halt* corresponding to the HALT instruction. From this node, there is no outgoing edge and hence this is a dead state. The rest of the SSG is made by appropriately stringing together the gadgets as per the design of the two counter machine. We claim that player $\Diamond$ has a strategy to reach a target location T with probability 1 if and only if he simulates all the instructions correctly and if and only if $\mathcal{M}$ does not halt.

Consider the first instruction of the two counter machine that is executed. By using all lemmas so far, whatever this instruction might be, on a correct simulation, the SSG enters a target T in some gadget with probability $\frac{1}{2}$ and continues execution from the state as specified by the instruction with probability $\frac{1}{2}$.

Therefore the probability of reaching a target T is $P_{total} = \frac{1}{2}$ (reaching T in the first gadget) $+ \frac{1}{2}P_{rest}$ where P_{rest} is the probability of reaching the target T node when continuing the simulation of $\mathcal{M}$ after the current instruction. Recall that each gadget corresponding to instructions went with probability $\frac{1}{2}$ to the next instruction ℓ_j to be simulated; the $\frac{1}{2}$ in the term $\frac{1}{2}P_{rest}$ comes from there.

We now apply the above process for finding P_{total} to find P_{rest} recursively for the next executed instruction from ℓ_j and we get $P_{total} = \frac{1}{2} + \frac{1}{2}(\frac{1}{2} + \frac{1}{2}P_{rest})$ where P_{rest} is the probability of reaching T when continuing execution from the subsequent instruction reached.

This above processes can be recursively repeated. If $\mathcal{M}$ reaches HALT, then we will reach a gadget where with probability $\frac{1}{2}$ we reach T and with probability $\frac{1}{2}$ we reach HALT. In this case, the above summation will add up to < 1 : $P_{total} = \sum_{n=1}^{I} 2^{-n} < 1$ where I is the total number of instructions executed to reach HALT. However if $\mathcal{M}$ does not halt, then the run is not finite, and we keep going one gadget after the other. Then we get the infinite sum $P_{total} = \sum_{n=1}^{\infty} 2^{-n} = 1$. Hence player $\Diamond$ will reach the target node T with probability 1 if and only if the two counter machine does not halt. ◀

4 Conclusion

In this paper, we have proposed stochastic stopwatch games, an extension of stochastic timed games and proved decidability and undecidability results for $1\frac{1}{2}$-stochastic stopwatch games. This work leads us to further open problems for e.g., how does the undecidability result change if we consider time-bounded qualitative reachability or when we consider only $\frac{1}{2}$-stochastic stopwatch games.

References

1. S. Akshay, Patricia Bouyer, Shankara Narayanan Krishna, Lakshmi Manasa, and Ashutosh Trivedi. Stochastic Timed Games Revisited. In Piotr Faliszewski, Anca Muscholl, and Rolf Niedermeier, editors, *41st International Symposium on Mathematical Foundations of Computer Science (MFCS 2016)*, volume 58 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:14, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. `doi:10.4230/LIPIcs.MFCS.2016.8`.
2. R. Alur, C. Courcoubetis, N. Halbwachs, T.A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138(1):3–34, February 1995. `doi:10.1016/0304-3975(94)00202-t`.
3. 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*, pages 209–229. Springer Berlin Heidelberg, 1993. `doi:10.1007/3-540-57318-6_30`.
4. Rajeev Alur and David L Dill. A theory of timed automata. *Theoretical computer science*, 126(2):183–235, 1994.
5. Eugene Asarin, Oded Maler, Amir Pnueli, and Joseph Sifakis. Controller synthesis for timed automata. *IFAC Proceedings Volumes*, 31(18):447–452, 1998. `doi:10.1016/S1474-6670(17)42032-5`.
6. C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous-time markov chains. *IEEE Transactions on Software Engineering*, 29(6):524–541, June 2003. `doi:10.1109/tse.2003.1205180`.

7 Christel Baier, Nathalie Bertrand, Patricia Bouyer, Thomas Brihaye, and Marcus Größer. Probabilistic and topological semantics for timed automata. In *FSTTCS 2007: Foundations of Software Technology and Theoretical Computer Science*, pages 179–191. Springer Berlin Heidelberg, 2007. doi:10.1007/978-3-540-77050-3_15.

8 Christel Baier, Nathalie Bertrand, Patricia Bouyer, Thomas Brihaye, and Marcus Grosser. Almost-sure model checking of infinite paths in one-clock timed automata. In *2008 23rd Annual IEEE Symposium on Logic in Computer Science*. IEEE, June 2008. doi:10.1109/lics.2008.25.

9 Nathalie Bertrand, Patricia Bouyer, Thomas Brihaye, Quentin Menet, Christel Baier, Marcus Groesser, and Marcin Jurdzinski. Stochastic timed automata. *Logical Methods in Computer Science*, 10(4), December 2014. doi:10.2168/lmcs-10(4:6)2014.

10 Patricia Bouyer and Vojtech Forejt. Reachability in stochastic timed games. In *ICALP*, pages 103–114. Springer, 2009. doi:10.1007/978-3-642-02930-1_9.

11 Thomas Brihaye, Véronique Bruyère, and Jean-François Raskin. On model-checking timed automata with stopwatch observers. *Information and Computation*, 204(3):408–433, 2006. doi:10.1016/j.ic.2005.12.001.

12 Thomas Brihaye, Laurent Doyen, Gilles Geeraerts, Joel Ouaknine, Jean-Francois Raskin, and James Worrell. Time-bounded reachability for monotonic hybrid automata: Complexity and fixed points. In *Automated Technology for Verification and Analysis*, pages 55–70. Springer International Publishing, 2013. doi:10.1007/978-3-319-02444-8_6.

13 Franck Cassez and Kim Larsen. The impressive power of stopwatches. In *CONCUR 2000 – Concurrency Theory*, pages 138–152. Springer Berlin Heidelberg, 2000. doi:10.1007/3-540-44618-4_12.

14 Thomas A. Henzinger, Peter W. Kopke, Anuj Puri, and Pravin Varaiya. What's decidable about hybrid automata? *Journal of Computer and System Sciences*, 57(1):94–124, August 1998. doi:10.1006/jcss.1998.1581.

15 Marcin Jurdzinski, Jeremy Sproston, and Francois Laroussinie. Model checking probabilistic timed automata with one or two clocks. *Logical Methods in Computer Science*, 4(3), September 2008. doi:10.2168/lmcs-4(3:12)2008.

16 Shankara Narayanan Krishna, Umang Mathur, and Ashutosh Trivedi. Weak singular hybrid automata. In *Lecture Notes in Computer Science*, pages 161–175. Springer International Publishing, 2014. doi:10.1007/978-3-319-10512-3_12.

17 Marta Kwiatkowska, Gethin Norman, Roberto Segala, and Jeremy Sproston. Verifying quantitative properties of continuous probabilistic timed automata. In *CONCUR 2000 – Concurrency Theory*, pages 123–137. Springer Berlin Heidelberg, 2000. doi:10.1007/3-540-44618-4_11.

18 Marta Kwiatkowska, Gethin Norman, Roberto Segala, and Jeremy Sproston. Automatic verification of real-time systems with discrete probability distributions. *Theoretical Computer Science*, 282(1):101–150, June 2002. doi:10.1016/s0304-3975(01)00046-9.

19 Marvin L. Minsky. *Computation: Finite and Infinite Machines*. Prentice-Hall, Inc., USA, 1967.

20 Nima Roohi and Mahesh Viswanathan. Time-bounded reachability for initialized hybrid automata with linear differential inclusions and rectangular constraints. In *Lecture Notes in Computer Science*, pages 191–205. Springer International Publishing, 2014. doi:10.1007/978-3-319-10512-3_14.

A Singular Stopwatch Automata with Two Stopwatches

Let 2-$\mathcal{H}$ be a singular stopwatch automata with two stopwatch variables $V = \{x, y\}$. When the current location $l \in \mathcal{Q}$ is known, we represent the rate of $c \in V$ as $r_c = \mathfrak{R}(l)[c]$ i.e., $r_x, r_y \in \{0, 1\}$ for $x, y \in V$ respectively. Let $c_{\max}$ be the maximum constant used in any of the guards of 2-$\mathcal{H}$. Note that, unlike clock regions defined in [4], we can not directly define regions for stopwatches, as the successor region depends on the current rate of change r_x, r_y of stopwatch x and y respectively. So, we need to define regions such that they agree with the successor function with different rates.

A.1 Regions and Region Automaton of 2-$\mathcal{H}$

In this section, we define a region abstraction, which is reachability preserving in the sense of the region automaton from [4]. Intuitively, a region $\Re$ is a collection of infinitely many valuations $\nu \in \mathbb{R}^2_{\geq 0}$, having some good properties preserving reachability. The number of regions must be finite.

▶ **Definition 14.** *Let $\varphi \in \mathsf{rect}(V)$ be a constraint. A region $\Re$ is compatible with φ if and only if for all valuations $\nu \in \Re$, either $\nu \models \varphi$ or $\nu \models \neg\varphi$.*

We define a map $\mathsf{Res} : \Re \to \Re$ that maps a region $\Re$ to the region $\mathsf{Res}(\Re)$ obtained from $\Re$ by assigning value 0 to all variables which were reset to 0.

▶ **Definition 15.** *A set of regions $\Re$ is compatible with resets (or with the map Res) if whenever a valuation $\nu' \in \Re'$ is reachable from a valuation $\nu \in \Re$ after a reset, then $\Re'$ is reachable from any $\nu \in \Re$ by the same reset.*

Construction of Regions $\mathsf{Reg}(V)$

We first construct a set of regions for 2-$\mathcal{H}$ that are compatible with resets and guards. For $z \in \{x, y\}$, we define the set of intervals

$$\mathcal{I}_z = \big\{[c] \mid 0 \leq c \leq c_{\max}\big\} \cup \big\{(c, c+1) \mid 0 \leq c < c_{\max}\big\} \cup \big\{(c_{\max}, \infty)\big\}$$

For a variable x and its current valuation $\nu[x]$, we use $I_x \in \mathcal{I}_x$ to represent its current interval, and $\mathsf{fract}(\nu[x])$ to denote the fractional part of $\nu[x]$. For example, if we have a variable x with valuation $\nu[x] = 4.5$ then it's interval $I_x = (4, 5) \in \mathcal{I}_x$ and $\mathsf{fract}(\nu[x]) = 0.5$. Let us define a relation $\sim_\Re$ between two valuation ν_1 and ν_2 such that $\nu_1 \sim_\Re \nu_2$ if and only if,

- $\forall z \in V$, $I_z \in \mathcal{I}_z$ $\nu_1[z] \in I_z \iff \nu_2[z] \in I_z$
- $x, y \in V$, $\mathsf{fract}(\nu_1[x]) \leq \mathsf{fract}(\nu_1[y]) \iff \mathsf{fract}(\nu_2[x]) \leq \mathsf{fract}(\nu_2[y])$

Clearly, the relation $\sim_\Re$ is an equivalence relation and this forms a finite partitioning of $\mathbb{R}^2_{\geq 0}$. Let us define such a partition by $\alpha = (I_x, I_y, \prec, \mathsf{Rel})$ where, I_x, I_y represents the interval of clocks x and y respectively, and $\prec$ is a total pre-order on the set $\mathsf{Rel} = \big\{x \in V \mid I_x \in \big\{(c, c+1) \mid 0 \leq c < c_{\max}\big\}\big\}$. Assume $\Re_\alpha$ represents the region defined by α.

We define $\mathsf{Reg}(V)$ to be the set of all such partitions $\Re_\alpha$. For each valuation $\nu \in \mathbb{R}^2_{\geq 0}$ of variables, the unique element $\Re$ of $\mathsf{Reg}(V)$ that contains ν is called a region, denoted $[\nu]$.

Successors of a Region

We define the successors of a region $\Re$, $\mathsf{Succ}_{(r_x,r_y)}(\Re) \subseteq \mathsf{Reg}(V)$, in the following natural way: $r_x, r_y \in \{0, 1\}, x, y \in V$,

$$\Re' \in \mathsf{Succ}_{(r_x,r_y)}(\Re) \text{ if } \exists \nu \in \Re, \exists t \in \Re \text{ such that } [\nu + (r_x, r_y)t] = \Re' \tag{1}$$

By $\nu + (r_x, r_y)t$, we represent the valuation $\big(\nu[x] + r_x \cdot t, \nu[y] + r_y \cdot t\big)$.

A finite partition $\mathsf{Reg}(V)$ of $\mathbb{R}^2_{\geq 0}$ is a *set of regions* whenever the following condition holds:

$$\Re' \in \mathsf{Succ}_{(r_x,r_y)}(\Re) \text{ if and only if } \forall \nu \in \Re, \exists t \in \Re \text{ such that } [\nu + (r_x, r_y)t] = \Re' \tag{2}$$

We now consider resets. Formally, we have

$$\Re' \in \mathsf{Res}(\Re) \rightarrow \forall \nu \in \Re, \exists \nu' \in \Re' \text{ such that } \nu' \in \mathsf{Res}(\nu) \tag{3}$$

Now, we will show that $\mathsf{Reg}(V)$ follows the conditions (2) and (3) defined above.

▶ **Lemma 16.** *For all partitions $\Re \in \mathsf{Reg}(V)$, $\Re' \in \mathit{Succ}_{(r_x,r_y)}(\Re)$ if and only if for all valuation $\nu \in \Re$, there exists $t \in \mathbb{R}_{\geq 0}$ such that $[\nu + (r_x, r_y)t] = \Re'$*

Proof. Consider a partition $\Re_\alpha$ defined by $\alpha = (I_x, I_y, \prec, \mathsf{Rel})$.

If $I_x = I_y = (c_{max}, \infty)$, then $\mathsf{Succ}_{(r_x,r_y)}(\Re_\alpha) = \{\Re_\alpha\}$ because, for all $\nu \in \Re_\alpha$, for all $t \in \mathbb{R}_{\geq 0}$, for $r_x, r_y \in \{0, 1\}$, $\nu + (r_x, r_y)t \in \Re_\alpha$.

If $r_x = r_y = 0$ then, $\mathsf{Succ}_{(r_x,r_y)}(\Re_\alpha) = \{\Re_\alpha\}$, for all $\Re_\alpha$.

If $\mathsf{Succ}_{(r_x,r_y)}(\Re_\alpha) \neq \{\Re_\alpha\}$, then there exists atleast one another region in $\mathsf{Succ}_{(r_x,r_y)}(\Re_\alpha)$ that is different from $\Re_\alpha$. Let $\Re_\beta$ denote the region that is closest to region to $\Re_\alpha$ such that, $\Re_\beta \in \mathsf{Succ}_{(r_x,r_y)}(\Re_\alpha)$, and for all $\nu \in \Re_\alpha$, for all $t \in \mathbb{R}_{\geq 0}$, if $\nu + (r_x, r_y)t \notin \Re_\alpha$, then $\exists t' \leq t$ such that $\nu + (r_x, r_y)t' \in \Re_\beta$. Assume that, such a region $\Re_\beta$ is defined by $\beta = (I'_x, I'_y, \prec', \mathsf{Rel}')$ and characterized as follows:

Let $Z = \{z \in V \mid I_z \text{ is of the form } [c]\}$, i.e., Z is the set of clocks with integer value.

1. If $Z \neq \emptyset$ and $r_x = r_y = 1$, (if $x \notin Z$ then $y \in Z$ and vice versa)

$$I'_z = \begin{cases} I_z & \text{if } z \notin Z, \\ (c, c+1) & \text{if } z \in Z, I_z = [c], \text{ and } 0 \leq c < c_{\max} \\ (c_{\max}, \infty) & \text{if } z \in Z \text{ and } I_z = [c_{\max}] \end{cases}$$

and, $x \prec' y$ if $I_x = [c], I'_x = (c, c+1)$ with $0 \leq c < c_{\max}$ and I'_y is of the form $(d, d+1)$, $\mathsf{Rel}' = \{x, y\}$.

2. If $Z \neq \emptyset$ and atleast one of r_x, r_y is 0, (if $x \notin Z$ then $y \in Z$ and vice versa, and if $r_x = 1$ then $r_y = 0$ and vice versa.)

$$I'_z = \begin{cases} I_z & \text{if } r_z = 0, \\ [c+1] & \text{if } z \notin Z, I_z = (c, c+1), r_z = 1, \text{ and } 0 \leq c < c_{\max} \\ (c, c+1) & \text{if } z \in Z, I_z = [c], r_z = 1 \text{ and } 0 \leq c < c_{\max} \\ (c_{\max}, \infty) & \text{if } z \in Z, I_z = [c_{\max}], \text{ and } r_z = 1 \end{cases}$$

and, $\mathsf{Rel}' = \{x, y\}, x \prec' y$ if $r_x = 1$ and $x \in Z, y \notin Z$. If ($r_x = 0$ and $x \in Z$) or ($r_x = 1$ and $x \notin Z$) or if $x, y \in Z$, then $\mathsf{Rel}' = \emptyset$.

3. If $Z = \emptyset$ and $r_x = r_y = 1$. Let M denote the set of variables with the maximum fractional part, whose interval is of the form $(c, c+1)$ for $0 \leq c < c_{\max}$. Then,

$$I'_z = \begin{cases} I_z & \text{if } z \notin M, \\ [c+1] & \text{if } z \in M \text{ and } I_z = (c, c+1) \text{ with } 0 \leq c < c_{\max} \end{cases}$$

One variable moves to an integer value, or both variables are in $(c_{\max}, \infty)$ thus, $\mathsf{Rel}' = \emptyset$.

4. If $Z = \emptyset$ and atleast one of r_x, r_y is 0. Then,

$$I'_z = \begin{cases} I_z & \text{if } r_z = 0, \\ [c+1] & \text{if } z \in M, r_z = 1, \text{ and } I_z = (c, c+1) \text{ with } 0 \le c < c_{\max} \\ I_z & \text{if } z \notin M, r_z = 1, \text{ and } I_z = (c, c+1) \text{ with } 0 \le c < c_{\max} \end{cases}$$

and, $x \prec' y$ is same as $x \prec y$ when $r_x = r_y = 0$. Otherwise, one of the variables gets an integer value, and hence $\mathsf{Rel}' = \emptyset$.

We now claim that,

$$\forall \nu \in \Re_\alpha, \exists t \in \mathbb{R}_{\ge 0} \text{ such that } \nu + t \in \Re_\beta$$

Let ν be a valuation in $\Re_\alpha$.

1. If $Z \neq \emptyset$ and $r_x = r_y = 1$. Let $\tau = \min\{1 - \mathsf{fract}(\nu[z]) \mid I_z \text{ is of the form } (c, c+1)\}$. Then $\nu + (1,1)\frac{\tau}{2}$ is in the region R_β.

2. If $Z \neq \emptyset$ and atleast one of r_x, r_y is 0.

  (i) If $x \in Z, y \notin Z$ and $r_x = 1$, then pick $\tau = \mathsf{fract}(\nu[y])$. Then $\nu + (1,0)\frac{\tau}{2}$ is in the region $\Re_\beta$.

  (ii) If $r_x = 0$ and $x \in Z$, then pick $\tau = 1 - \mathsf{fract}(\nu[y])$. Then $\nu + (0,1)\tau$ is in the region $\Re_\beta$.

  (iii) If $r_x = 1$ and $x \notin Z$, then pick $\tau = 1 - \mathsf{fract}(\nu[x])$. $\nu + (1,0)\tau$ is in the region $\Re_\beta$.

  (iv) If $x, y \in Z$, and $r_x = 1$, then pick $\tau = 0.5$. Then $\nu + (1,0)\tau$ is in the region $\Re_\beta$.

3. If $Z = \emptyset$ and $r_x = r_y = 1$.
Pick the variable $z \in M$. Let $\tau = 1 - \mathsf{fract}(\nu[z])$. Then $\nu + (1,1)\tau$ is in the region $\Re_\beta$.

4. If $Z = \emptyset$ and atleast one of r_x, r_y is 0.
If $r_x = 1$ and $r_y = 0$. Pick $\tau = 1 - \mathsf{fract}(\nu[x])$. Then $\nu + (1,0)\tau$ is in the region $\Re_\beta$.

Thus we obtain that $\Re_\beta \in \mathsf{Succ}_{(r_x,r_y)}(\Re_\alpha)$ is the closest successor of $\Re_\alpha$. Inducting on $\Re_\beta$, we get the closest successor of $\Re_\beta$, which is also a successor of $\Re_\alpha$, 2 steps away, and so on. We write $\Re_\alpha \xrightarrow{n} \Re^n_\alpha$ if $\Re^n_\alpha$ is the nth closest successor of $\Re_\alpha$ with respect to some choice of rates (r_x, r_y). This clearly means that there is a sequence of regions $\Re^0_\alpha, \Re^1_\alpha, \Re^2_\alpha, \dots, \Re^n_\alpha$ such that $\Re^0_\alpha = \Re_\alpha$, and $\Re^{i+1}_\alpha$ is the closest successor of $\Re^i_\alpha$ for all $1 \le i < n$.

In this way, we can find all successors $\Re'_\alpha$ of $\Re_\alpha$ such that $\Re'_\alpha \in \mathsf{Succ}_{(r_x,r_y)}(\Re_\alpha)$ if and only if for all $\nu \in \Re_\alpha$ there exists some $t \in \mathbb{R}_{\ge 0}$ such that $\nu + (r_x, r_y)t \in \Re'_\alpha$. Hence, $\mathsf{Reg}(V)$ is indeed a set of regions partitioning $\mathbb{R}^2_{\ge 0}$. ◀

Given two valuations $\nu_1, \nu_2 \in \Re_\alpha$ for some region $\Re_\alpha$, we say that ν_1 and ν_2 are equivalent if they lie in the same region, i.e., $[\nu_1] = [\nu_2]$.

Reg(V) compatible with resets and guards

▶ **Lemma 17.** $\mathsf{Reg}(V)$ *is compatible with the guards* φ *and with the resets* Res.

Proof.

(1) Let $\Re' \in \mathsf{Res}(\Re)$. Consider $\nu_1, \nu_2 \in \Re$, i.e., $[\nu_1] = [\nu_2]$. Clearly, $\nu_1[x]$ and $\nu_2[x]$ lie in the same interval; same with $\nu_1[y]$ and $\nu_2[y]$. If the operation Res resets x, then $\mathsf{Res}(\nu_1) = \big(0, \nu_1[y]\big)$ and $\mathsf{Res}(\nu_2) = \big(0, \nu_2[y]\big)$. Since $\nu_1[y]$ and $\nu_2[y]$ are in the same interval, we have $[\mathsf{Res}(\nu_1)] = [\mathsf{Res}(\nu_2)]$. Similar results are obtained when y is reset, or when both x, y are reset.

(2) Let $[\nu_1] = [\nu_2]$ be valuations in the same region $\Re$. Let φ be a guard. The result can be proved by structural induction on φ. If φ is atomic of the form $x \sim c$, clearly, $\nu_1 \models \varphi$ if and only if $\nu_2 \models \varphi$, since ν_1 and ν_2 are equivalent. Assume for guards of size $\leq n-1$. It can be seen that the inductive hypothesis can be easily extended to guards of size n. Thus, $\mathsf{Reg}(V)$ is a finite set of regions compatible with guards and resets, partitioning $\mathbb{R}^2_{\geq 0}$. ◀

Hence, we can use the region abstraction for the above set of regions to obtain a region automaton $\mathsf{Reg}(2\text{-}\mathcal{H})$ capturing the untimed language of 2-$\mathcal{H}$. The set of states of such a region automaton is the set $\mathcal{Q} \times \mathsf{Reg}(V)$, where $\mathcal{Q}$ is the set of locations of 2-$\mathcal{H}$. The initial location of $\mathsf{Reg}(2\text{-}\mathcal{H})$ is $\big(l_0, (0,0)\big)$ where $l_0 \in \mathcal{Q}_0$ is the initial location of 2-$\mathcal{H}$. The transitions of $\mathsf{Reg}(2\text{-}\mathcal{H})$ are defined as $(l, \Re) \xrightarrow{a} (l', \Re')$ if and only if there is a region $\hat{\Re}$ and a transition from l to l' on $(\varphi, a, \mathsf{Res})$ in 2-$\mathcal{H}$ such that,

(1) $\hat{\Re} \in \mathsf{Succ}_{(r_x, r_y)}(\Re)$. ($r_x, r_y$ are the rates at the location l),
(2) For all $\nu \in \hat{\Re}$, $\nu \models \varphi$, and
(3) $\mathsf{Res}(\hat{\Re}) = \Re'$

The final states of the region automaton are the states $(f, \Re)$ such that f is a final location of 2-$\mathcal{H}$. It can be seen that the language accepted by this region automaton is indeed the untimed counterpart of $\mathcal{L}(2\text{-}\mathcal{H})$. We thus have, the following result.

▶ **Theorem 18.** *The region automaton construction for singular stopwatch automata* 2-$\mathcal{H}$ *with two stopwatches is a correct abstraction. For each run* ρ *in* 2-$\mathcal{H}$ *from an initial state* (l_0, ν_0) *to a state* (l_n, ν_n)*, if and only if we have a run* ρ' *in* $\mathsf{Reg}(2\text{-}\mathcal{H})$ *from* $(l_0, (0,0))$ *to* $(l_n, \Re_n)$ *such that* $\nu_n \in \Re_n$*.*

The proof is straightforward since in each step, we obtain a new region which is compatible with the constraints and resets of the transition taken.

However, this does not extend to 3 variables, as shown below.

A.2 Problem in extending the region construction to 3 variables

It is an interesting exercise to note where the above region construction fails if we try to extend it to 3 variables with at least one stopwatch. In a nutshell, the problem arises in defining the successor regions of $\Re$ i.e., in defining $\Re_\beta$. The above construction only works for 2 variables and fails if we try to extend the same to 3 variables. The following example helps to illustrate this point. Consider the natural extension of case 4 in the construction above to 3 variables

If $Z = \emptyset$ and one of r_x, $x \in V$ is 0. Here M is the set of variables with the maximum fractional part as before.

$$I'_z = \begin{cases} I_z & \text{if } r_z = 0, \\ [c+1] & \text{if } z \in M, r_z = 1, \text{ and } I_z = (c, c+1) \text{ with } 0 \leq c < c_{\max} \\ I_z & \text{if } z \notin M, r_z = 1, \text{ and } I_z = (c, c+1) \text{ with } 0 \leq c < c_{\max} \end{cases}$$

The problem here arises in constructing the total preorder Rel'. Consider the case when $z \prec x \prec y$ and $r_x = 0$. Consider two distinct valuations ν and ν' such that $\nu[y] = \nu'[y]$ and they belong to the same region $\Re$ in consideration. Now depending on the value of $t = 1 - \mathsf{fract}(\nu[y])$, the successor valuations $\nu_1 = \nu + (0,1,1)t$ and $\nu'_1 = \nu' + (0,1,1)t$ of ν, ν' respectively, might belong to different regions as they might result in different partial orders i.e., $\mathsf{fract}(\nu_1[z]) > \mathsf{fract}(\nu_1[x])$ in one case and $\mathsf{fract}(\nu'_1[z]) < \mathsf{fract}(\nu'_1[x])$

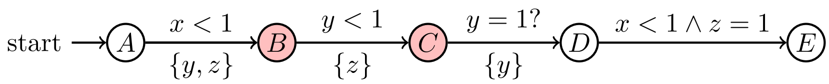

Figure 7 A singular stopwatch automaton with three stopwatches.

in the other. Coming up with such valuations is not difficult and it can be verified by the valuations $\nu[x, y, z] = (0.20, 0.75, 0)$ and $\nu'[x, y, z] = (0.30, 0.75, 0)$. The successor of $\nu'[x, y, z] = (0.30, 0.75, 0)$ after time $1 - 0.75$ is $\nu_1'[x, y, z] = (0.30, 1, 0.25)$, having $z \prec x$ as $0.25 < 0.3$ and the successor of ν after time $1 - 0.75$ is $\nu_1[x, y, z] = (0.20, 1, 0.25)$ having order $x \prec z$ as $0.2 < 0.25$. One might think that further subdividing into smaller regions might help out but as shown in [11] it can be seen that no matter how many such finite number of subdivisions are made this problem will still persist.

This idea can be materialized using a stopwatch automaton with three variables as shown in the Figure 7 and details are given in Example 19.

▶ **Example 19.** We show that any amount of partitioning (finite number) will fail to correctly bi-simulate the given timed automaton.

Note that the shaded nodes B, C have $r_x = 0$ and $r_x = 1$ elsewhere.

Suppose we have partitioned $[0, 1]$ using the n points $p_1, p_2 ..., p_n$ Lets consider the following run of the automaton. We first spend time $t < \frac{p_1}{2}$ in state A and then subsequently time t_1 in state B therefore we will enter state C with clock value $t, t_1, 0$. Due to the continuity of the real line $\exists \delta \in \mathbb{R}_{\geq 0}$, $\exists i$ st. $1 - t - \delta \in (p_i, p_{i+1}) \wedge 1 - t + \delta \in (p_i, p_{i+1})$ Now consider the two clock valuations $\nu = (t, 1 - t - \delta, 0)$ and $\nu' = (t, 1 - t + \delta, 0)$ using the above assertion clearly ν and ν' belong to the same region $\big((0, p_1), (p_i, p_{i+1}), [0]\big)$. Hence they have the same successor and should behave in exactly the same way if the bi-simulation is correct. Now say we enter state C with these clock values we then spend $t \pm \delta$ time in C and enter D with valuation $(t, 0, t \pm \delta)$ now only the valuation $(t, 0, t + \delta)$ will allow us to take the edge to enter E hence clearly these valuations differ in there behavior but the current partitioning is not fine enough to distinguish between them.