View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Loughborough University Institutional Repository

On the Decidability and Complexity of Problems for Restricted Hierarchical Hybrid Systems

Paul C. Bell[1], Shang Chen[1], Lisa Jackson[2]

Department of Computer Science [1], Department of Aeronautical and Automotive Engineering [2], Loughborough University.
P.Bell@lboro.ac.uk, S.Chen3@lboro.ac.uk, L.M.Jackson@lboro.ac.uk

Abstract

We study variants of a recently introduced hybrid system model, called a Hierarchical Piecewise Constant Derivative (HPCD). These variants (loosely called Restricted HPCDs) form a class of natural models with similarities to many other well known hybrid system models in the literature such as Stopwatch Automata, Rectangular Automata and PCDs. We study the complexity of reachability and mortality problems for variants of RHPCDs and show a variety of results, depending upon the allowed powers. These models form a useful tool for the study of the complexity of such problems for hybrid systems, due to their connections with existing models.

We show that the reachability problem and the mortality problem are co-NP-hard for bounded 3-dimensional RHPCDs (3-RHPCDs). Reachability is shown to be in PSPACE, even for n-dimensional RHPCDs. We show that for an unbounded 3-RHPCD, the reachability and mortality problems become undecidable. For a nondeterministic variant of 2-RHPCDs, the reachability problem is shown to be PSPACE-complete.

Keywords: Hybrid Systems, Reachability and Mortality, Piecewise Affine Maps, (Hierarchical) Piecewise Constant Derivatives.

1. Introduction

Hybrid automata are an important class of mathematical model allowing one to capture both discrete and continuous dynamics in the same framework. There is currently much interest in *hybrid systems*, since they can be used to model many practical real world systems in which we have a discrete controller acting in a continuous environment. Their analysis has a huge range of potential applications, such as aircraft traffic management systems, aircraft autopilots, automotive engine control [1], chemical plants [2] and automated traffic systems for example.

Hybrid systems are described by a state-space model given by the Cartesian product of a discrete and continuous set. The system evolves over time according to a set of defined rules until some condition is satisfied, at which point a discrete, non-continuous event occurs. Such an event can cause an update to certain variables and change the continuous dynamics of the continuous variables.

A fundamental question concerning hybrid systems is that of *reachability*: does there exist a trajectory starting from some initial state (or set of states) which evolves to reach a given final state (or set of states) in finite time (defined formally in Section 2)? Related questions, such as *convergence* (does there exist a state, or periodic set of states, towards which the system converges for any initial state) or *control problems* (given an input, can the system be controlled to avoid some 'bad' set of states?), are also important, see [3], for example. Unfortunately, many reachability problems are *undecidable*, even for very restricted hybrid systems [4, 5, 6, 7]. The objective of studying the decidability boundary is twofold; to obtain the most expressive system for which reachability is decidable and to study the simplest system for which it is undecidable.

An important and intuitive model of hybrid system is that of a *Piecewise Constant Derivative* (PCD) system. In this model, we partition the continuous state space into a finite number of nonempty regions, each of which is assigned a constant derivative defining the dynamics of a point within that region (see Section 2 for full details). It was proven in [8] that reachability for PCD systems in two dimensions (2-PCD) is decidable, but for three dimensions (3-PCD), the problem becomes undecidable [4]. One of the important properties of a PCD, which leads to its reachability problem being decidable in dimension two, is that trajectories can never 'cross' each other since each region has a constant derivative assigned. It can be proven that the trajectories are either periodic, or else form an expanding or contracting spiral which can be proven using geometric arguments on the *edge-to-edge successor* function of a 2-PCD.

In [9], a related model, called a *Hierarchical Piecewise Constant Derivative* (HPCD) system was introduced. An HPCD is a 2-dimensional hybrid automaton where the dynamics in each discrete location is given by a 2-PCD (formal details are given in Section 2). Certain edges in the HPCD are called (transition) guards and cause the HPCD to change location if ever the trajectory reaches such an edge. When transitioning between locations, an affine reset rule may be applied. If all regions of the underlying PCDs are bounded, then the HPCD is called bounded. This model can thus be seen as an extension of a 2-PCD.

A 1-dimensional *Piecewise Affine Map* (1-PAM) is a piecewise function which is applied to the 1-dimensional real line, such that the function within each interval of the real line is affine (see Section 2 for details). The reachability problem for 1-PAMs is stated as an open problem in [10, 11, 12, 13, 14], but it becomes undecidable in the 2-dimensional case with fewer than 800 intervals [10]. In [14], 1-PAMs are proven to be equivalent to a 2-dimensional system called a planar pseudo-billiard system, also known as a "strange billiards" model in bifurcation and chaos theory [15] (see '*simulations*' under Section 2 for the definitions of equivalence and simulation). Some decidable results are known under restricted cases. In [12], it is proven that reachability is decidable for 1-dimensional Onto PAMs, which is a model such that every interval in the PAM can be exactly mapped to another. In [13], it is shown that for 1-PAMs over the integers (where all coefficients, the initial point and the final point are integers), the reachability problem is PSPACE-complete, which implies that reachability for rational 1-PAMs is at least PSPACE-hard. If PAMs are replaced by polynomials, the decidability of the reachability problem is open for any dimension [9]. If PAMs are replaced by piecewise rational maps, the reachability problem is undecidable even for dimension one [14].

Reachability for bounded 1-PAMs was shown to be equivalent to that of reachability for bounded HPCDs with either: i) comparative guards, identity resets and elementary flows in Proposition 3.20 of [12] or else ii) affine resets, non-comparative guards and elementary flows in Lemma 3.4 of [12] (See Section 2 for definitions). The authors of [12] also study reachability problems for PCDs defined on 2-*dimensional manifolds*, which we do not consider here.

Related to the reachability problem is the *mortality problem*. The mortality problem is the problem of determining if the trajectories starting from *all* initial points/configurations eventually halt (defined formally in Section 2). The mortality problem has been studied in many different contexts [16, 13, 11, 17, 18] and has connections with program verification, especially in a discrete setting. Similar to the case of reachability, the mortality for 1-PAMs is also stated as an open problem in [11, 13], and undecidability also starts at dimension two, in both the integer case [13], and for the rational case [11]. Global convergence is also known to be undecidable in dimension two [11], although both problems are decidable in dimension one when the piecewise affine function is continuous. The author of [13] also shows Π_2^0-completeness for the integer case.

However, neither reachability nor mortality is a superclass of the other. For the mortality problem, we must prove that *all* initial points will eventually halt, or else the system can be called *immortal* (meaning that the system may diverge, become periodic or quasi-periodic for example). Mortality for 1-PAMs over the integers is known to be PSPACE-complete [13]. Whether the undecidability results in dimension two still hold for a fixed number of intervals is unknown, in both the rational and integer cases.

Similarly to [12], we also aim to study the following question: "What is the simplest class of hybrid systems for which reachability is intractable or undecidable?" To this end, we define the model of *Restricted HPCD* (RHPCD), which is a deterministic bounded HPCD with elementary flows (derivatives of all continuous variables come from $\{0, \pm 1\}$), identity resets and non-comparative guards and is thus a simpler form of HPCD. These restrictions on the resets, derivatives and guards seem natural ones to consider. For example, restricting to identity resets means the trajectory will not have discontinuities in the continuous component, which is similar to a PCD trajectory. Restricting the derivatives to elementary flows ($\{0, \pm 1\}$) has similarities to a *stopwatch automaton*, for which all derivatives are from $\{0, 1\}$. Restricting the guards to be non-comparative gives strong similarities to the guards of a *rectangular automaton* [19], as well as the diagonal-free clock constraints of an *updatable timed automaton* [20].

We prove that a bounded 1-PAM can also be simulated by an RHPCD with arbitrary constant flows or with *linear* resets. Together with the results in [12], the reachability problem for bounded HPCDs is thus shown to be equivalent

	∞ num. of regions	Linear resets	Affine resets	Comparative guards	Arbitrary const. flows	Num. of locations	
Decidable	×	×	×	×	×	$N < \infty$	*
	×	×	×	✓	✓	1	[8]
1-PAM equivalent	×	×	×	×	✓	$\lceil \log_2 n \rceil + 3$	*
	×	×	×	✓	×	$4n$	[12]
	×	×	✓	×	×	1	[12]
	×	✓	×	×	×	$\lceil \log_2 n \rceil + 3$	*
Undecidable	✓	×	×	×	×	1	[12]

Table 1: Summary of decidability status of the reachability problem for 2-RHPCDs when certain conditions are allowed (✓) or disallowed (×). Starred results are contributions of this paper.

to that of bounded 1-PAMs when the HPCD only has one of the following: comparative guards, linear resets (or affine resets) or arbitrary constant flows, see Table 1 for an overview.

We then consider an n-dimensional analogue of RHPCDs, which we denote n-RHPCDs. We show that reachability is decidable (and in PSPACE) for bounded n-RHPCDs and mortality is decidable for bounded 2-RHPCDs. We show a lower bound that reachability and mortality for bounded 3-RHPCDs is co-NP-hard.

We also extend the n-RHPCD model with nondeterminism and unboundedness. If the 2-RHPCD model is endowed with a nondeterministic transition function between locations, then the reachability problem becomes PSPACE-hard. Furthermore, we show that the reachability and mortality problems for unbounded 3-RHPCDs is actually undecidable by an encoding of a Minsky machine. Note that the reachability problem for a 3-HPCD is undecidable, even with only one location, since HPCDs are a superclass of 3-PCDs for which reachability is undecidable [4].

Preliminary versions of this paper appeared in [21, 22].

2. Preliminaries

We write $I \times \{c\}$ to denote $\{(x,c)|x \in I\} \subseteq \mathbb{Q}^2$, where $I \subseteq \mathbb{Q}$ is an (open, half-open or closed) bounded rational interval and $c \in \mathbb{Q}$ is a constant; similarly for $\{c\} \times I$. By abuse of notation, for an interval $I = (s,t)$ where $s,t \in \mathbb{Q}$ and $s \leq t$, a function $f(x) : \mathbb{Q} \to \mathbb{Q}$ and a constant $m \in \mathbb{Q}$, we define $\{f(I)+m\} = (f(s)+m, f(t)+m)$ if $f(s) < f(t)$; otherwise $\{f(I)+m\} = (f(t)+m, f(s)+m)$. Similar definitions exist for half-open and closed intervals. Let $S \in \mathbb{R}^n$ be a set in the n-dimensional Euclidean space. We denote the *closure* of S by $\overline{S}$ and the *interior* of S by $\mathrm{int}(S)$. We use similar definitions as [12] for the following.

Definition 1. (HA) *An n-dimensional Hybrid Automaton (HA) [23] is a tuple $\mathcal{H} = (\mathcal{X}, Q, f, I_0, Inv, \delta)$ consisting of the following components:*

(1) A continuous state space $\mathcal{X} \subseteq \mathbb{R}^n$. Each $\boldsymbol{x} \in \mathcal{X}$ can be written $\boldsymbol{x} = (x_1, \ldots, x_n)$, and we use variables $x_1, \ldots, x_n$ to denote components of the state vector.

(2) A finite set of discrete locations *Q.*

(3) A function $f : Q \to (\mathcal{X} \to \mathbb{R}^n)$, which assigns a continuous vector field on $\mathcal{X}$ to each location. In location $l \in Q$, the evolution of the continuous variables is governed by the differential equation $\dot{\boldsymbol{x}} = f_l(\boldsymbol{x})$. The differential equation is called the dynamics *of location l.*

(4) An initial condition *$I_0 : Q \to 2^{\mathcal{X}}$ assigning initial values to variables in each location.*

(5) An invariant *Inv: $Q \to 2^{\mathcal{X}}$. For each $l \in Q$, the continuous variables must satisfy the condition Inv(l) in order to remain in location l, otherwise it must make a discrete transition or halt.*

(6) A set of discrete transitions *δ. Every $tr \in \delta$ is of the form $tr = (l, g, \gamma, l')$, where $l, l' \in Q$, $g \subset \mathcal{X}$ is called the* guard, *defining when the discrete transition can occur, $\gamma \subset \mathcal{X} \times \mathcal{X}$ is called the* reset relation *applied after the transition from l to l'. By abuse of notation, we also use $\gamma : \mathcal{X} \to \mathcal{X}$ as a function if there will be no confusion.*

An HA is *deterministic* if it has at most one solution for its differential equation in each location and the guards of all the outgoing discrete transitions for each location are mutually exclusive (i.e. the intersection of any two such guards is empty). We consider deterministic HAs, unless otherwise stated. The *size* of an HA is its description size, i.e. the amount of space required to store a description of the HA under a reasonable encoding scheme (for example storing elements of $\mathbb{R}^n$ using a binary encoding). A configuration of an HA is a pair from $Q \times \mathcal{X}$. A *trajectory* of a hybrid automaton $\mathcal{H}$ over a time interval $[0, T]$ and starting from configuration $(l_0, \mathbf{x_0})$ where $l_0 \in Q, \mathbf{x_0} \in \mathcal{X}$ is a pair of functions $\pi_{l_0,\mathbf{x_0}} = (\lambda_{l_0,\mathbf{x_0}}(t), \xi_{l_0,\mathbf{x_0}}(t))$ such that there exists a sequence of times $t_0 = 0 < t_1 < t_2 < \ldots < t_k = T$ and

(1) $\lambda_{l_0,\mathbf{x_0}}(t) : [0, T) \to Q$ is a piecewise function constant on every interval $[t_i, t_{i+1})$.

(2) $\xi_{l_0,\mathbf{x_0}}(t) : [0, T) \to \mathbb{R}^n$ is a piecewise differentiable function and in each piece $\xi_{l_0,\mathbf{x_0}}$ is càdlàg (right continuous with left limits everywhere).

(3) $\xi_{l_0,\mathbf{x_0}}(t) \in \mathrm{Inv}(\lambda_{l_0,\mathbf{x_0}}(t_i))$ for all $t < T$, where $t \in [t_i, t_{i+1})$.

(4) On any interval $[t_i, t_{i+1})$ where $\lambda_{l_0,\mathbf{x_0}}$ is constant and $\xi_{l_0,\mathbf{x_0}}$ is continuous,

$$\xi_{l_0,\mathbf{x_0}}(t) = \xi_{l_0,\mathbf{x_0}}(t_i) + \int_{t_i}^{t} f_{\lambda_{l_0,\mathbf{x_0}}(t_i)}(\xi_{l_0,\mathbf{x_0}}(\tau))d\tau$$

for all $t \in [t_i, t_{i+1})$.

(5) For any t_i, there exists a transition $(l, g, \gamma, l') \in \delta$ such that

(i) $\lambda_{l_0,\mathbf{x_0}}(t_i) = l$ and $\lambda_{l_0,\mathbf{x_0}}(t_{i+1}) = l'$;

(ii) $\xi^{-}_{l_0,\mathbf{x_0}}(t_{i+1}) \in g$ where $\xi^{-}_{l_0,\mathbf{x_0}}(t)$ means the left limit of $\xi_{l_0,\mathbf{x_0}}$ at t;

(iii) $(\xi^{-}_{l_0,\mathbf{x_0}}(t_{i+1}), \xi_{l_0,\mathbf{x_0}}(t_{i+1})) \in \gamma$.

If $(\lambda_{l_0,\mathbf{x_0}}(t), \xi_{l_0,\mathbf{x_0}}(t))$ is defined over $[0,\infty)$, then the trajectory is called infinite. Given a trajectory $\pi_{l_0,\mathbf{x_0}} = (\lambda_{l_0,\mathbf{x_0}}(t), \xi_{l_0,\mathbf{x_0}}(t))$ with sequence of times $t_0 = 0 < t_1 < t_2 < \ldots < t_k = T$, we denote by $\lambda_{l_0,\mathbf{x_0}}(t_0), \lambda_{l_0,\mathbf{x_0}}(t_1), \ldots, \lambda_{l_0,\mathbf{x_0}}(t_k)$ the *symbolic dynamics* of the trajectory, which will be unique for a deterministic HA (and can be infinite). This gives the sequence of locations that the HA visits during the trajectory from time 0 to T.

We can now state two important problems that we will study for various models.

Definition 2. (Reachability and Mortality) *Given an HA $\mathcal{H}$, an initial configuration $c = (l_0, \mathbf{x_0})$ and a final configuration $c' = (l_f, \mathbf{x_f})$, the* reachability problem *is to determine if there exists a time $0 < t < \infty$ such that $\lambda_{l_0,\mathbf{x_0}}(t) = l_f$ and $\xi_{l_0,\mathbf{x_0}}(t) = \mathbf{x_f}$.*

There is more than one possible way to define the mortality problem for HA. We define the mortality problem *in the following way. $\mathcal{H}$ is called* immortal *if there exists at least one configuration $c = (l_0, \mathbf{x_0})$ of $\mathcal{H}$ for which there is an infinite trajectory starting at c, and such that for any $0 < t < \infty$, there exist $t < t_1 < \infty$ such that $\xi_{l_0,\mathbf{x_0}}(t) \neq \xi_{l_0,\mathbf{x_0}}(t_1)$. Otherwise, $\mathcal{H}$ is called* mortal*, in which case we say all the trajectories halt. The* mortality problem *is to determine if a given HA is mortal.*

Definition 3. (n-PCD) *An n-dimensional Piecewise Constant Derivative (n-PCD) system [4] is a pair $\mathcal{H} = (\mathbb{P}, \mathbb{F})$ such that:*

(1) $\mathbb{P} = \{P_s\}_{1\le s\le k}$ is a finite family of nonoverlapping polytopes in $\mathbb{R}^n$ with nonempty interiors, where each $P_s \subseteq \mathbb{R}^n$ is defined as the intersection of finitely many open or closed halfspaces. We also call P_s a region.

(2) $\mathbb{F} = \{\boldsymbol{c}_s\}_{1\le s\le k}$ is a family of vectors in $\mathbb{R}^n$.

(3) The dynamics are given by $\dot{\boldsymbol{x}} = \boldsymbol{c}_s$ for $\boldsymbol{x} \in P_s$.

An n-PCD $\mathcal{H} = (\mathbb{P}, \mathbb{F})$ can equivalently be defined as a restricted type of HA which has n continuous variables, for which there is a location for each $P_s \in \mathbb{P}$, which has corresponding invariant P_s and all derivatives are constant in each location. The guards correspond to the boundary edges between polytopes and no reset is allowed during a transition. We thus see that a PCD is a partitioning of $\mathbb{P}$ into finitely many regions, each of which has an assigned constant derivative or slope. The trajectories are therefore broken lines, with breakpoints at the boundaries of regions. Points along the trajectory follow the derivative of the region they lie inside.

An n-PCD is called bounded if for its regions $\mathbb{P} = \{P_s\}_{1\le s\le k}$, there exists $r \in \mathbb{Q}^+$, such that for all P_s, we have that $P_s \subseteq B_{\mathbf{0}}(r)$, where $B_{\mathbf{0}}(r)$ is an origin-centered open ball of radius r of appropriate dimension. We define the *support set* of a PCD $\mathcal{H}$ as $\mathrm{Supp}_{\mathrm{PCD}}(\mathcal{H}) = \bigcup_{1\le s\le k} P_s$.

In the following we slightly modify the definition of HPCD [12] to allow different dimensions to be studied.

Definition 4. (n-HPCD) *An n-dimensional Hierarchical Piecewise Constant Derivative (n-HPCD) system is a hybrid automaton $\mathcal{H} = (\mathcal{X}, Q, f, I_0, Inv, \delta)$ such that Q and I_0 are defined as in Definition 1, with the dynamics at each $l \in Q$ given by an n-PCD and for each transition $tr = (l, g, \gamma, l')$: (1) its (transition) guard $g \subseteq \mathbb{R}^n$, defined below, is a convex region of dimension $(n-1)$; and (2) the reset relation γ is an affine function of the form: $\boldsymbol{x}' = \gamma(\boldsymbol{x}) = A\boldsymbol{x} + \boldsymbol{b}$, where $A \in \mathbb{R}^{n \times n}$ and $\boldsymbol{b} \in \mathbb{R}^n$. We denote the* internal guards *of an HPCD location to be the boundary edges of the underlying PCD regions which can cause a change of dynamics when they are reached. The* transition guards *are the guards used in transitions between locations. The Invariant (Inv) for a location l is defined to be $Supp_{PCD}(l)$, minus the transition guard for that location, where $Supp_{PCD}(l)$ is the support set of the underlying PCD on l. If all the PCDs are bounded, then the n-HPCD is said to be bounded.*

It can thus be seen that HPCDs are in fact two-dimensional linear Hybrid Automata [12, 24]. The definition of HPCD, as described by [12], is given to emphasise the fact that the trajectory of an HPCD "mostly behaves likely a PCD, with a few reset induced discontinuities". Therefore, the definitions of trajectories, symbolic dynamics and the reachability/mortality problems, can also be defined on HPCD and can be understood in terms of the representation as a two-dimensional linear Hybrid Automaton.
In this paper, we are interested in a *restricted* form of n-HPCD.

1. Under the HPCD model, when transitioning between locations, we may apply an affine reset to non-continuously modify the current point. An n-HPCD has identity (or no) resets if for every transition $tr = (l, g, \gamma, l')$, $\gamma(\mathbf{x}) = \mathbf{x}$ for all points $\mathbf{x} \in \mathbb{R}^n$. This means that starting from any initial configuration $(l_0, \mathbf{x_0})$, for the trajectory $\pi_{l_0,\mathbf{x_0}} = (\lambda_{l_0,\mathbf{x_0}}(t), \xi_{l_0,\mathbf{x_0}}(t))$ we have that $\xi_{l_0,\mathbf{x_0}}(t)$ is a continuous function of t. Note that the trajectory for a PCD is also continuous, and thus this seems to be a natural restriction.

2. An n-HPCD system has *elementary flows* if the derivatives of all variables in every region of each location are from $\{0, \pm 1\}$, otherwise it has *arbitrary constant flows.*

3. Guards are used to change the derivative being applied within a location (internal guards), or to change which location we are in (transition guards) and can be described by Boolean combinations of atomic formulae (linear inequalities). If each atomic formula contains only one variable, then the guard is called non-comparative (meaning the guard is aligned with ones of the axes). An n-HPCD has *non-comparative guards* if all guards (both internal and transition) are non-comparative, e.g., for a 3-RHPCD, $\frac{3}{2} \leq x \leq 7 \wedge y = -1 \wedge 2 \leq z \leq 7$ is a non-comparative guard, but

$0 \leq x \leq 1 \wedge 0 \leq y \leq \frac{1}{2} \wedge z = 5 \wedge x = 2y$ is a comparative guard (due to the term $x = 2y$). Note that non-comparative guards are also known as *rectangular constraints* in the literature.

Definition 5. (n-RHPCD) *An* n-dimensional Restricted Hierarchical Piecewise Constant Derivative System (n-RHPCD) *is a bounded n-HPCD with identity resets, non-comparative guards and elementary flows. See Figure 6a and Figure 6b for an example of a* 3*-RHPCD.*

The following model is the class of 1-dimensional Piecewise Affine Maps (1-PAM). Our approach follows a similar style to [12] where we show various classes of HPCDs where reachability is equivalent to that of a 1-PAM.

Definition 6. (1-PAM) *A* 1*-dimensional Piecewise Affine Map (*1*-PAM) is a function $f : \mathbb{R} \to \mathbb{R}$ (See Figure 3a for an example) such that:*

(1) Domain of f : dom(f)=$\bigcup I_i$, where I_i are disjoint rational intervals.

(2) $\exists a_i, b_i \in \mathbb{Q}$ such that $\forall x \in I_i,\ f(x) = a_i x + b_i$.

(3) f is closed, i.e., range(f) $\subseteq$ dom(f).

A 1*-PAM is called* bounded *if none of its intervals is infinite. In the sequel we will write* 1*-PAM refer to bounded* 1*-PAM unless otherwise stated.*

Open Problem 1. (1-PAM Reachability) *Given a* 1*-dimensional Piecewise Affine Map f, an initial point $x \in \mathbb{Q}$ and a final point $y \in \mathbb{Q}$, does there exist $t \in \mathbb{N}$, such that $f^t(x) = y$?* [1]

In order to prove our undecidability result for an unbounded 3-RHPCD later in the paper, we will require the following well-known computational model.

Definition 7. (Minsky machine) *Informally speaking, a Minsky machine is a two-counter automaton that can increment and decrement counters by one and test them for zero. It is known that a two-counter Minsky machine represents a universal model of computation [25]. Due to their simple structure, Minsky machines are often convenient for proving undecidability results.*

We can represent a counter machine as a simple imperative program $\mathcal{M}$ consisting of a sequence of instructions labelled by natural numbers from 1 *to some $L \in \mathbb{N}$. Any instruction is one of the following forms:*

l: `Add` 1 `to` c_k; `goto` l';

l: `If` $c_k \neq 0$ `then subtract` 1 `from` c_k; `goto` l';
`else goto` l'';

---

[1] $f^t(x)$ denotes $\underbrace{f(f(\ldots f(x)\ldots))}_{t}$

l: `Halt`*;*

where $k \in \{1, 2\}$ *and* $l, l', l'' \in \{1, \dots, L\}$*.*

The machine $\mathcal{M}$ *starts executing with some initial nonnegative integer values in counters* c_1 *and* c_2 *and the control at instruction labelled* 1*. We assume the semantics of all above instructions is clear. Without loss of generality, one can suppose that every machine contains exactly one instruction of the form* l : `Halt` *which is the last one* ($l = L$)*. It should be clear that the execution process (run) is deterministic and has no failure. Any such process is either finished by the execution of* L : `Halt` *instruction or lasts forever.*

As a consequence of the universality of Minsky machines, their halting problem is undecidable:

Theorem 1 ([25])**.** *It is undecidable whether a two-counter Minsky machine halts when both counters initially contain* 0*.*

In Section 3 we extend the results of [12] regarding simulations of 1-PAMs by 2-HPCDs. We follow the similar approach for the definition of simulation used in [4, 12]. We define a simulation with respect to reachability. This means that if a model $\mathcal{A}$ can be simulated by a model $\mathcal{B}$, then it implies that if the reachability problem for $\mathcal{B}$ is decidable (or undecidable), then it must also be decidable (or undecidable) for $\mathcal{A}$. Since we will show simulations of both 1-PAMs and Minsky machines (defined below), we give the definition in terms of a simulation of an arbitrary *deterministic transition system*, which is a pair $\mathcal{A} = (S, \delta')$, where S is a set of states and δ' is a transition function $\delta' : S \to S$.

Definition 8. (Simulation) *We say that a deterministic transition system* $\mathcal{A}$*, with initial configuration* c_0 *and final configuration* c_f*, can be simulated by a 2-HPCD* $\mathcal{H}$ *with respect to the reachability problem if (1) configuration* c_0 *(resp.* c_f*) of* $\mathcal{A}$ *is encoded by a configuration* $(l_0, \mathbf{x_0})$ *(resp.* $(l_f, \mathbf{x_f})$*) of* $\mathcal{H}$*; (2) every configuration of* $\mathcal{A}$ *is encoded by a unique configuration of* $\mathcal{H}$*; (3) a one-step computation of* $\mathcal{A}$ *given by* $\delta'(q_k) = q_{k'}$ *is represented by a trajectory segment from* $(\lambda_{l_0,\mathbf{x_0}}(t), \xi_{l_0,\mathbf{x_0}}(t))$ *to* $(\lambda_{l_0,\mathbf{x_0}}(t'), \xi_{l_0,\mathbf{x_0}}(t'))$ *for some* $0 \le t < t' < \infty$ *on* $\mathcal{H}$*, where* $(\lambda_{l_0,\mathbf{x_0}}(t), \xi_{l_0,\mathbf{x_0}}(t))$ *is the encoding of* q_k*,* $(\lambda_{l_0,\mathbf{x_0}}(t'), \xi_{l_0,\mathbf{x_0}}(t'))$ *is the configuration encoding* $q_{k'}$ *and* $(\lambda_{l_0,\mathbf{x_0}}(t''), \xi_{l_0,\mathbf{x_0}}(t''))$ *is not the encoding of any configuration of* $\mathcal{A}$ *for* $t < t'' < t'$*.*

Finally, we will also require the following *simultaneous incongruences problem*, which is known to be NP-complete [26, 27].

Problem 1. (Simultaneous incongruences) *Given a set* $\{(a_1, b_1), \dots, (a_n, b_n)\}$ *of ordered pairs of positive integers with* $a_i \le b_i$ *for* $1 \le i \le n$*. Does there exist an integer* k *such that* $k \not\equiv a_i \pmod{b_i}$ *for every* $1 \le i \le n$*?*

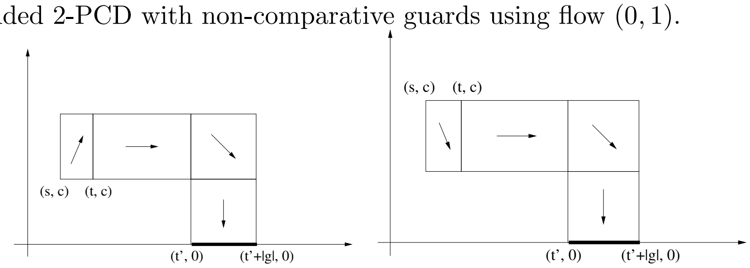
3. Restrictions of 2-HPCDs

In this section, we add some restrictions to the model of 2-HPCDs and explore the decidability of the reachability problem for them. Our starting point is the model of 2-dimensional Restricted HPCD (2-RHPCD, see Section 2 for definitions). We first prove that a 2-RHPCD endowed with arbitrary constant flows can simulate a 1-PAM.

Mappings - A well-known technique for the analysis of PCDs is to study the edge-to-edge successor function [4], also called the Poincaré map [28] of the system. We will use a related concept in this section for HPCDs. Given an HPCD $\mathcal{H}$ and two line segments $L = [\mathbf{p_1}, \mathbf{p_2}]$ and $L' = [\mathbf{p'_1}, \mathbf{p'_2}]$, where $\mathbf{p_1}, \mathbf{p_2}, \mathbf{p'_1}, \mathbf{p'_2} \in \mathbb{R}^\mathbf{2}$. We say that $\mathcal{H}$ *maps* L to L' in location l if for any $0 \le \alpha \le 1$, there exists a $t \ge 0$ such that for the trajectory defined over $[0, t]$, $\xi_{l,(\mathbf{p_1}+(\mathbf{p_2}-\mathbf{p_1})\alpha)}(t) = (\mathbf{p'_1} + (\mathbf{p'_2} - \mathbf{p'_1})\alpha)$ and if the symbolic dynamics of the trajectory is the same for any such choice of $0 \le \alpha \le 1$. Note that $L' = [\mathbf{p'_1}, \mathbf{p'_2}] = [\mathbf{p'_2}, \mathbf{p'_1}]$ and so the definition of mapping holds if we can map L to one of these two representations. A similar definition holds for when L is an open or half-open interval, mutatis mutandis. We call L and L' intervals by abuse of notation (if there is no confusion with rational intervals).

Lemma 1. *Given a* 1*-dimensional interval* $I = (s, t)$*, an affine function* $f(x) = ax + b$ *and a value* m*, where* $a, b, m, s, t \in \mathbb{Q}$ *are constants. Then there exists a* 2*-RHPCD system with arbitrary constant flows which maps* $I \times \{0\}$ *to* $\{f(I) + m\} \times \{0\}$*.*

Proof. We prove this lemma by 3 steps.
Step 1 - Interval $I \times \{0\}$ can be mapped to interval $I \times \{c\}$, where $c \in \mathbb{Q}^+$, by a bounded 2-PCD with non-comparative guards using flow $(0, 1)$.



(a) $a > 0$ (b) $a < 0$

Figure 1: Lemma 1 Step 2: map $(s, t) \times \{c\}$ to $(t', s') \times \{0\}$.

Step 2 - Suppose we have an affine function $f(x) = ax+b$, and the 1-dimensional rational interval $I = (s, t)$. For any constant t' where $t' \ge t > s$, define $g = f(t) - f(s)$ and $s' = t' + |g|$. Assume that $c > |g| + |b| > 0$. Then we show the interval $I \times \{c\}$ can be mapped to $I' \times \{0\} = (t', s') \times \{0\}$ by a bounded 2-PCD system with non-comparative guards, see Figure 1. We need to consider 2 cases, $a > 0$ and $a < 0$. Note the 'orientation' of the interval will be reversed after the mapping.

1. $a > 0$. See Figure 1(a). We use flows $(1, a), (1, 0), (1, -1)$ and $(0, -1)$ to map interval $(s, t) \times \{c\}$ to $(t', t' + |g|) \times \{0\}$.

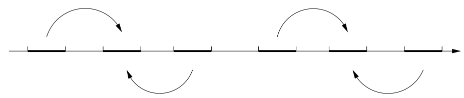


Figure 2: Idea of Theorem 2: map every two adjacent intervals into one interval

2. $a < 0$. See Figure 1(b). We use flows $(1, a), (1, 0), (1, -1)$ and $(0, -1)$ to map $(s, t) \times \{c\}$ to $(t', t' + |g|) \times \{0\}$. As we assume $c > |g| + |b| > 0$, so $c - |g| > |b| > 0$, which means the rectangle $\{(x, y)|t < x < t', c - |g| < y < |g|\}$ does not intersect with the x-axis.

Step 3 - Using a similar idea we can show the interval $I' \times \{0\} = (t', s') \times \{0\}$ can be mapped to $\{f(I) + m\} \times \{0\}$, where $\{f(I)\} = (f(s), f(t))$ if $a > 0$ and $\{f(I)\} = (f(t), f(s))$ if $a < 0$, by a bounded PCD system with non-comparative guards. We can use only the upper or lower half plane of the 2-PCD. Here we only prove the case when $a > 0$ and $f(t) + m < t'$ by using the lower half plane, other cases can be proven similarly.

(i) Use flow $(-1, -1)$ to map $(t', s') \times \{0\}$ to $\{\frac{1}{2}(t' + f(t) + m)\} \times (-\frac{1}{2}|t' - f(t) - m| - |g|, -\frac{1}{2}|t' - f(t) - m|)$;

(ii) Use flow $(-1, 1)$ to map $\{\frac{1}{2}(t' + f(t) + m)\} \times (-\frac{1}{2}|t' - f(t) - m| - |g|, -\frac{1}{2}|t' - f(t) - m|)$ to $(f(s) + m, f(t) + m) \times \{0\}$.

Combining Steps 1, 2 and 3 we get the result of the lemma using a 2-location 2-RHPCD with arbitrary constant flows and non-comparative guards. In location 1 we realize Step 1 and jump to location 2, i.e., the guards are $s_i \leq x < t_i \wedge y = c$. In location 2 we realize Step 2 and Step 3 together because Step 2 only uses the upper plane of a 2-PCD and Step 3 only requires the lower plane of a 2-PCD. A similar proof holds for when I is an open or half-open interval, mutatis mutandis. □

Theorem 2. *A 1-PAM with n intervals can be simulated by a 2-RHPCD with $\lceil \log_2 n \rceil + 3$ locations such that one of the variables has arbitrary constant flows.*

Proof. Suppose 1-PAM $\mathcal{A}$ is defined by $f(x) = a_i x + b_i$ if $x \in I_i$, with $1 \leq i \leq n$ and I_i are rational intervals. In the sequel, we assume all the intervals I_i in $\mathcal{A}$ are left closed and right open. Other cases can be proved similarly. Let the left and right endpoints of I_i be s_i and t_i respectively. First, we show that this 1-PAM can be simulated straightforwardly by an $(n + 1)$-location 2-RHPCD with arbitrary constant flows. We need a single location p as the global state and n locations q_i for each interval I_i, $1 \leq i \leq n$.

1. In location p, we define the corresponding points of the 1-PAM $\mathcal{A}$ on interval $[s_1, t_n) \times \{0\}$. We then map each $I_i \times \{0\}$ to the interval $I_i \times \{c\}$, where $c = |\max\{|a_i|\}(t_n - s_1)| + \max\{|b_i|\}$. (See Lemma 1, Step 1). The transition guards of p are: $s_i \leq x < t_i \wedge y = c$, in which we jump to q_i.

2. In location q_i, map $I_i \times \{c\}$ to $\{f(I_i)\} \times \{0\}$ (see Lemma 1, Step 2&3). The transition guard of q_i is: $s_1 \leq x < t_n \wedge y = 0$, with a jump to location p.

The above method requires $n+1$ locations for a 1-PAM with n intervals. We now give an improved method using a 2-RHPCD with only $\lceil \log_2 n \rceil + 3$ locations. The main idea is to map every two adjacent intervals into one interval in a single location, which is illustrated in Figure 2. The next location performs a similar 'folding' of $\lceil n/2 \rceil$ intervals iteratively and thus we require $O(\log n)$ locations.

Suppose the 1-PAM $\mathcal{A}$ contains n intervals. For every $n \neq 2^d$, $d \in \mathbb{N}$, there exists a minimum integer $k \in \mathbb{N}$ such that $\log_2(n+k) = \lceil \log_2 n \rceil$. The 1-PAM $\mathcal{A}$ can be expanded to $\mathcal{A}'$ such that $f(x) = a_i x + b_i$ if $x \in I_i$, where $i \in \{1, \ldots, n\}$. For every $i \in \{n+1, \ldots, n+k\}$, the length of each new added interval is given by $|I_i| = \epsilon$, and the corresponding affine function is $f(x) = x$. This expansion does not change the dynamics of the 1-PAM $\mathcal{A}$, thus we assume $n = 2^d$, $d \in \mathbb{Z}$.

Again, let the left endpoint and the right endpoint of I_i be s_i and t_i respectively. Define c to be $c = |\max\{|a_i|\}(t_n - s_1)| + \max\{|b_i|\}$ and l to be $l = |t_n - s_1|$.

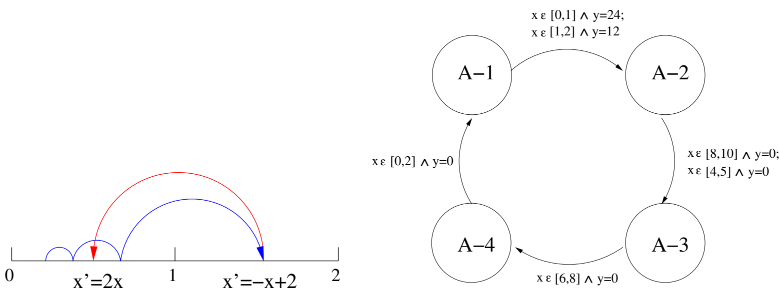
Step 1 Each point $x \in [s_1, t_n]$ of the 1-PAM is encoded by the corresponding point of the interval $[s_1, t_n] \times \{0\}$ in the initial location of the 2-RHPCD. For every $i \in \{1, 2, ..., n\}$, map $I_i \times \{0\}$ to interval $I_i \times \{2(n-i+1)c\}$. (See Lemma 1, Step 1). In this step each interval is mapped to a different height $y = 2(n-i+1)c$. There is a $2c$-length 'gap' between every two intervals I_i and I_{i+1} and I_i is 'higher' than I_{i+1}. In Lemma 1 Step 2 this clearly prevents intersections in the following step.

Step 2 Map each interval $I_i \times \{2(n-i+1)c\}$ to $\{f(I_i) + 2(n-i+1)l\} \times \{0\}$. (See Lemma 1, Step 2). Then between every two intervals there is a 'gap' whose length is at least l.

Step 3 For i from 1 to $\frac{n}{2}$, let $j = 2i - 1$, we can find an interval between $\{f(I_j) + 2(n-j+1)l\} \times \{0\}$ and $\{f(I_{j+1}) + 2(n-j+2)l\} \times \{0\}$ of length l with no function defined on it yet. Using a similar idea as that in the proof of Lemma 1 (Step 3), we can map $\{f(I_j) + 2(n-j+1)l\} \times \{0\}$ using the upper plane and $\{f(I_{j+1}) + 2(n-j+2)l\} \times \{0\}$ using the lower plane to this interval. When we hit the interval, it causes a transition to the next location.

Step 4 Repeat Step 3 for $\log_2(n)$ times until only 1 interval, I_f, remains. Each location maps or 'folds' adjacent intervals of length l into a interval of length l between them, see Figure 2 for an example.

Step 5 If the orientation of I_f is 'reversed' with respect to the initial interval of the 1-PAM $\mathcal{A}$, then map I_f to this initial interval; otherwise, we reverse it before mapping it to the initial interval $[s_1, t_n] \times \{0\}$ in the first location.



(a) A 1-dim. Piecewise Affine Map (b) 2-RHPCD with arbitrary constant flows

Figure 3: The 1-PAM with its equivalent 2-HPCD

Step 1, 2 and 5 each requires 1 location. Step 3 and Step 4 require $\log_2 n$ locations, thus $(\log_2 n) + 3$ locations are required.

In this method every point w in the 1-PAM $\mathcal{A}$ is encoded by a point $(w, 0)$ in the interval $[s_1, t_n) \times \{0\}$ in the location defined in Step 1, including the initial and final points, and a one-step computation of $\mathcal{A}$ from point w to $f(w) = w'$ is represented by a trajectory segment of the 2-RHPCD from point $(w, 0)$ to $(w', 0)$ in the location defined in Step 1. This trajectory segment is calculated from the locations defined in Step 1 to Step 5 above. Thus it is a simulation and the statement of the theorem holds. □

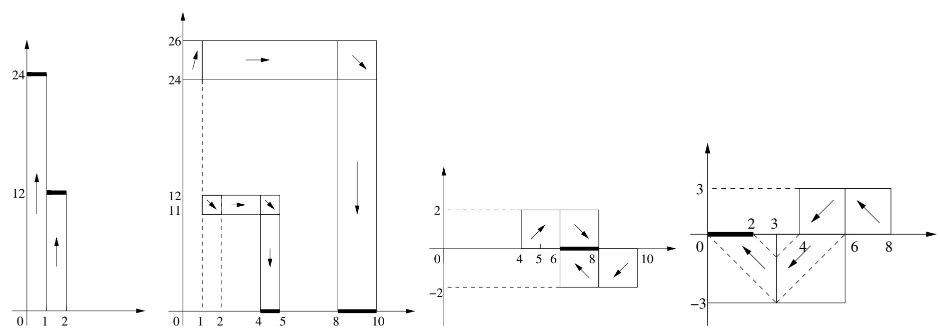
The difficulty of simulating a 1-PAM by a 2-PCD is that regions cannot overlap in a 2-PCD, i.e., one region has only one deterministic constant flow. Thus it is impossible to map several different intervals into a single interval under a 2-PCD, leading us to believe that $\Omega(\log_2 n)$ is a lower bound of the number of locations required to simulate an n-interval 1-PAM by a 2-RHPCD with arbitrary constant flows.

Example 1. *We give an example of a 1-PAM below and show how to simulate it by a 2-RHPCD with arbitrary constant flows in Figures 3, 4.*

$$f(x) = \begin{cases} 2x, & if \quad x \in [0, 1) \\ -x + 2, & if \quad x \in [1, 2] \end{cases}$$

Let the initial point be x_0. The initial location of the 2-HPCD is A-1, with variables $(x, y) = (x_0, 0)$. 2-PCD A-1 corresponds to Theorem 2, Step 1. 2-PCD A-2 separates each interval onto the x axis (Theorem 2, Step 2). 2-PCD A-3 combines together these two intervals (Theorem 2, Step 3). Finally, in A-4, as the final interval $[6, 8]$ has the same orientation as the initial interval $[0, 2]$, we reverse it before mapping it back to the initial interval (Theorem 2, Step 5).

We now show that a 2-RHPCD with linear resets can simulate a 1-PAM.



(a) A-1 (b) A-2 (c) A-3 (d) A-4

Figure 4: The 2-PCDs of the 2-HPCD in Figure 3b (transition guards in bold).

Lemma 2. *The interval $I \times \{0\}$ can be mapped to $\{f(I) + m\} \times \{0\}$ by a 2-RHPCD system with linear resets, where $f(x) = ax + b$ is an affine function, $I = (s, t)$ is a 1-dimensional interval and $a, b, m, s, t \in \mathbb{Q}$ are constants.*

Proof. The proof is similar to the proof of Lemma 1.

Step 1 First map the interval $I \times \{0\}$ to the interval $I \times \{c\}$ by flow $(0, 1)$. Define the transition guard to be $I \times \{c\}$, which jumps to location 2 with linear reset: $x' = |a|x, y' = y$.

Step 2 Using the similar idea in Lemma 1 Step 2, we can map the interval $|a|I \times \{c\}$ to the interval $(t', t' + |g|) \times \{0\}$ by the flows $(1, 1)$ if $(a > 0)$ or $(1, -1)$ if $a < 0$, $(1, 0)$, $(1, -1)$ and $(0, -1)$, where t' and g are defined the same as in Lemma 1.

Step 3 Exactly the same as Lemma 1 Step 3. □

Theorem 3. *A 1-PAM with n intervals can be simulated by a 2-RHPCD containing $\lceil \log_2 n \rceil + 3$ locations with linear resets.*

Proof. Apply Lemma 2 instead of Lemma 1 in the proof of Theorem 2. □

Definition 9. (1-POM) *Let f be a 1-PAM. We call f a* 1-dimensional piecewise offset map *(1-POM) if $f(x) = x + b_i$ for all $x \in I_i$.*

Corollary 1. *A 1-POM can be simulated by a 2-RHPCD, and a 2-RHPCD can be simulated by a 1-POM.*

Proof. The first part follows immediately from Theorem 2. As any coefficient of the linear part of a 1-POM is 1, only elementary flows are required for simulating a 1-POM by a 2-RHPCD. The second part is from [12]. □

4. Higher dimensional RHPCDs

In this section, we start by showing that reachability and mortality problems are co-NP-hard for 3-RHPCDs by an encoding of the simultaneous incongruences problem (see Problem 1). Although this bound may seem quite limited, recall that the system is deterministic, which substantially limits its power. We later show that reachability is in PSPACE for bounded n-RHPCDs, for any $n \geq 1$ and mortality is in PSPACE for $n = 2$. We start with a technical lemma.

Lemma 3. *There exist solutions for the simultaneous incongruences problem with a collection* $\{(a_1, b_1), \ldots, (a_n, b_n)\}$ *if and only if there exists a solution* k *such that* $0 < k \leq \rho$*, where* $\rho = lcm(b_1, \ldots, b_n)$ *and* $lcm(b_1, \ldots, b_n)$ *is the least common multiple of* $b_1, \ldots, b_n$*.*

Proof. The sufficient part is trivial. We show the necessary part. Given an instance $\{(a_1, b_1), \ldots, (a_n, b_n)\}$, let $\rho = \text{lcm}(b_1, \ldots, b_n)$. Then for every $1 \leq i \leq n$, $\rho \equiv 0 \pmod{b_i}$.

For every integer $k > \rho$, we can rewrite k as $k = k_0 + m\rho$, where $0 < k_0 \leq \rho$ and $m \in \mathbb{N}$. Suppose there exists a solution $k_s > \rho$. According to the simultaneous incongruences problem, we know that $k_s \not\equiv a_i \pmod{b_i}$ for all i, where $1 \leq i \leq n$. So we can find a k_0, where $0 < k_0 \leq \rho$, and a positive integer m such that

$$k_s \equiv k_0 + m\rho \not\equiv a_i \pmod{b_i},$$

for every i, where $1 \leq i \leq n$. But $\rho \equiv 0 \pmod{b_i}$ for all $1 \leq i \leq n$, thus

$$k_0 \not\equiv a_i \pmod{b_i}$$

for all $1 \leq i \leq n$, thus k_0 is the solution we want. □

Theorem 4. *The reachability problem for bounded* 3*-RHPCDs is co-NP-hard.*

Proof. Consider an instance of the simultaneous incongruences problem with n pairs $S = \{(a_1, b_1), \ldots, (a_n, b_n)\} \subseteq \mathbb{N}_{>0} \times \mathbb{N}_{>0}$. We will encode the instance into a reachability problem for a 3-RHPCD denoted $\mathcal{H}$. Starting from $k = 1$, we test whether $k \bmod b_i \not\equiv a_i$ holds for each pair (a_i, b_i). If it does hold for every i, then the current value of k is the solution and the reachability problem for the given 3-RHPCD will not have a solution. If for some i we find that $k \bmod b_i \equiv a_i$, then the current value of k is not a potential solution to S. We then increase the value of k by 1 and start the testing all over again. By Lemma 3 there are at most ρ integers to test. If we 'test' all values of $1 \leq k \leq \rho$ and k is not a solution to instance S, then the 3-RHPCD will reach the accepting configuration of $\mathcal{H}$.

We construct $\mathcal{H}$ in the following way. We define 5 locations P, Q, I_1, I_2 and I_3. Locations P and Q are used to 'perform' the modulo operation test for a certain value of k for every pair (a_i, b_i), where $1 \leq i \leq n$. Locations I_1, I_2 and I_3 will increase the value of k by 1 when we find the current k is not a potential solution. See Figure 5.

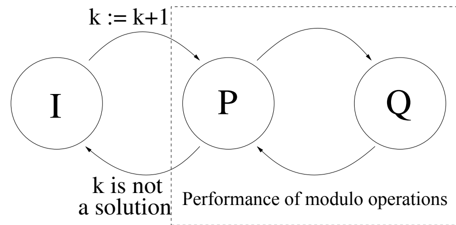
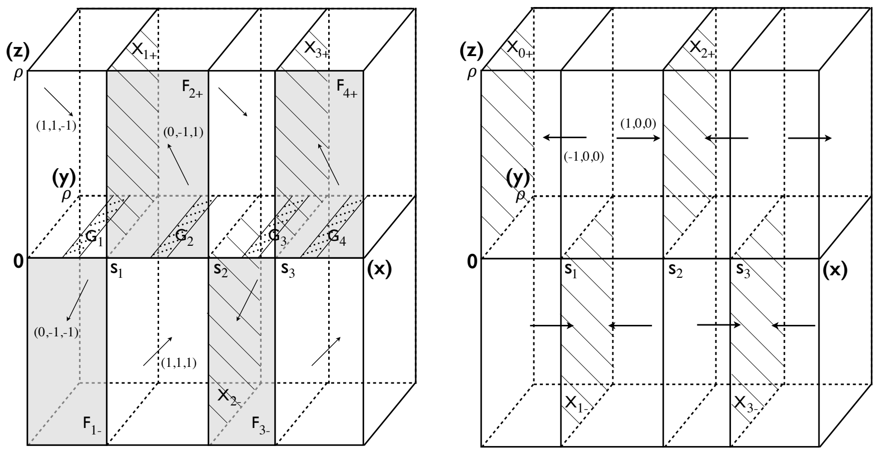


Figure 5: Reachability for 3-RHPCD (location I actually represents 3 locations I_1, I_2 and I_3)



(a) Location P

(b) Location Q

Figure 6: 3-RHPCD encoding simultaneous incongruences problem (only location P and location Q are shown)

We now go to the specifics. Without loss of generality, we assume n is even. Define regions A_i and B_i in locations P and Q :

$$\begin{aligned} A_i &= (s_{i-1}, s_i) \times (0, \rho) \times (0, \rho); \\ B_i &= (s_{i-1}, s_i) \times (0, \rho) \times (-\rho, 0), \end{aligned}$$

where $i \in \{1, 2, \ldots, n\}$, $s_0 = 0$, $s_i = \sum_{j=1}^{i} b_j$ for $1 \leq i \leq n$, and $\rho = \operatorname{lcm}(b_1, \ldots, b_n)$. We call a region A_i, B_i *odd* or *even* depending if i is odd or even. Let $A = \cup_{i=1}^{n} A_i$ and $B = \cup_{i=1}^{n} B_i$. We then define surface $O = [0, s_n] \times [0, \rho] \times \{0\}$, which lies between the regions A and B.

We define four types of surfaces F_{i+}, F_{i-}, X_{i+} and X_{i-}, which will act as transition guards between locations P and Q:

$$\begin{aligned} F_{i+} &= (s_{i-1}, s_i) \times \{0\} \times (0, \rho), && 1 \leq i \leq n; \\ F_{i-} &= (s_{i-1}, s_i) \times \{0\} \times (-\rho, 0), && 1 \leq i \leq n; \\ X_{i+} &= \{s_i\} \times (0, \rho) \times (0, \rho), && 0 \leq i \leq n; \\ X_{i-} &= \{s_i\} \times (0, \rho) \times (-\rho, 0), && 0 \leq i \leq n \end{aligned}$$

Finally we define a set of ε-width strips G_i for $1 \leq i \leq n$:

$$G_i = (s_{i-1} + a_i - \tfrac{\varepsilon}{2}, s_{i-1} + a_i + \tfrac{\varepsilon}{2}) \times [0, \rho] \times \{0\}$$

To carry out the modulo operation for a certain pair (a_i, b_i), we use different regions and derivatives in locations P and Q depending on whether i is odd or even. The support set of both P and Q is given by $\overline{A} \cup \overline{B}$.

If i is odd, then define the derivative to be $(1, 1, -1)$ in region $\overline{A}_i$ and $(0, -1, -1)$ in region $B_i \cup F_{i-}$ of location P. Define the derivative $(-1, 0, 0)$ in $\overline{A}_i$ and $(1, 0, 0)$ in $B_i \cup F_{i-}$ in location Q. We define transition guards from P to Q to be $\overline{X}_{i+} \cup F_{i-}$, and transition guards from Q to P as $\overline{X}_{i-} \cup \overline{X}_{(i-1)+}$. Finally we define a transition guard from P to I_1 for G_i. See Figure 6.

If i is even, then the roles of P and Q will essentially be reversed. The dynamics of a trajectory will be the same as for when i was odd, but reflected in the y plane. Specifically then, we define the derivative to be $(0, -1, 1)$ in $\overline{A}_i \cup F_{i+}$ and $(1, 1, 1)$ in $\overline{B}_i$ of location P. Define the derivative $(1, 0, 0)$ in $A_i \cup F_{i+}$ and derivative $(-1, 0, 0)$ for region $\overline{B}_i$ of location Q. We define transition guards from P to Q by $\overline{X}_{i-} \cup F_{i+}$, and transition guards from Q to P as $\overline{X}_{i+} \cup \overline{X}_{(i-1)-}$. Finally, define a transition guard from P to I_1 for G_i. These details are summarised in Table 2.

Location	Support Set	Flows/derivatives	Transition Guards
P	$\overline{A} \cup \overline{B}$	$\overline{A}_i$ (i is odd): $(1, 1, -1)$ $B_i \cup F_{i-}$ (i is odd): $(0, -1, -1)$ $A_i \cup F_{i+}$ (i is even): $(0, -1, 1)$ $\overline{B}_i$ (i is even): $(1, 1, 1)$	$\overline{X}_{i+} \cup F_{i-}$ if i is odd, $\overline{X}_{i-} \cup F_{i+}$ if i is even, jump to Q
			G_i: jump to I_1
Q	$\overline{A} \cup \overline{B}$	$\overline{A}_i$ (i is odd): $(-1, 0, 0)$ $B_i \cup F_{i-}$ (i is odd): $(1, 0, 0)$ $A_i \cup F_{i+}$ (i is even): $(1, 0, 0)$ $\overline{B}_i$ (i is even): $(-1, 0, 0)$	$\overline{X}_{i-} \cup \overline{X}_{(i-1)+}$ if i is odd, $\overline{X}_{i+} \cup \overline{X}_{(i-1)-}$ if i is even, jump to P
I_1	$\overline{A}$	$(-1, 0, 0)$	$x = 0$ jump to I_2
I_2	$\overline{A}$	$(0, 0, 1)$	$z = 1$ jump to I_3
I_3	$\overline{A}$	$(0, -1, 1)$	$y = 0$ jump to P

Table 2: Reachability problem for 3-RHPCD

For a point (x, y, z), we use the z coordinate to represent the current value of k and the y coordinate as a memory. Assuming i is odd, we start at point $\mathbf{x}_0 = (s_{i-1}, 0, k)$ in location P. Since $\mathbf{x}_0 \in \overline{A}_i$, we move according to the flow $\dot{\mathbf{x}} = (1, 1, -1)$. While $|z| > 0$, every time when $x = b_i + s_{i-1} = s_i$, we jump to Q, since we have a transition guard at $\overline{X}_{i+} = \{s_i\} \times [0, \rho] \times [0, \rho]$. In this case we thus enter location Q at point $(s_i, b_i, k - b_i)$. In Q we keep variables y and z unchanged and simply reset x to 0. To see this, note that $(s_i, b_i, k - b_i) \in \overline{A}_i$ since we assumed $k - b_i > 0$, and therefore we apply the flow $\dot{\mathbf{x}} = (-1, 0, 0)$. We transition back to P when the trajectory reaches the transition guard $\overline{X}_{(i-1)+}$. Each time the trajectory goes from P to Q and jumps back to P, the absolute value of variable z will therefore be subtracted by b_i so we have points $(s_{i-1}, 0, k)$, $(s_{i-1}, b_i, k - b_i)$, $(s_{i-1}, 2b_i, k - 2b_i), \ldots$ each time we transition from P to Q to P in this way.

Eventually, the trajectory will reach the O surface (i.e., $z = 0$), and the value of x will be equal to $s_{i-1} + (k \bmod b_i)$ by the above reasoning. It can thus be seen that when this happens, the trajectory starting at initial point $(s_{i-1}, 0, k)$ of P has changed to $(s_{i-1} + (k \bmod b_i), k, 0)$, since the derivative in P is $\dot{\mathbf{x}} = (1, 1, -1)$ and the z coordinate went from k to 0.

We now have two cases, according to whether $k \bmod b_i$ is equivalent to a_i or not.

1. If $k \bmod b_i \not\equiv a_i$, we reset y to 0 and $|z|$ to k by switching the value of these two variables, and enter region $\overline{B}_{(i+1)}$ to test whether $k \bmod b_{i+1} \not\equiv a_{i+1}$. To do this, assume that the point $(s_{i-1} + (k \bmod b_i), k, 0)$, which lies on the O surface, does not intersect with G_i (the transition guard). In this case, the trajectory enters region B_i and thus the derivative $(0, -1, -1)$ is applied. Eventually, the point will thus reach transition guard $F_{i-} = (s_{i-1}, s_i) \times \{0\} \times (-\rho, 0)$ of location P. This occurs at point $(s_{i-1} + (k \bmod b_i), 0, -k)$ when the y component is 0. Since $(s_{i-1} + (k \bmod b_i), 0, -k) \in F_{i-}$, having transitioned to location Q, we apply derivative $(1, 0, 0)$ and eventually reach point $(s_i, 0, -k) \in X_{i-}$. At this stage we transition to P and we are at point $(s_i, 0, -k)$. This concludes verifying that $k \bmod b_i \not\equiv a_i$. We now move on to consider the next pair (a_{i+1}, b_{i+1}) if there are any additional pairs to check. As explained above, we notice that since we are at point $(s_i, 0, -k)$, the roles of A_i and B_i are interchanged, now that $i + 1$ is an even number, but otherwise the dynamics works in a similar way as just described, just reflected about the y plane.

2. If $k \bmod b_i \equiv a_i$, meaning the current value of k is *not* a potential solution, then $(s_{i-1} + (k \bmod b_i), k, 0) \in G_i = (s_{i-1} + a_i - \frac{\varepsilon}{2}, s_{i-1} + a_i + \frac{\varepsilon}{2}) \times [0, \rho] \times \{0\}$. The transition guard of P thus causes a transition to location I_1 and then I_2 and I_3, (defined in Table 2). This changes the trajectory to point $(0, 0, k + 1)$ and 'restarts' in location P to test whether the new value $k + 1$ is a correct solution[2]. To see this, note that if we start at point $(s_{i-1} + (k \bmod b_i), k, 0) \in \overline{A}$, then we apply derivative $(-1, 0, 0)$ until we hit transition guard on the plane $x = 0$ at which point we are at point $(0, k, 0)$. Similar analysis shows that location I_2 moves point $(0, k, 0)$ to $(0, k, 1)$ before transitioning to I_3. Point $(0, k, 1)$ is then moved to point

[2]Note that here in the guards we do not require exactly $x = a_i + s_{i-1}$, but allow some error ε, so tiny perturbations will not affect our result. The same analysis can be applied to Theorem 5. This implies that the system has robust reachability and mortality problems, but we do not expand on the details here. See more details about robustness in [7].

$(0, 0, k+1)$ which reaches the transition guard at $y = 0$ and transitions back to P.

If there exists a correct solution k to the simultaneous incongruences problem, then starting from point $(0, 0, 1)$ in location P, which is the initial configuration, we will eventually reach point $(0, 0, k)$ in location P and will then traverse through each region $A_i \cup B_i$ for each $1 \leq i \leq n$, before finally reaching a point (x', y', z') in location P lying on the surface $(s_{n-1}, s_n) \times (0, \rho) \times \{0\}$ with $x' \notin (s_{n-1} + a_n - \frac{\varepsilon}{2}, s_{n-1} + a_n + \frac{\varepsilon}{2})$. A trajectory reaching these regions for location P indicates that there exists a solution to the instance of the simultaneous incongruence problem. Therefore, we can simply remove these regions from the support set of location P, so that upon reaching them the trajectory halts without reaching the final configuration (we thus define that these regions have no outgoing transition guard on them). If there does not exist a solution to the simultaneous incongruences problem, then for each value of $1 \leq k \leq n$, there exists some (a_j, b_j) such that $k \bmod b_j \equiv a_j$. As shown in Step (2) above, this means that the trajectory will never reach one of these final regions and we will visit points $(0, 0, 1)$, then $(0, 0, 2)$, then $(0, 0, 3)$ and so on in location P, until eventually we reach point $(0, 0, \rho)$ in location P, which we define as the final configuration. Reaching this configuration means that the instance of the simultaneous incongruence problem has no solution, therefore the reachability problem is co-NP-hard.

The number of regions and guards in the constructed 3-RHPCD is clearly polynomial in the number of pairs of the simultaneous incongruences problem. Furthermore, the points defining each such region can be represented in binary and are therefore polynomial in the description size of the simultaneous incongruences problem. □

Theorem 5. *The mortality problem for bounded* 3*-RHPCDs is co-NP-hard.*

Proof. We encode an instance of the simultaneous incongruences problem into a bounded 3-RPHCD. We construct our 3-RHPCD in such a way that the system is mortal if and only if there is no solution for the corresponding simultaneous incongruences problem, otherwise the system is immortal. This will therefore prove that the mortality problem for 3-RHPCDs is co-NP-hard.

For a pair (a_i, b_i) in the simultaneous incongruences problem, the derivatives of the associated regions $\overline{A}_i$ and $\overline{B}_i$ in locations P and Q are defined the same as in the proof of Theorem 4. In contrast to Theorem 4, in the mortality problem, we are not only concerned about some trajectories starting from certain points $(0, 0, k), 0 < k \leq \rho$, but want to know whether *all* the trajectories halt, starting from any point within the support set of any location.

In the following part we assume i is odd, similar analysis can be applied to the case when i is even. According to the flow $\dot{\mathbf{x}} = (1, 1, -1)$ of an odd region $\overline{A}_i$ in location P, there are 2 boundaries the trajectories will eventually reach: the O surface and the $y = \rho$ surface (some trajectories may also reach the $\overline{X}_{i+}$ or $\overline{X}_{i-}$ surface, but they will jump to location Q and jump back, then reach either one of the above two surfaces at the end). In odd $\overline{A}_i$ in P, all the trajectories which reach the $y = \rho$ surface or reach the strip G_i on the O surface are considered as mortal trajectories and will jump to location M, in which all the trajectories will eventually halt. The trajectories which reach the O surfaces but do not reach the strip G_i are considered as the potential solution trajectories and move on by following the flows for a further check.

Location	Support Set	Flows	Guards
P	$\overline{A} \cup \overline{B}$	$\overline{A}_i$ (i is odd): $(1, 1, -1)$ $A_i \cup F_{i+}$ (i is even): $(0, -1, 1)$ $B_i \cup F_{i-}$ (i is odd): $(0, -1, -1)$ $\overline{B}_i$ (i is even): $(1, 1, 1)$	$\overline{X}_{i+}$ $(i = 1, 3, ..., n-1)$, $\overline{X}_{i-}$ $(i = 2, 4, ...n)$, F_{i+} $(i = 2, 4, ..., n)$, F_{i-} $(i = 1, 3, ..., n-1)$: jump to Q
			$y = \rho, G_i$: jump to M
Q	$\overline{A} \cup \overline{B}$	$\overline{A}_i$ (i is odd): $(-1, 0, 0)$ $A_i \cup F_{i+}$ (i is even): $(1, 0, 0)$ $B_i \cup F_{i-}$ (i is odd): $(1, 0, 0)$ $\overline{B}_i$ (i is even): $(-1, 0, 0)$	$\overline{X}_{i+}$ $(i = 0, 2, ..., n-2)$, $\overline{X}_{i-}$ $(i = 1, 3, ..., n-1)$: jump to P
			$\overline{X}_{n+}$: jump to T
T	$\overline{A} \cup \overline{B}$	$(-1, 0, 0)$	$x = 0$: jump to P
M	$\overline{A} \cup \overline{B}$	$(-1, 0, 0)$	None

Table 3: Mortality problem for 3RHPCD

In contrast to the proof of Theorem 4, in region $\overline{A}_n$ (or $\overline{B}_n$ depending on if i is odd or even) if any trajectory reaches the surface O but does not reach the strip G_n, we do not conclude that we find a solution k and halt with a successful answer. Instead, we keep moving in P until we reach the guard, jump to location T, reset the trajectory to the point $(0, 0, k)$ and go to location P to start the test again. If k indeed is a correct solution to the corresponding simultaneous incongruences problem, the system will loop forever; otherwise the trajectory will go to location M at some region odd $\overline{A}_i$ or even $\overline{B}_i$ in location P. In location M, we have no outgoing transitions and follow derivative $(-1, 0, 0)$. Since the support set is bounded, any trajectory which reaches M will thus eventually halt. Full details are shown in Table 3.

Therefore, if there exists a solution, k, to the simultaneous incongruences problem, then there does exist an infinite trajectory, starting from $(0, 0, k)$ which loops forever and the 3-RHPCD is thus immortal. On the other hand, if there does not exist a solution, then regardless of where we start from in the system, the trajectory will eventually halt and the 3-RHPCD is thus mortal. Therefore the mortality problem is co-NP-hard. □

Proposition 1. *The reachability problem for bounded n-RHPCDs and the mortality problem for bounded 2-RHPCDs is in PSPACE.*

Proof. Given an n-RHPCD $\mathcal{H}$, an initial configuration $(q_0, \mathbf{x_0})$ and a final configuration $(q_f, \mathbf{x_f})$, we first show that starting from $(q_0, \mathbf{x_0})$, the trajectory will hit the internal and transition guards finitely many times before either reaching $(q_f, \mathbf{x_f})$, or detecting a cycle, or hitting some endpoints (at which time the calculation halts), thus 'convergence' to a point is not possible.

By the definition of n-RHPCD (see Definition 4, 5), the guards of $\mathcal{H}$ are of the form

$$\left(\bigwedge_{1\le i \le n \,\wedge\, i \ne j} (a_i \prec x_i \prec' b_i) \right) \wedge (x_j = c_j)$$

where $j \in \{1, \ldots, n\}, x_i, x_j, a_i, b_i, c_j \in \mathbb{Q}$, and $\prec, \prec' \in \{<, \le\}$.

By definition, the components of $\mathbf{x_0} = (x_{0_1}, \ldots, x_{0_n})$ and $\mathbf{x_f} = (x_{f_1}, \ldots, x_{f_n})$ are rational numbers, i.e., $\mathbf{x_0}, \mathbf{x_f} \in \mathbb{Q}^n$. Define γ to be the least common multiple of all the denominators of the constants appearing in the description the n-RHPCD $\mathcal{H}$ (i.e. the guards, invariants, initial and final points) and the continuous components of the initial and final configurations $\mathbf{x_0}, \mathbf{x_f}$. Multiply all such constants by $\gamma \in \mathbb{N}$, i.e., let

$$A_i = \gamma a_i, B_i = \gamma b_i, C_j = \gamma c_j, \mathbf{X_0} = \gamma \mathbf{x_0}, \mathbf{X_f} = \gamma \mathbf{x_f}.$$

Thus, $A_i, B_i, C_j \in \mathbb{Z}$ and $\mathbf{X_0}, \mathbf{X_f} \in \mathbb{Z}^n$. Define a new n-RHPCD $\mathcal{H}'$ with initial configuration $(q_0, \mathbf{X_0})$ and final configuration $(q_f, \mathbf{X_f})$ by replacing a_i, b_i, c_j, $\mathbf{x_0}, \mathbf{x_f}$ by $A_i, B_i, C_j, \mathbf{X_0}, \mathbf{X_f}$. Clearly, $\mathcal{H}$ reaches $\mathbf{x_f}$ iff $\mathcal{H}'$ reaches $\mathbf{X_f}$, and $\mathcal{H}'$ is described by integer values only.

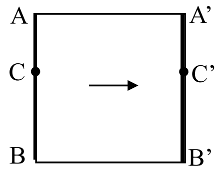
For $\mathcal{H}'$, the trajectory starts at integer configuration $\mathbf{X_0}$, and the guards of $\mathcal{H}'$ are defined by integers. Since all the flows of $\mathcal{H}'$ are chosen from the set $\{0, 1, -1\}$, when one variable x_i of a point of the trajectory, $\mathbf{X_t}$, changes its value from one integer to another, any other variable x_j of $\mathbf{X_t}$ remains an integer. So every time the trajectory hits a guard, i.e., the condition $(\bigwedge_{1\le i\le n \,\wedge\, i\ne j}(A_i \prec x_i \prec' B_i)) \wedge (x_j = C_j)$ is satisfied by the components of $\mathbf{X_t}$, $\mathbf{X_t}$ will have integer components.

We now prove that the problem can be solved in PSPACE. Note that the representation size of γ is clearly polynomial in the representation size of $\mathcal{H}$, thus so is the size of $\mathcal{H}'$. We now show that the representation size of the number of possible transition configurations (the configuration when the trajectory hits the guard and takes transition) of $\mathcal{H}'$ is also polynomial in the size of $\mathcal{H}$.

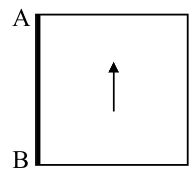
Let $k > 0$ be the number of locations of $\mathcal{H}'$. Since $\mathcal{H}$ is bounded, we can calculate $\tau \in \mathbb{N}$ to be the maximal absolute value of the endpoint of any invariant of $\mathcal{H}$ over all locations. Thus the range of variables of $\mathcal{H}'$ is contained within $[-\gamma\tau, \gamma\tau]$. Since we have n variables, the maximal number of transition configurations of $\mathcal{H}'$, starting at initial configuration $(q_0, \mathbf{X_0})$ is thus $k(2\gamma\tau)^n$, which can be represented in size polynomial in the size of $\mathcal{H}$, since it requires at least $k\log((\gamma\tau)^n) = nk\log(\gamma\tau)$ space to store $\mathcal{H}$ and

$$\frac{\log(k(2\gamma\tau)^n)}{nk\log(\gamma\tau)} = \frac{\log(k) + n\log(2\gamma\tau)}{nk\log(\gamma\tau)} < c$$

for some computable constant $c > 0$. We can use a counter to keep track of the number of transitions the trajectory of $\mathcal{H}'$ makes, starting from $(q_0, \mathbf{X_0})$. As each transition is taken, we can determine if the final configuration was reached



(a) AB is mapped onto $A'B'$



(b) AB “collapses” to A

Figure 7: Edge-to-edge and edge-to-point mappings

since the last transition. Otherwise, we increment the counter and proceed. If the counter reaches $k(2\gamma\tau)^n$, then the configurations must be periodic and we can halt. Hence the reachability problem is in PSPACE.

Using a similar approach, we can show that the mortality problem for 2-RHPCDs is also in PSPACE. According to the constants in the description of $\mathcal{H}$, we can use a similar method as in the reachability proof to find a γ' which allows us to define a new 2-RHPCD $\mathcal{H}''$ such that $\mathcal{H}''$ is mortal iff $\mathcal{H}$ is mortal, where $\mathcal{H}''$ is described by integer coefficients. From the reachability result above we know it is possible to enumerate every integer configuration of $\mathcal{H}''$, as $\mathcal{H}''$ is bounded, and check whether every trajectory halts starting from integer configuration of $\mathcal{H}''$ in PSPACE.

Intuitively, if we connect every adjacent integer point in a 2-PCD (a location) of $\mathcal{H}''$, then each 2-PCD is tessellated by squares of length 1 and the corner points of all squares are integer points since they are the integer configurations of $\mathcal{H}''$. Also each square has exactly one dynamic vector where $\dot{x}_1, \dot{x}_2 \in \{0, 1, -1\}$. We name this technique a *rectilinear tessellation*. An edge of a square will either be mapped onto another edge, or “collapse” to a single point. See Figure 7 for example. In the first case, the local coordinates of the points on an edge are preserved after the mapping. In other words, if point C is on edge AB and C' on edge $A'B'$ is the image of C after the mapping, then $\frac{|AC|}{|CB|} = \frac{|A'C'|}{|C'B'|}$. Thus all points on the same edge have the same symbolic dynamics. Hence for the mortality problem, we only need to consider the corner points of all squares (all the integer points), as well as the middle points of all the edges (in the case the edge is defined by an open set and the end points does not belong to the edge), and all other points will have the same symbolic dynamics as them. To check the middle points of edges simply double the size of γ' and all the points become integers. As long as all the trajectories halt starting from the integer points and the middle points of all the edges, we can conclude that the whole system is mortal. According to the result above, clearly this can be done in PSPACE.

Note that the PSPACE result of mortality only holds for 2-RHPCD, as the local coordinates of points are not preserved in higher dimensions. □

5. Extensions of RHPCDs

Theorem 6. *The reachability and mortality problems are undecidable for unbounded 3-RHPCDs.*

Proof. Consider a two counter (Minsky) machine $\mathcal{M}$, with set of instructions $\{p_i\}$ and two counters c_1 and c_2. For configuration (p_i, c_1, c_2), we define two locations P_i and T_i in an unbounded 3-RHPCD to encode instruction p_i. There are 3 'types' of instruction, where c_k represents a counter ($k \in \{1, 2\}$):

Type I - p_i: `Add` 1 `to` c_k; `goto` p_j;

Type II - p_i: `If` $c_k \neq 0$ `then subtract` 1 `from` c_k; `goto` p_{j_1};
`else goto` p_{j_2};

Type III - p_i : `Halt`.

Given a vector $\mathbf{x} = (x, y, z)$ in an unbounded 3-RHPCD, we use variable x to represent the counter c_1, y to represent the counter c_2 and z as a timer which ensures x or y increases or decreases by exactly 1 at each step.

To encode a Type I instruction p_i on c_1, (resp. c_2), we start from point $(c_1, c_2, 0)$ in location P_i, define the flow in P_i to be $\dot{\mathbf{x}} = (1, 0, 1)$ (resp. $\dot{\mathbf{x}} = (0, 1, 1)$) and the guard to be $z = 1$, jump to T_i. Then in P_i the value of counter c_1 (resp. c_2) is increased by 1. In T_i we define the flow $\dot{\mathbf{x}} = (0, 0, -1)$ and guard $z = 0$ to reset the timer z to 0 and jump to P_j.

For a Type II instruction when $k = 1$, the flow in P_i is defined as $\dot{\mathbf{x}} = (-1, 0, 1)$ with guards: (1) $x = 0 \wedge z < 1$, jump to P_{j_2}; (2) $z = 1$, jump to T_i. In this case, we immediately test whether $x = 0$ when entering P_i and jump to P_{j_2} if it is true. Otherwise, for one time unit we apply derivative $(-1, 0, 1)$, which decreases counter c_1 by 1 (the x-coordinate) and increase the timer by one (the z-coordinate), at which point guard (2) is true. We then go to T_i, define the flow $\dot{\mathbf{x}} = (0, 0, -1)$ and guard $z = 0$ to reset the timer z to 0 and jump to P_{j_1}. A similar encoding can be defined when $k = 2$ mutatis mutandis.

We may assume without loss of generality that the machine only halts when both counters have value zero and the (single) halting instruction is denoted p_H. The reachability problem starts at point $(x, y, 0)$ in initial location P_0 and the problem is to determine if the 3-RHPCD ever reaches point $(0, 0, 0)$ in location P_H. Note that the defined region for the 3-RHPCD is unbounded in the x and y coordinates in all locations, since these coordinates store the counters c_1 and c_2 respectively. The number of regions is bounded. Full details are shown in Table 4.

As any configuration (p_i, c_1, c_2) of $\mathcal{M}$ including the initial point is encoded by the point $(c_1, c_2, 0)$ in location P_i in the 3-RHPCD, the halting configuration $(p_H, 0, 0)$ of $\mathcal{M}$ is encoded by the point $(0, 0, 0)$ in location P_H in the 3-RHPCD, and a one-step computation from (p_i, c_1, c_2) to (p_j, c_1', c_2') in $\mathcal{M}$ is encoded by the trajectory segment from point $(c_1, c_2, 0)$ in location P_i to the point $(c_1', c_2', 0)$ in location P_j, thus a 3-RHPCD can simulate a two counter machine and the reachability problem for a 3-RHPCD is undecidable.

Minsky machine $\mathcal{M}$	3-RHPCD	
p_i	P_i	T_i
`Add` 1 `to` c_1; `goto` p_j	support set: R flow: $\dot{\mathbf{x}} = (1, 0, 1)$ guard: $z = 1$, go to T_i	support set: R flow: $\dot{\mathbf{x}} = (0, 0, -1)$ guard: $z = 0$, go to P_j
`If` $c_1 \neq 0$ `then` $c_1 := c_1 - 1$; `goto` p_{j_1}; `else goto` p_{j_2}	support set: R flow: $\dot{\mathbf{x}} = (-1, 0, 1)$ guard: $z = 1$ go to T_i $x = 0 \wedge z < 1$, go to P_{j_2}	support set: R flow: $\dot{\mathbf{x}} = (0, 0, -1)$ guard: $z = 0$, go to P_{j_1}

Table 4: An unbounded 3-RHPCD simulating the Minsky machine $\mathcal{M}$ for counter c_1, where $R = [0, \infty) \times [0, \infty) \times [0, 1]$

Next, we deal with proving that the mortality problem is also undecidable for unbounded 3-RHPCDs. It was proven in [11] that determining if a Minsky machine, $\mathcal{M}'$, is mortal (i.e. if it halts on all possible configurations) is undecidable. Our approach will be to encode such a Minsky machine $\mathcal{M}'$ using an unbounded 3-RHPCD in a similar way to above. The problem arises however that for mortality, we must prove that *every* initial configuration will eventually halt. We now define a variant of simulation which is required for this proof.

Previously, we defined simulation in terms of reachability, but now we use a similar notion in terms of mortality. Given a Minsky machine $\mathcal{M}'$, we say that an HPCD $\mathcal{H}$ simulates $\mathcal{M}'$ with respect to mortality if properties (2) and (3) of Definition 8 are true, and for any configuration c of $\mathcal{H}$, the trajectory of c will, in finite time, either reach a configuration c' which is the unique encoding of a configuration $m_{c'}$ of $\mathcal{M}'$, after which $\mathcal{H}$ behaves as a simulation of $m_{c'}$, or else halt before reaching such a configuration. Note under this definition that we do not have an initial configuration of $\mathcal{M}'$ or $\mathcal{H}$. Thus $\mathcal{H}$ is mortal if and only if $\mathcal{M}'$ is.

If there exists an immortal run of the machine $\mathcal{M}'$, then there also exists an infinite trajectory of the 3-RHPCD by the above proof. Assume by contradiction that machine $\mathcal{M}'$ is mortal but there exists an infinite trajectory of the 3-RHPCD. We will deal with points not reaching the halting location first.

Assume first that such a trajectory starts in a location P_i or T_i where i is not the halting instruction. Then, by the construction, after a finite number of transitions, the system will reach some location P_j at point $(x, y, 0)$. Assuming that $x, y > 1$, then clearly $(x, y, 0)$ starting in location P_j has a similar dynamics as $(\lfloor x \rfloor, \lfloor y \rfloor, 0)$ starting in location P_j until either $x < 1$ or $y < 1$. This is because the length of time between transitions will always be 1 until this point by the use of timer z and the derivatives of all variables always come from $\{0, \pm 1\}$.

We now deal with the case where $x < 1$ or $y < 1$ at some point, corresponding to a counter being (almost) zero. We slightly modify the 3-RHPCD so that for Type II instructions, if the second guard is true ($x = 0 \wedge z < 1$), we will first zero the z-coordinate (using a new location similar to T_i), before transitioning to P_{j_2}. This means that after such a transition, z, as well as one or both of x, y will be zero. This means that again, $(x, y, 0)$ behaves the same as $(\lfloor x \rfloor, \lfloor y \rfloor, 0)$ in this location. Therefore, any initial configuration corresponds to some initial configuration of $\mathcal{M}'$ and therefore will eventually have zero in both counters and jump to the halting instruction which we define next.

In a similar way to the proof of Theorem 5, we define a 'mortal location' l_M. We define the invariant of l_M as the cube $[0,1)\times[0,1)\times[0,1)$ and the derivative of this cube to be $(-1,-1,-1)$; thus any trajectory reaching l_M halts. Since a correct encoding of $\mathcal{M}'$ will only reach the halting state if both counters are zero, we see that $(0,0,0)$ in location l_M is the unique encoding of the halting configuration of $\mathcal{M}'$. □

We now show a lower bound for *nondeterministic* bounded 2-RHPCDs. A nondeterministic RHPCD can potentially have more than one possible discrete transition available within a location (transition guards can be overlapping).

Corollary 2. *The reachability problem for nondeterministic bounded* 2*-RHPCDs is PSPACE-hard.*

Proof. It was shown in [29] that the reachability (i.e. halting) problem for a nondeterministic bounded 1-counter machine $\mathcal{M}$ is PSPACE-complete when the value of the counter is bounded by a constant $c\in\mathbb{N}$ and when the machine may add or subtract an arbitrary constant $p\in[0,c]$ to the counter in each transition. Transitions are endowed with guards, which are intervals $[g_1,g_2]$ with $0\le g_1\le g_2\le c$, defining that a transition may be taken when the counter lies within the interval. An instruction k, defining a transition between locations p_i and p_j is written in the form $k=(p_i,p,g_1,g_2,p_j)$. See [29] for full details.

Theorem 6 shows a simulation of an (unbounded) 2-counter machine by an unbounded 3-RHPCD, where the x and y coordinates store the values of the two counters c_1 and c_2 (respectively) and the z coordinate is a timer, bounded in the interval $[0,1]$ and used to add/subtract one from a counter. We use a similar construction in dimension two to simulate $\mathcal{M}$. The x coordinate is used to store the counter and the y coordinate is used as the timer to add or subtract an arbitrary amount from $[0,c]$ to the counter in each location.

To simulate an instruction $k=(p_i,p,g_1,g_2,p_j)$, we first define a location P_k. Let $I=[0,c]\times(0,c]$ and then define the invariant of P_k to be $I\cup([g_1,g_2]\times\{0\})$, thus P_k is only defined when the y coordinate is positive, or equal to 0 with the x coordinate in $[g_1,g_2]$. The derivative of P_k is $(1,1)$ if $p>0$ or else $(-1,1)$ and the transition guard to location T_k is defined at $[0,c]\times\{p\}$ (we thus remove $[0,c]\times\{p\}$ from the invariant of the location since they should not overlap). Therefore, starting from a point $(g,0)$ in location P_k, where $g\in[g_1,g_2]$, the trajectory hits the guard at point $(g\pm p,p)$, depending on whether we added or subtracted p. T_k works as in Theorem 6 to zero the timer (y coordinate), with derivative $(0,-1)$ and invariant I. Thus configuration $(g,0)$ in P_k will reach point $(g\pm p,0)$ in location T_k. The transition guard of T_k is defined at $[0,c]\times\{0\}$ and nondeterministically transitions to any location $P_{k'}$ where k' is an instruction of the form (p_j,p',g_1',g_2',p_j') for some p',g_1',g_2',p_j'.

Clearly the description size of the 2-RHPCD is polynomial is the size of $\mathcal{M}$. The initial configuration of the 2-RHPCD is point $(0,0)$ in location P_1.

Determining if $\mathcal{M}$ ever reaches the halting state p_H with counter 0 is PSPACE-complete, which proves the PSPACE-hardness of reaching point $(0, 0)$ in location P_H of the 2-RHPCD since the above construction simulates the operations of machine $\mathcal{M}$ when started in the initial configuration and the reduction from $\mathcal{M}$ is polynomial time. □

Corollary 3. *The reachability problem for nondeterministic bounded* 2*-RHPCDs is PSPACE-complete.*

Proof. Proposition 1 can clearly be seen to still hold even when the system is nondeterministic, since the description size of the number of configurations is still bounded by a polynomial. Thus, by Proposition 1 and Corollary 2 the corollary holds. □

6. Conclusions

We proved for 2-HPCDs that affine resets (or even linear resets), comparative guards and arbitrary constant flows have the same computational power. Being endowed with any one of these powers will make the 2-HPCD model 1-PAM equivalent. We showed that for bounded 3-dimensional Restricted Hierarchical Piecewise Constant Derivative systems (3-RHPCDs), the reachability and mortality problems are co-NP-hard. Reachability is shown to be in PSPACE, even in the n-dimensional case. For 2-RHPCDs the mortality problem is also in PSPACE, and reachability is PSPACE-complete if the model is also nondeterministic. For unbounded 3-RHPCDs, we showed that both problems are undecidable. There remain several interesting open problems regarding reachability and mortality:

- Is the mortality problem for n-RHPCD in PSPACE (or even decidable) for $n > 2$?
- Is there an n for which mortality for n-RHPCD is PSPACE-hard?
- What is the complexity of mortality in dimension two?

It would also be interesting to study other dynamical systems type problems for various types of HPCDs, such as global convergence to a fixed point, global asymptotic stability for example.

The model of RHPCDs restricts various components of the hybrid automaton in ways which have parallels to other models, such as stopwatch automata, rectangular automata and PCDs. RHPCDs have decidable reachability problems for them but endowing them with small additional powers renders them much more powerful. Therefore they seem a useful tool in studying the frontier of undecidability and tractability, in a similar way to the model of HPCDs which inspired them.

Acknowledgements - We thank the referees for their careful checking of this manuscript and especially for the suggestions of an improved lower bound for reachability of nondeterministic 2-RHPCDs.

References

[1] A. Balluchi, L. Benvenuti, M. di Benedetto, C. Pinello, A. Sangiovanni-Vincentelli, Automotive engine control and hybrid systems: challenges and opportunities, Proceedings of the IEEE 88 (7) (2000) 888–912.

[2] N. Bauer, S. Kowalewski, G. Sand, A case study: Multi product batch plant for the demonstration of control and scheduling problems, in: ADPM, Dortmund, Germany, 2000, pp. 969–974.

[3] V. Blondel, J. N. Tsitsiklis, A survey of computational complexity results in systems and control, Automatica 36 (2000) 1249–1274.

[4] E. Asarin, O. Maler, A. Pnueli, Reachability analysis of dynamical systems having piecewise constant derivatives, Theoretical Computer Science 138 (1995) 35–65.

[5] E. Asarin, G. Schneider, S. Yovine, Algorithmic analysis of polygonal hybrid systems, part I: Reachability, Theoretical Computer Science 379 (2007) 231–265.

[6] V. Blondel, J. Tsitsiklis, When is a pair of matrices mortal?, Information Processing Letters 63 (1997) 283–286.

[7] T. Henzinger, J.-F. Raskin, Robust undecidability of timed and hybrid systems, in: Hybrid Systems: Computation and Control, Vol. 1790, 2000, pp. 145–159.

[8] O. Maler, A. Pnueli, Reachability analysis of planar multi-linear systems, in: CAV93, Vol. LNCS 697, Springer-Verlag, 1993, pp. 194–209.

[9] E. Asarin, G. Schneider, Widening the boundary between decidable and undecidable hybrid systems, in: CONCUR'02, Vol. 2421 of LNCS, Springer, 2002, pp. 193–208.

[10] P. Koiran, M. Cosnard, M. Garzon, Computability with low-dimensional dynamical systems, Theoretical Computer Science 312 (1) (1994) 113–128.

[11] V. D. Blondel, O. Bournez, P. Koiran, C. Papadimitriou, J. N. Tsitsiklis, Deciding stability and mortality of piecewise affine dynamical systems, Theoretical Computer Science 255 (1-2) (2001) 687–696.

[12] E. Asarin, V. Mysore, A. Pnueli, G. Schneider, Low dimensional hybrid systems - decidable, undecidable, don't know, Information and Computation 211 (2012) 138–159.

[13] A. M. Ben-Amram, Mortality of iterated piecewise affine functions over the integers: Decidability and complexity, in: 30th Int. Symp. on Theoretical Aspects of Comp Sci. (STACS 2013), Vol. 20, 2013, pp. 514–525.

[14] O. Kurganskyy, I. Potapov, F. Sancho-Caparrini, Reachability problems in low-dimensional iterative maps, International Journal of Foundations of Computer Science 19 (04) (2008) 935–951.

[15] K. Peters, U. Parlitz, Hybrid systems forming strange billiards, International Journal of Bifurcation and Chaos 13 (09) (2003) 2575–2588.

[16] O. Bournez, M. Branicky, On the mortality problem for matrices of low dimensions, Theory of Computing Systems 35 (4) (2002) 433–448.

[17] P. C. Bell, V. Halava, T. Harju, J. Karhumäki, I. Potapov, Matrix equations and Hilbert's tenth problem, International Journal of Algebra and Computation 18 (2008) 1231–1241.

[18] V. Halava, T. Harju, M. Hirvensalo, Undecidability bounds for integer matrices using Claus instances, International Journal of Foundations of Computer Science (IJFCS) 18,5 (2007) 931–948.

[19] T. Henzinger, P. Kopka, A. Puri, P. Varaiya, What's decidable about hybrid automata?, in: 27th ACM STOC, ACM Press, 1995, pp. 373–382.

[20] P. Bouyer, C. Dufourd, E. Fleury, A. Petit, Updatable timed automata, Theoretical Computer Science 321 (2) (2004) 291–345.

[21] P. C. Bell, S. Chen, Reachability problems for hierarchical piecewise constant derivative systems, in: Reachability Problems, Vol. 8169 of Lecture Notes in Computer Science, 2013, pp. 46–58.

[22] P. C. Bell, S. Chen, L. M. Jackson, Reachability and mortality problems for restricted hierarchical piecewise constant derivatives, in: Reachability Problems'14, Vol. LNCS 8762, 2014, pp. 32–45.

[23] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P. H. Ho, X. Nicollin, A. Olivero, J. Sifakis, S. Yovine, The algorithmic analysis of hybrid systems, Theoretical Computer Science 138 (1) (1995) 3–34.

[24] T. A. Henzinger, The theory of hybrid automata, in: 11th Annual IEEE Symposium on Logic in Computer Science, 1996, pp. 278–292.

[25] M. Minsky, Computation: Finite and Infinite Machines, Prentice-Hall International, Englewood Cliffs, 1967.

[26] M. R. Garey, D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman and Co. New York, NY, USA, 1979.

[27] L. J. Stockmeyer, A. R. Meyer, Word problems requiring exponential time, in: Proceedings of the Fifth Annual ACM Symposium on Theory of Computing, STOC '73, ACM, 1973, pp. 1–9.

[28] M. Hirsch, S. Smale, Differential Equations, Academic Press Inc., 1974.

[29] J. Fearnley, M. Jurdzinski, Reachability in two-clock timed automata is PSPACE-complete, in: Automata, Languages and Programming, Vol. LNCS 7966, 2013, pp. 212–223.