Completeness of Lyapunov Abstraction

Rafael Wisniewski[*]
Section of Automation & Control
Aalborg University, Denmark
raf@es.aau.dk

Christoffer Sloth[†]
Section of Automation & Control
Aalborg University, Denmark
ces@es.aau.dk

In this work, we continue our study on discrete abstractions of dynamical systems. To this end, we use a family of partitioning functions to generate an abstraction. The intersection of sub-level sets of the partitioning functions defines cells, which are regarded as discrete objects. The union of cells makes up the state space of the dynamical systems. Our construction gives rise to a combinatorial object - a timed automaton. We examine sound and complete abstractions. An abstraction is said to be sound when the flow of the time automata covers the flow lines of the dynamical systems. If the dynamics of the dynamical system and the time automaton are equivalent, the abstraction is complete.

The commonly accepted paradigm for partitioning functions is that they ought to be transversal to the studied vector field. We show that there is no complete partitioning with transversal functions, even for particular dynamical systems whose critical sets are isolated critical points. Therefore, we allow the directional derivative along the vector field to be non-positive in this work. This considerably complicates the abstraction technique. For understanding dynamical systems, it is vital to study stable and unstable manifolds and their intersections. These objects appear naturally in this work. Indeed, we show that for an abstraction to be complete, the set of critical points of an abstraction function shall contain either the stable or unstable manifold of the dynamical system.

1 Introduction

Formal verification is used to thoroughly analyze dynamical systems. To enable formal verification of a dynamical system, the system can be abstracted by a model of reduced complexity - often a discrete model. A discrete abstraction of a dynamical system is generated by partitioning its state space into cells and determining transitions between the cells. The generation of the abstraction is quite difficult if the behaviors of the discrete and dynamical models should be equivalent (the abstraction is complete). Therefore, most works make the implicit assumption that the studied dynamical system has only one equilibrium point, and that the equilibrium point is stable. This assumption simplifies the abstraction procedure considerably, and allows the identification of partitioning functions that are transversal to solution trajectories. However, what happens if this assumption is relaxed?

Solution trajectories of a dynamical system from [12] are illustrated in Figure 1. The trajectories live on the torus (the torus is obtained by identifying opposing facets with each other). The system has a stable equilibrium point (b), an unstable equilibrium point (a), and two saddle points (α and β). A complete abstraction cannot be generated for such system without choosing a partitioning function that has a level set that includes either the stable manifold or the unstable manifold, i.e. the vector field must be tangential to the partitioning function on this set. Furthermore, it implies that the partitioning functions must identify the stable and unstable manifolds; however, the identification of the stable and unstable manifolds is known to be a very hard problem. This implies that we cannot hope to derive

[*]R. Wisniewski is supported by the EDGE (Efficient Distribution of Green Energy) project.
[†]C. Sloth is supported by MT-LAB, a VKR Centre of Excellence.

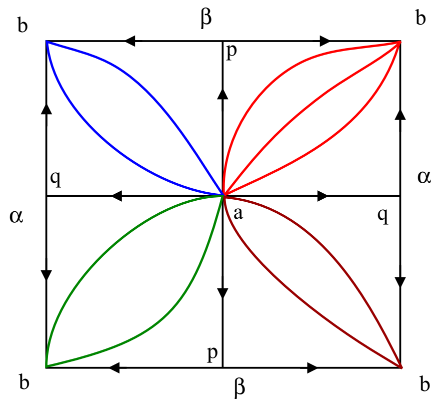

Figure 1: Flow lines of a dynamical system on a torus. The line segments between identical letters are identified to form the torus.

algorithms for finding complete abstractions of general dynamical systems; hence, we conclude that only sound abstractions can be generated in general.

Numerous indirect verification methods for dynamical systems have been developed, i.e., verification methods based on abstracting a considered system by a model of reduced complexity, while preserving certain properties of the original system. To relate the considered system with an abstraction of it, the notions of soundness and completeness are used. Roughly speaking, the reachable set of a sound (complete) abstraction includes (equals) the reachable set of the original system. Remark that the original system and the abstraction may be of different categories. Therefore, the relation between their reachable sets should be appropriately defined, as a solution trajectory of a dynamical system cannot be directly compared to a run of an automaton. This is clarified in Section 3.

Discrete abstractions are generated in [3] for hybrid systems and in [10] for continuous systems. The method developed in [10], is in the class of sign based abstractions. This method relies on partitioning the state space using level sets of multiple functions and subsequently generating the transitions in the abstract model based on the Lie derivative of the partitioning function with respect to the vector field. None of the above methods generate complete abstractions; hence, the dynamics of the abstractions are only known to include the dynamics of the original system. This implies that the result of the verification may be very conservative. Other methods partition the state space by identifying positively invariant sets. In [1], the generation of box shaped invariant sets is treated and is motivated by examples from biology. Tools have also been developed for formal verification. An example is SpaceEx [4] that is based on finding the reachable states using support functions. This tool has good scalability properties, but does not rely on complete abstractions.

Our previous work [9], considers transversal functions in the partitioning of the state space. This commonly accepted transversality condition implies that there is no complete partitioning, even for a dynamical system with critical set being isolated critical points (e.g. a single saddle point). A similar issue was detected in the abstraction of mechanical systems [8]; however, the issue was resolved by only requiring the Lie derivative of the partitioning function along the vector field to be non-positive.

In this paper, we present work in progress on generating complete abstractions of more general dynamical systems. We provide simple illustrative examples that highlight the difficulties faced in the generation of abstractions. Additionally, we show that the existing conditions for generating transversal abstractions cannot trivially be extended. Throughout the paper, we consider dynamical systems without limit cycles.

The paper is organized as follows. Section 2 contains preliminary definitions, Section 3 explains how a dynamical system and an abstract model are related, and Section 4 explains how the state space is partitioned using level sets of functions. Section 5 describes how a timed automaton can be generated from the partitioning and shows that the set of critical points of a complete abstraction function contains either the stable or unstable manifold of the dynamical system, and finally Section 6 comprises conclusions.

1.1 Notation

The set $\{1,\dots,k\}$ is denoted by $\boldsymbol{k}$. B^A is the set of maps $A \to B$. The power set of A is denoted by 2^A. The cardinality of the set A is denoted by $|A|$. We consider the Euclidean space $(\mathbb{R}^n, \langle,\rangle)$, where $\langle,\rangle$ is the standard scalar product. $\mathbb{N} = \{1,2,\dots\}$ is the set of natural numbers, and $\mathbb{Z} = \{\dots,-1,0,1,\dots\}$ is the set of integers.

2 Preliminaries

The purpose of this section is to provide definitions related to dynamical systems and timed automata. Further details on timed automata are found in Appendix A.

2.1 Dynamical Systems

A dynamical system $\Gamma = (X, f)$ has state space $X \subseteq \mathbb{R}^n$ and dynamics described by a system of ordinary differential equations $f : X \to \mathbb{R}^n$

$$\dot{x} = f(x). \tag{1}$$

We assume that f has only non-degenerate singularities [6] and no limit cycles.

The solution of (1) from an initial state $x_0 \in X_0 \subseteq X$ at time $t \geq 0$ is described by the flow function $\phi_\Gamma : [0,\varepsilon] \times X \to X$, $\varepsilon > 0$ satisfying

$$\frac{d\phi_\Gamma(t,x_0)}{dt} = f\left(\phi_\Gamma(t,x_0)\right) \tag{2}$$

for all $t \in [0,\varepsilon]$ and $\phi_\Gamma(0,x_0) = x_0$.

For a map $f : A \to B$, and a subset $C \subseteq A$, we define $f(C) \equiv \{f(x) |\, x \in C\}$. Thus, the reachable set is defined as follows.

Definition 1 (Reachable Set of Dynamical System)**.** *The reachable set of a dynamical system Γ from a set of initial states $X_0 \subseteq X$ on the time interval $[t_1,t_2]$ is*

$$\phi_\Gamma([t_1,t_2], X_0). \tag{3}$$

A positive invariant set is defined in the following.

Definition 2 (Positive Invariant Set)**.** *Given a system $\Gamma = (X, f)$, a set $U \subseteq X$ is said to be positively invariant if for all $t \geq 0$*

$$\phi_\Gamma(t,U) \subseteq U. \tag{4}$$

2.2 Timed Automata

We use the notation of [2] in the definition of a timed automaton. Let $\Psi(C)$ be a set of clock constraints ψ for a set of clocks C. This set contains all invariants and guards of the timed automaton, and is described by the following grammar

$$\psi ::= c \bowtie k | \psi_1 \wedge \psi_2, \tag{5a}$$

where

$$c \in C, k \in \mathbb{R}_{\geq 0}, \text{ and } \bowtie \in \{\boldsymbol{\leq}, \boldsymbol{<}, \boldsymbol{=}, \boldsymbol{>}, \boldsymbol{\geq}\}. \tag{5b}$$

Note that the clock constraint k should be a rational number, but in this paper, no effort is made to convert the clock constraints into rational numbers. However, any real number can be approximated by a rational number with an arbitrary small error $\varepsilon > 0$. To make a clear distinction between syntax and semantics, the elements of $\bowtie$ are bold to indicate that they are syntactic operations.

Definition 3 (Timed Automaton)**.** *A timed automaton $\mathscr{A}$ is a tuple $(E, E_0, C, \Sigma, I, \Delta)$, where*

- *E is a finite set of locations, and $E_0 \subseteq E$ is the set of initial locations.*
- *C is a finite set of clocks.*
- *Σ is the alphabet.*
- *$I : E \to \Psi(C)$ assigns invariants to locations.*
- *$\Delta \subseteq E \times \Psi(C) \times \Sigma \times 2^C \times E$ is a finite set of transition relations. A transition relation is a tuple $(e, G_{e\to e'}, \sigma, R_{e\to e'}, e')$ assigning an edge between two locations, where e is the source location and e' is the destination location. $G_{e\to e'} \in \Psi(C)$ is the set of guards, σ is a symbol in the alphabet Σ, and $R_{e\to e'} \subseteq C$ is a subset of clocks.*

A discrete flow map $\Phi_{\mathscr{A}} : \mathbb{R}_{\geq 0} \times E_0 \to 2^E$ is defined for the timed automata. Details on the flow map and its semantics are found in Appendix A.

3 Abstractions of Dynamical Systems

To evaluate the generated abstraction, it is necessary to define a relation between solution trajectories of a dynamical system and a timed automaton. We define an abstraction function, which associates subsets of the state space to locations of a timed automaton, which makes it possible to define sound and complete abstractions.

Let $\Lambda \subseteq \mathbb{N}$ be an index set. An abstraction of the dynamical system $\Gamma = (X, f)$ consists of a finite number of sets $E \equiv \{e_\lambda | \lambda \in \Lambda\}$ called cells. The cells cover the state space X

$$X = \bigcup_{\lambda \in \Lambda} e_\lambda.$$

To the partition E, we associate an abstraction function, which to each point in the state space associates the cells that this point belongs to.

Definition 4 (Abstraction Function)**.** *Let $\Lambda \subseteq \mathbb{N}$ be an index set and let $E \equiv \{e_\lambda | \lambda \in \Lambda\}$ be a finite partition of the state space $X \subseteq \mathbb{R}^n$. An abstraction function for E is the multivalued function $\alpha_E : X \to 2^E$ defined by*

$$\alpha_E(x) \equiv \{e \in E | x \in e\}. \tag{6}$$

For a given dynamical system Γ, our aim is to simultaneously devise a partition E of the state space X and create a timed automaton $\mathscr{A}$ with locations E such that

1. The abstraction is **sound** on an interval $[t_1,t_2]$:

$$\alpha_E \circ \phi_\Gamma(t,X_0) \subseteq \Phi_{\mathscr{A}}(t,\alpha_E(X_0)), \text{ for all } t \in [t_1,t_2].$$

2. The abstraction is **complete** on an interval $[t_1,t_2]$:

$$\alpha_E \circ \phi_\Gamma(t,X_0) = \Phi_{\mathscr{A}}(t,\alpha_E(X_0)) \text{ for all } t \in [t_1,t_2].$$

4 Partitioning the State Space

This section presents the method for partitioning the state space by functions. The partition of the state space is generated by intersecting sublevel sets of functions, and has two components: slices and cells. A slice is a sublevel set of one partitioning function, whereas a cell is a connected component of the intersection of sublevel sets of more functions.

Let $A \subseteq \mathbb{R}^n$, then $\mathrm{cl}(A)$ denotes the closure of A. We define a slice as the set-difference of positively invariant sets.

Definition 5 (Slice)**.** *A nonempty set S is a slice if there exist two open sets A_1 and A_2 such that*

1. A_1 is a proper subset of A_2,

2. A_1 and A_2 are positively invariant, and

3. $S = \mathrm{cl}(A_2 \backslash A_1)$.

Since A_1 and A_2 are positively invariant sets, a trajectory initialized in S can propagate to A_1, but no solution initialized in A_1 can propagate to S. We adopt the convention that $\emptyset$ is a positively invariant set of any dynamical systems.

To devise a partition of a state space, we need to define collections of slices, called slice-families.

Definition 6 (Slice-Family)**.** *Let $k \in \mathbb{N}$ and*

$$A_0 \subset A_1 \subset \cdots \subset A_k$$

be a collection of positive invariant sets of a dynamical system $\Gamma = (X,f)$ with $X \subseteq A_k$ and $A_0 = \emptyset$. We say that the collection

$$\mathscr{S} \equiv \{S_i = \mathrm{cl}(A_i \backslash A_{i-1}) |\ i \in \boldsymbol{k}\}$$

is a slice-family generated by the sets $\{A_i |\ i \in \boldsymbol{k}\}$ or just a slice-family.

We associate a function to each slice-family $\mathscr{S}$ to provide a simple way of describing the boundary of a slice. Such a function is called a partitioning function.

Definition 7 (Partitioning Function)**.** *Let $\Gamma = (X,f)$ be a dynamical system and let $\mathscr{S}$ be a slice-family generated by the sets $\{A_i |\ i \in \boldsymbol{k}\}$. A continuously differentiable function $\varphi : X \to \mathbb{R}$ is a partitioning function for $\mathscr{S}$ if there is a sequence*

$$a_0 < \ldots < a_k, \ \ a_i \in \mathbb{R} \cup \{-\infty,\infty\}$$

where

$$\mathrm{cl}(A_i) = \varphi^{-1}([a_{i-1},a_i]). \tag{7}$$

Next, a transversal intersection of slices is defined.

Definition 8 (Transversal Intersection of Slices). *We say that the slices S_1 and S_2 intersect each other transversally and write*

$$S_1 \pitchfork S_2 = S_1 \cap S_2 \tag{8}$$

if their boundaries intersect each other transversally.

Cells are generated via intersecting slices.

Definition 9 (Extended Cell). *Let $\mathscr{S} = \{\mathscr{S}^i | i \in \boldsymbol{k}\}$ be a collection of k slice-families and let*

$$\mathscr{G}(\mathscr{S}) \equiv \{1, \dots, |\mathscr{S}^1|\} \times \dots \times \{1, \dots, |\mathscr{S}^k|\} \subset \mathbb{N}^k.$$

Denote the j^{th} slice in $\mathscr{S}^i$ by S^i_j and let $g \in \mathscr{G}(\mathscr{S})$. Then

$$e_{\text{ex},g} \equiv \pitchfork_{i=1}^{k} S^i_{g_i}, \tag{9}$$

where g_i is the i^{th} component of the vector g. Any nonempty set $e_{\text{ex},g}$ is called an extended cell of $\mathscr{S}$.

The cells in (9) are called extended cells, since the transversal intersection of slices may form multiple disjoint sets in the state space; however, it is desired to have cells that are connected.

Definition 10 (Cell). *Let $\mathscr{S} = \{\mathscr{S}^i | i \in \boldsymbol{k}\}$ be a collection of k slice-families. A cell $e_{(g,h)}$ of $\mathscr{S}$ is a connected component of an extended cell of $\mathscr{S}$*

$$\bigcup_h e_{(g,h)} = e_{\text{ex},g}, \text{ where} \tag{10a}$$

$$e_{(g,h)} \cap e_{(g,h')} = \emptyset \qquad \forall h \neq h'. \tag{10b}$$

A finite partition based on the transversal intersection of slices is defined in the following.

Definition 11 (Finite Partition). *Let $\mathscr{S} = \{\mathscr{S}^i | i \in \boldsymbol{k}\}$ be a collection of slice-families. We define a finite partition $E(\mathscr{S})$ by*

$$e \in E(\mathscr{S}) \tag{11}$$

if and only if e is a cell of $\mathscr{S}$.

We propose to use only functions that are nonincreasing along trajectories of the dynamical system Γ as partitioning functions.

Definition 12 (Nonincreasing Partitioning Function). *Let X be an open connected subset of $\mathbb{R}^n$ and suppose that $f : X \to \mathbb{R}^n$ is continuous. Then a real non-degenerate differentiable function $\varphi : X \to \mathbb{R}$ is said to be a partitioning function for f if*

$$\psi(x) \equiv \sum_{j=1}^{n} \frac{\partial \varphi}{\partial x_j}(x) f_j(x) \tag{12a}$$

$$\psi(x) \leq 0 \quad \forall x \in X. \tag{12b}$$

Figure 2 shows a state space partitioned by two partitioning functions, with level sets illustrated by red respectively blue lines. The intersection of slices generates an extended cell (gray area), consisting of four cells (each connected component of the gray area).

In the next section, we show how the partitioning functions should be chosen to generate a complete abstraction.

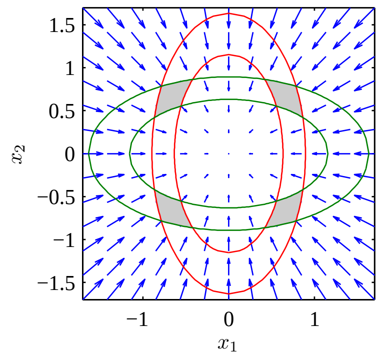

Figure 2: Illustration of a vector field in a state space, partitioned using level sets of two partitioning functions.

5 Generation of Timed Automaton from Finite Partition

A timed automaton $\mathscr{A}$ is generated from a finite partition $E(\mathscr{S})$ as follows.

Definition 13 (Generation of Timed Automaton)**.** *Given a finite collection of slice-families $\mathscr{S} = \{\mathscr{S}^i |\ i \in \boldsymbol{k}\}$, and pairs of times $\mathscr{T} = \{(\underline{t}^i_{g_i}, \bar{t}^i_{g_i}) \in \mathbb{R}^2_{\geq 0} |\ i \in \boldsymbol{k}, g_i \in \{1,\dots,|\mathscr{S}^i|\}\}$. We define the timed automaton $\mathscr{A}(\mathscr{S},\mathscr{T}) = (E, E_0, C, \Sigma, I, \Delta)$ by*

- ***Locations:*** *The locations of $\mathscr{A}$ are given by*

$$E = E(\mathscr{S}). \tag{13}$$

This means that a location $e_{(g,h)}$ is identified with the cell $e_{(g,h)} = \alpha^{-1}_{E(\mathscr{S})}(\{e_{(g,h)}\})$ of the partition $E(\mathscr{S})$, see Definition 4.

- ***Clocks:*** *The set of clocks is $C = \{c^i |\ i \in \boldsymbol{k}\}$.*
- ***Alphabet:*** *The alphabet is $\Sigma = \{\sigma^i |\ i \in \boldsymbol{k}\}$.*
- ***Invariants:*** *In each location $e_{(g,h)}$, we impose an invariant*

$$I(e_{(g,h)}) = \bigwedge_{i=1}^{k} c^i \leq \bar{t}^i_{g_i}. \tag{14}$$

- ***Transition relations:*** *If a pair of locations $e_{(g,h)}$ and $e_{(g',h')}$ satisfy the following two conditions*
  1. *$e_{(g,h)}$ and $e_{(g',h')}$ are adjacent; that is $e_{(g,h)} \cap e_{(g',h')} \neq \emptyset$, and*
  2. *$g'_i \leq g_i$ for all $i \in \boldsymbol{k}$.*

  Then there is a transition relation

$$\delta_{(g,h)\to(g',h')} = (e_{(g,h)}, G_{(g,h)\to(g',h')}, \sigma^i, R_{(g,h)\to(g',h')}, e_{(g',h')}),$$

where

$$G_{(g,h)\to(g',h')} = \bigwedge_{i=1}^{k} \begin{cases} c^i \geq \underline{t}^i_{g_i} & \text{if } g_i - g'_i = 1 \\ c^i \geq 0 & \text{otherwise.} \end{cases} \tag{15a}$$

Note that $g_i - g'_i = 1$ whenever a transition labeled σ^i is taken.

Let $i \in \boldsymbol{k}$. We define $R_{(g,h)\to(g',h')}$ by

$$c^i \in R_{(g,h)\to(g',h')} \tag{15b}$$

iff $g_i - g'_i = 1$.

To ensure that the properties of an abstraction is not only valid for a particular choice of level sets, we impose the derived condition for any choice of level set in the partition.

From Definition 13, it is seen that to generate a timed automaton, it is required to devise a partition of the state space, and find a set of invariant and guard conditions. Therefore, we provide a condition under which an abstraction is complete, recall the definition of a complete abstraction in Section 3.

Proposition 1 ([9])**.** *Given a dynamical system $\Gamma = (X, f)$, a collection of partitioning functions $\{\varphi^i | \, i \in \boldsymbol{k}\}$, a collection of values $\{a^i_{g_i} \in \mathbb{R} | \, i \in \boldsymbol{k}, g_i \in \{1, \dots, |\mathscr{S}^i|\}\}$ generating $\mathscr{S}$, and pairs of times $\mathscr{T} = \{(\underline{t}^i_{g_i}, \bar{t}^i_{g_i}) | \, i \in \boldsymbol{k}, g_i \in \{1, \dots, |\mathscr{S}^i|\}\}$. The timed automaton $\mathscr{A}(\mathscr{S}, \mathscr{T})$ is a complete abstraction of Γ if and only if for any $i \in \boldsymbol{k}$*

1. *for any pair of regular values $(a^i_{g_i-1}, a^i_{g_i})$ with $g \in \mathscr{G}(\mathscr{S})$ (see the definition of $\mathscr{G}(\mathscr{S})$ in Definition 9) there exists a time $t^i_{g_i}$ such that for all $x_0 \in (\varphi^i)^{-1}(a^i_{g_i})$*

$$\phi_\Gamma(t^i_{g_i}, x_0) \in (\varphi^i)^{-1}(a^i_{g_i-1}) \tag{16}$$

and

2. $\bar{t}_{S^i_{g_i}} = \underline{t}_{S^i_{g_i}} = t^i_{g_i}$.

We say that $\{\varphi^i | \, i \in \boldsymbol{k}\}$ generates a complete abstraction if there exist times $\mathscr{T}$ such that Proposition 1 is satisfied.

In the introduction of the paper, we claim that it is impossible to generate a complete abstraction of a system with a saddle point using transversal partitioning functions. In the following example a complete abstraction is given for such a system by nonincreasing partitioning functions.

Example 1. *Consider the following two-dimensional linear vector field*

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -x_1 \\ x_2 \end{bmatrix}. \tag{17}$$

The system has a saddle point and its phase plot is shown in Figure 3.

We choose the following partitioning functions

$$\varphi_1(x_1, x_2) = x_1^2, \tag{18a}$$
$$\varphi_2(x_1, x_2) = -x_2^2, \tag{18b}$$

Recall from (12a) *that we denote Lie derivatives by ψ. The Lie derivatives of the partitioning functions along the vector field in* (17) *become*

$$\psi_1(x_1, x_2) = -2x_1^2, \tag{19a}$$
$$\psi_2(x_1, x_2) = -2x_2^2. \tag{19b}$$

This implies that the abstraction generated by $\{\varphi_1, \varphi_2\}$ is complete. Completeness is concluded from Proposition 1.

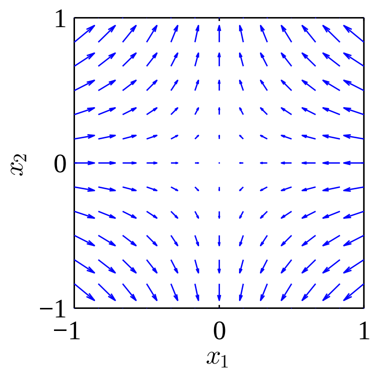

Figure 3: Phase plot of a linear system with a saddle point.

Proposition 1 does not provide a straightforward method for computing a complete abstraction, as the conditions are not numerically tractable. Therefore, we rephrase (16) as a relation between the level sets of the partitioning function and its derivative along the vector field. It is difficult to determine if the partitioning functions in Example 1 generate a complete abstraction from Proposition 1; however, this is clear from the following proposition, as the level sets of ϕ and ψ coincide.

Proposition 2. *Let Γ be a dynamical system and φ be a smooth function, in particular a partitioning function. Then for any regular value $a \in \mathbb{R}$, the following statements are equivalent*

1. *there exists $b \in \mathbb{R}$ such that*

$$\{x \in \mathbb{R}^n \mid \varphi(x) - a = 0\} \subseteq \{x \in \mathbb{R}^n \mid \psi(x) - b = 0\}, \tag{20}$$

where

$$\psi(x) \equiv d\varphi(f)(x) = \sum_i \frac{\partial \varphi}{\partial x_i}(x) f_i(x).$$

2. *there exists $\varepsilon > 0$ such that for any $t \in (-\varepsilon, \varepsilon)$*

$$\varphi(\phi_\Gamma(t, x_1)) = \varphi(\phi_\Gamma(t, x_2)) \quad \forall x_1, x_2 \in \varphi^{-1}(a). \tag{21}$$

Proof. We show that 2) implies 1). We differentiate both sides of

$$\varphi(\phi_\Gamma(t, x_1)) = \varphi(\phi_\Gamma(t, x_2))$$

with respect to t

$$\sum_i \frac{\partial \varphi}{\partial x_i}(\phi_\Gamma(t, x_1)) f_i(\phi_\Gamma(t, x_1)) = \sum_i \frac{\partial \varphi}{\partial x_i}(\phi_\Gamma(t, x_2)) f_i(\phi_\Gamma(t, x_2)), \tag{22a}$$

$$\psi(\phi_\Gamma(t, x_1)) = \psi(\phi_\Gamma(t, x_2)). \tag{22b}$$

At $t = 0$, $\psi(x_1) = \psi(x_2)$. Hence, (20) is satisfied.

To show that 1) implies 2), we define a convenient state transformation inspired by [7, p. 13]. In the new coordinates, the vector field has only one nonzero component.

Let $M = \varphi^{-1}(a)$ be a smooth manifold. By Sard's Theorem, there exists an open neighborhood U of a such that any point in U is a regular value of φ [6, p. 132]. Define the smooth function $\eta : \varphi^{-1}(U) \to \mathbb{R}$ by

$$\eta \equiv \frac{1}{||\nabla\varphi||^2}, \tag{23}$$

where $\nabla\varphi$ is the gradient of φ (with respect to a Riemannian metrics $\langle\cdot,\cdot\rangle$ on M). The function η is well defined, since $\varphi^{-1}(U) \subset M$ is an (open) set of regular points of φ. Define the vector field ξ on $\varphi^{-1}(U)$ by

$$\xi \equiv \eta\nabla\varphi. \tag{24}$$

The derivative of φ in the direction of ξ is

$$d\varphi(\xi) = \langle\nabla\varphi, \xi\rangle = \eta\langle\nabla\varphi, \nabla\varphi\rangle = 1, \tag{25}$$

Choose $a', a'' \in \mathbb{R}$ such that $a' < a < a''$ and $[a', a''] \subset U$. We define the map $F(\cdot,\cdot) : M \times [a', a''] \to \varphi^{-1}([a', a'']) \subset X$, where $F(x_0, t)$ is the solution of ξ from initial state x_0. From (25), we have

$$t \mapsto \varphi \circ F(x_0, t) = a + t. \tag{26}$$

We represent the vector field f and function φ in new coordinates $q = F(x,t)$

$$\tilde{f}(q) = (DF(q))^{-1} f \circ F(q) \tag{27a}$$
$$\tilde{\varphi}(q) = \varphi \circ F(q) \tag{27b}$$

Notice that the vector fields f and $\tilde{f}$ are F-related. The differential of $\tilde{\varphi}$ is

$$d\tilde{\varphi}(q) = d\varphi(F(q))DF(q). \tag{28}$$

Hence, the derivative of $\tilde{\varphi}$ if the direction of $\tilde{f}$ is

$$d\tilde{\varphi}(\tilde{f}) = d\varphi \circ F\ DF(DF)^{-1} f \circ F \tag{29}$$
$$= d\varphi(f) \circ F. \tag{30}$$

Let $x_1, x_2 \in M$, then $F(x_1, t)$ and $F(x_2, t)$ are regular points for $t \in (-\varepsilon, \varepsilon)$. By (20)

$$d\varphi(f)(F(x_1,t)) = d\varphi(f)(F(x_2,t)). \tag{31}$$

We define $\tilde{\psi} = d\tilde{\varphi}(\tilde{f})$, and conclude that

$$\tilde{\psi}(x_1, t) = \tilde{\psi}(x_2, t) \tag{32}$$

From (26), $\tilde{\varphi}$ only depends on the last coordinate. Therefore, denoting $f = (f_1, \ldots, f_n)$, we have

$$\tilde{\psi} = d\tilde{\varphi}\tilde{f} = \tilde{f}_n. \tag{33}$$

Since $\tilde{\psi}(x_1,t) = \tilde{\psi}(x_2,t)$ for any pair $x_1, x_2 \in M$ and $t \in [a', a'']$, we have $\tilde{f}_n(x_1,t) = \tilde{f}_n(x_2,t)$. In other words, the nth component of the vector field $\tilde{f}$ depends only on its last n coordinate t. As a consequence, denoting the flow map of the vector field $\tilde{f}$ by $\phi_{\tilde{\Gamma}}$, we have

$$\phi_{\tilde{\Gamma}}((x_1,0),t) = \phi_{\tilde{\Gamma}}((x_2,0),t) \in M \times (-\varepsilon, \varepsilon). \tag{34}$$

The vector field f and $\tilde{f}$ are F-related, hence also

$$\phi_{\Gamma}(x_1,t) = \phi_{\Gamma}(x_2,t). \tag{35}$$

Thus, the inclusion (16) holds. □

Unfortunately, we cannot relax proposition to include critical values as well, i.e., it does not hold for any $a \in \mathbb{R}$. This is clarified in the following, by presenting a dynamical system with state space $X \subseteq \mathbb{R}^2$ for which there exists no complete abstraction generated by transversal partitioning functions.

According to Proposition 1, it takes the same time for two trajectories to propagate between level sets of complete partitioning functions. Consider a dynamical system with flow lines shown in Figure 4. Level sets of two partitioning functions are illustrated at the saddle point inside the gray circle. From the

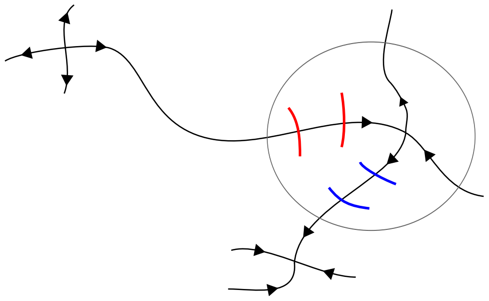

Figure 4: Flow lines of a dynamical system, with a saddle point, a stable equilibrium point, and an unstable equilibrium point.

red level set, one trajectory goes to the saddle point; however, the remaining trajectories diverge from the saddle point. Therefore, it cannot take the same time to propagate between any two level sets, if the partitioning function is decreasing along the vector field.

In the following, we denote the stable (unstable) manifold by $W^{\mathrm{s}}(p)$ ($W^{\mathrm{u}}(p)$) [5].

Lemma 1. *Let φ be a partitioning function generating a complete abstraction of $\Gamma = (X, f)$ and let p be a singular point of f. Then p is a critical point of φ.*

Proof. Suppose that p is a regular point of φ with regular value c. Then $\varphi^{-1}(c)$ is an $n-1$ dimensional manifold. Without loss of generality, we assume that p is not stable. Let $x \in \varphi^{-1}(c) \backslash W^{\mathrm{s}}(p)$. Then there exists $\tau > 0$ such that $\phi(\tau, x) \notin \varphi^{-1}(c)$, but $p = \phi(\tau, p) \in \varphi^{-1}(c)$, since p is a singular point of f. This contradicts completeness. □

The following theorem is the main contribution of the paper, showing that complete abstractions identify stable and unstable manifolds.

Theorem 1. *Let $\mathrm{Reg}(\varphi)$ denote the set of regular values of φ, generating a complete abstraction of $\Gamma = (X, f)$, let X be a connected compact manifold, and let p be a singular point of f. Suppose that $\varphi^{-1}(\mathrm{Reg}(\varphi)) \cap W^{\mathrm{s}}(p) \neq \emptyset$. Then*

$$W^u(p) \subseteq \varphi^{-1}(\varphi(p)).$$

Sketch of Proof. Let $Z \equiv \varphi^{-1}(\mathrm{Reg}(\varphi)) \cap W^{\mathrm{s}}(p)$ and let $R(x) \equiv \varphi^{-1}(\varphi(x))$. Suppose that there exists a point $x \in Z$ such that $Y(x) \equiv R(x) \backslash W^{\mathrm{s}}(p) \neq \emptyset$.

Per definition of $W^{\mathrm{s}}(p)$, $\lim_{t \to \infty} \phi(x, t) = p$. Let $y \in Y(x)$ then there exists a singular point $p' \neq p$ such that $y \in W^{\mathrm{s}}(p')$. Note that any solution goes to a singular point, as the state space X is compact.

Since $\varphi(x) = \varphi(y)$, the abstraction is complete, and the flow map ϕ is continuous, we conclude from Proposition 2 that $\varphi(p) = \varphi(p')$. Thus, for any $z \in W(p, p') \equiv W^{\mathrm{u}}(p) \cap W^{\mathrm{s}}(p')$ the value of φ is constant, i.e., $\varphi(p) = \varphi(z)$. Therefore, $W(p, p') \subseteq R(p)$.

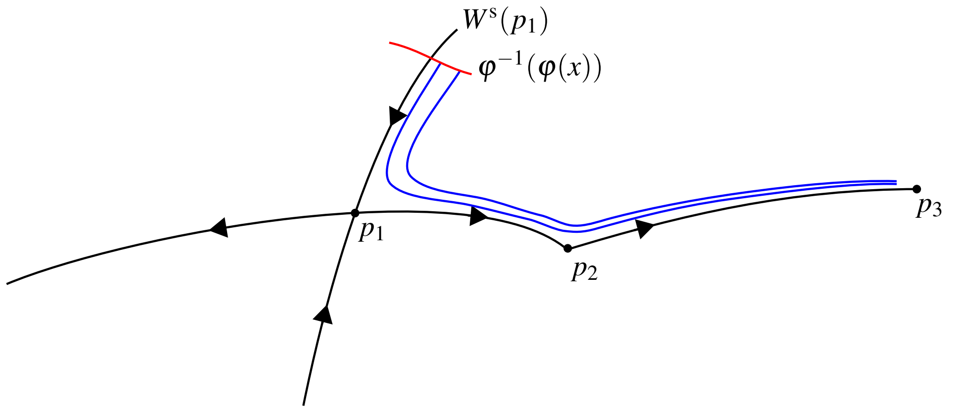

Figure 5: State space of a system with singular points p, q_1, and q_2. No solution initialized on the red level set reaches q_1, but solutions get arbitrarily close to q_1.

To show that $W^{\mathrm{u}}(p) \subseteq R(p)$, we exploit that any $z \in R(x)$, $z \in W^{\mathrm{s}}(p'')$ for some singular point p''. Therefore, $\varphi(p) = \varphi(p'')$ for any such singular point. We use notation $p \preceq p'$ iff there is a flow line from p to p'. Whenever there is a sequence $\{p_1, \dots, p_n\}$ of singular points of the vector field f such that

$$p = p_1 \preceq p_2 \preceq \cdots \preceq p_n,$$

then $\varphi(p_i) = \varphi(p)$ for all $i \in \{1, \dots, n\}$. Such a scenario is illustrated in Figure 5. Thus, we have

$$W^{\mathrm{u}}(p) = \bigcup_{p \preceq \cdots \preceq p'} W(p, p') \subseteq R(p).$$

If $Y(x) = \emptyset$ for all $x \in Z$, then

$$\bigcup_{x \in Z} R(x) \subseteq W^{\mathrm{s}}(p).$$

The dimension of $R(x)$ is $n-1$. Furthermore, since $R(x)$ is a closed compact manifold and the vector field is transversal to $R(x)$, there exists $\tau > 0$ such that

$$\phi(R(x), (-\tau, \tau))$$

that is an embedded manifold of dimension n. Thereby $\dim(W^{\mathrm{s}}(p)) = n$, and $W^{\mathrm{u}}(p) = \{p\} \subseteq R(p)$. □

One could think that the inclusion $W^u(p) \subseteq \varphi^{-1}(\varphi(p))$ in Theorem 1 could be replaced by an equality; however, the following counterexample demonstrates that this cannot be done.

Example 2. *Consider the following two dimensional linear vector field from Example 1*

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -x_1 \\ x_2 \end{bmatrix}. \tag{36}$$

The system has a saddle point $p = (0,0)$ and its phase plot is shown in Figure 3. We modify the partitioning function $\varphi_1(x_1,x_2) = x_1^2$ from Example 1, to obtain a case where

$$W^u(p) \subsetneq \tilde{\varphi}_1^{-1}(\tilde{\varphi}_1(p)).$$

For this purpose, we construct a bump function, according to [11]. Let

$$f(t) = \begin{cases} e^{-1/t} & \text{for } t > 0 \\ 0 & \text{for } t \leq 0 \end{cases}$$

$$g(t) = \frac{f(t)}{f(t) + f(1-t)}.$$

From f and g, we choose the partitioning function

$$\tilde{\varphi}_1(x_1, x_2) = g\left(\frac{x_1^2 - a^2}{b^2 - a^2}\right) \cdot x_1^2, \tag{37}$$

with $a = 0.5$ and $b = 2$. The graphs of $g\left(\frac{x_1^2-a^2}{b^2-a^2}\right)$, $\tilde{\varphi}_1(x_1,x_2)$, and $\varphi_1(x_1,x_2)$ are shown in Figure 6.

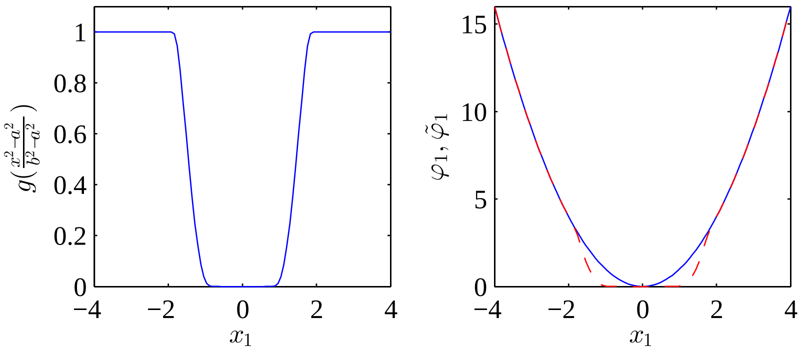

Figure 6: The left subplot shows the graph of $g\left(\frac{x_1^2-a^2}{b^2-a^2}\right)$ and the right subplot shows the graphs of $\varphi_1(x_1,x_2)$ (blue) and $\tilde{\varphi}_1(x_1,x_2)$ (dashed red).

The partitioning function shown in (37) *generates a complete abstraction and for this particular function $W^u(p)$ is a proper subset of $\tilde{\varphi}_1^{-1}(\tilde{\varphi}_1(p))$.*

Theorem 1 is vital in the field of abstracting generic dynamical systems, as it shows that a partitioning function generating a complete abstraction has to be constant on the unstable manifolds; hence, finding such a function is as difficult as finding the stable and unstable manifolds. This is practically impossible for systems of dimension greater than three. As an example, on all black lines in Figure 7 a complete partitioning function must be constant. In conclusion, research in the field should be focused on method development for generating sound rather than complete abstractions.

6 Conclusion

This paper highlights some issues with the use of transversal partitions. Complete abstractions of general dynamical systems cannot be generated using transversal partitioning functions. This issue arises as a partitioning function generating a complete abstraction has to be constant on the unstable manifolds.

The identification of stable and unstable manifolds is a very difficult problem. Therefore, it will not be possible to derive algorithms for generating complete abstraction of generic dynamical systems.

We have derived a theorem stating that the set of critical points of partitioning functions generating a complete abstraction must contain either the stable or unstable manifold of the dynamical system.

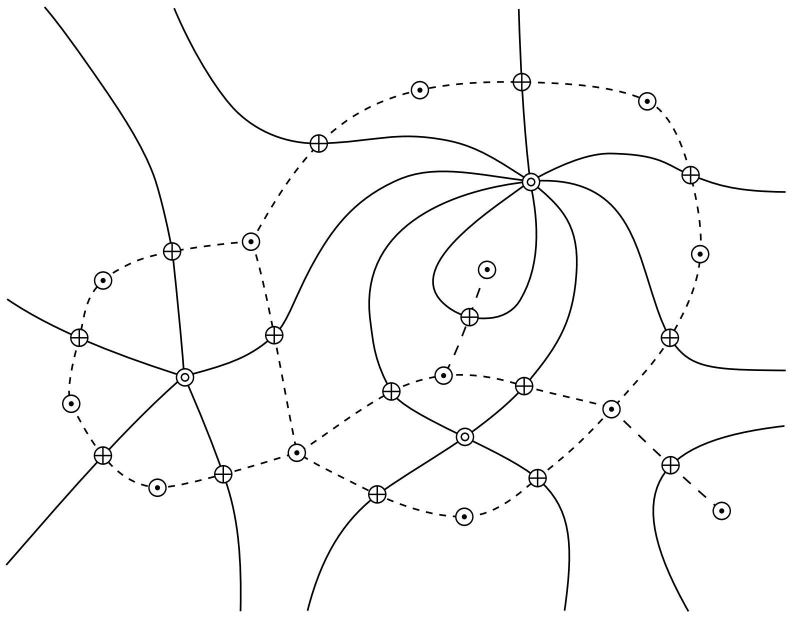

Figure 7: Morse complex, where a ”+” indicates a saddle, a ”·” indicates a maximum, and a ”◦” indicates a minimum. The lines connecting the singular points are stable and unstable manifolds.

This makes algorithmic generation of complete abstractions practically impossible for general dynamical systems.

References

[1] Alessandro Abate, Ashish Tiwari & Shankar Sastry (2009): *Box invariance in biologically-inspired dynamical systems*. Automatica 45(7), pp. 1601–1610, doi:10.1016/j.automatica.2009.02.028.

[2] Rajeev Alur & David L. Dill (1994): *A theory of timed automata*. Theoretical Computer Science 126(2), pp. 183–235, doi:10.1016/0304-3975(94)90010-8.

[3] Mireille Broucke (1998): *A Geometric Approach to Bisimulation and Verification of Hybrid Systems*. In: Proceedings of the 37th IEEE Conference on Decision and Control, Tampa, FL, USA, pp. 4277–4282, doi:10.1109/CDC.1998.761977.

[4] Goran Frehse, Colas Le Guernic, Alexandre Donzé, Scott Cotton, Rajarshi Ray, Olivier Lebeltel, Rodolfo Ripado, Antoine Girard, Thao Dang & Oded Maler (2011): *SpaceEx: Scalable Verification of Hybrid Systems*. In Ganesh Gopalakrishnan & Shaz Qadeer, editors: Computer Aided Verification, Lecture Notes in Computer Science 6806, Springer Berlin / Heidelberg, pp. 379–395, doi:10.1007/978-3-642-22110-1_30.

[5] Morris W. Hirsch, Stephen Smale & Robert L. Devaney (2004): *Differential Equations, Dynamical Systems & An Introduction to Chaos*, 2. edition. Elsevier.

[6] John M. Lee (2000): *Introduction to Smooth Manifolds*. Springer.

[7] John W. Milnor (1963): *Morse Theory*. Annals of Mathematics Studies 51 , Princeton University Press.

[8] Christoffer Sloth & Rafael Wisniewski (2012): *Abstractions for Mechanical Systems*. In: Proceedings of the 4th IFAC Workshop on Lagrangian and Hamiltonian Methods for Nonlinear Control, Bertinoro, Italy, pp. 96–101, doi:10.3182/20120829-3-IT-4022.00049.

[9] Christoffer Sloth & Rafael Wisniewski (2013): *Complete Abstractions of Dynamical Systems by Timed Automata*. Nonlinear Analysis: Hybrid Systems 7(1), pp. 80–100, doi:10.1016/j.nahs.2012.05.003.

[10] Ashish Tiwari (2008): *Abstractions for hybrid systems*. Formal Methods in System Design 32(1), pp. 57–83, doi:10.1007/s10703-007-0044-3.

[11] Loring W. Tu (2008): *An Introduction to Manifolds*. Springer, doi:10.1007/978-1-4419-7400-6.

[12] Rafael Wisniewski & Martin Raussen (2007): *Geometric analysis of nondeterminacy in dynamical systems*. Acta Informatica 43(7), pp. 501–519, doi:10.1007/s00236-006-0037-5.

A Timed Automata Details

The semantics of a timed automaton is defined in the following.

Definition 14 (Clock Valuation)**.** *A clock valuation on a set of clocks C is a mapping $v : C \to \mathbb{R}_{\geq 0}$. The initial valuation v_0 is given by $v_0(c) = 0$ for all $c \in C$. For a valuation v, a scalar $d \in \mathbb{R}_{\geq 0}$, and $R \subseteq C$, the valuations $v+d$ and $v[R]$ are defined as*

$$(v+d)(c) = v(c) + d, \tag{38a}$$

$$v[R](c) = \begin{cases} 0 & \textit{for } c \in R, \\ v(c) & \textit{otherwise}. \end{cases} \tag{38b}$$

It is seen that (38a) is used to progress time and (38b) is used to reset the clocks in the set R to zero. We denote the set of maps $v : C \to \mathbb{R}_{\geq 0}$ by $\mathbb{R}^C_{\geq 0}$.

Remark 1. *This notation indicates that we identify a valuation v with C-tuples of nonnegative reals in $\mathbb{R}^{|C|}_{\geq 0}$, where $|C|$ is the number of elements in C. We impose the Euclidian topology on $\mathbb{R}^C_{\geq 0}$.*

Definition 15 (Semantics of Clock Constraint)**.** *A clock constraint in $\Psi(C)$ is a set of clock valuations $\{v : C \to \mathbb{R}_{\geq 0}\}$ given by*

$$[\![c \bowtie k]\!] = \{v : C \to \mathbb{R}_{\geq 0} | v(c) \bowtie k\} \tag{39a}$$

$$[\![\psi_1 \wedge \psi_2]\!] = [\![\psi_1]\!] \cap [\![\psi_2]\!]. \tag{39b}$$

For convenience, we denote $v \in [\![\psi]\!]$ by $v \models \psi$ and denote the transition (e, v, σ, e', v') by $(e, v) \xrightarrow{\sigma} (e', v')$ in the following.

Definition 16 (Semantics of Timed Automaton)**.** *The semantics of a timed automaton given by the tuple $\mathscr{A} = (E, E_0, C, \Sigma, I, \Delta)$ is the transition system $[\![\mathscr{A}]\!] = (S, S_0, \Sigma \cup \mathbb{R}_{\geq 0}, T_\mathrm{s} \cup T_\mathrm{d})$, where S is the set of states*

$$S = \{(e, v) \in E \times \mathbb{R}^C_{\geq 0} |\ v \models I(e)\},$$

$S_0 \subseteq S$ is the set of initial states

$$S_0 = \{(e, v) \in E_0 \times \mathbb{R}^C_{\geq 0} |\ v = v_0\}.$$

Note that $E \times \mathbb{R}^C_{\geq 0}$ induces subspace topology on S.

$T_\mathrm{s} \cup T_\mathrm{d}$ is the union of the following sets of transitions

$$T_\mathrm{s} = \{(e, v) \xrightarrow{\sigma} (e', v') |\ \exists (e, G_{e \to e'}, \sigma, R_{e \to e'}, e') \in \Delta \textit{ such that } v \models G_{e \to e'} \textit{ and } v' = v[R_{e \to e'}]\},$$

$$T_\mathrm{d} = \{(e, v) \xrightarrow{d} (e, v + d) |\ \forall d' \in [0, d] : v + d' \models I(e)\}.$$

Hence, the semantics of a timed automaton is a transition system that comprises an infinite number of states: product of E and $\mathbb{R}^{C}_{\geq 0}$ and two types of transitions: the transition set T_{s} between discrete states with possibly a reset of clocks belonging to a subset $R_{e \to e'}$, and the transition set T_{d} that corresponds to time passing within the invariant $I(e)$.

In the following, we define an analog to the solution of a dynamical system for a timed automaton.

Definition 17 (Run of Timed Automaton)**.** *A run of a timed automaton $\mathscr{A}$ is a possibly infinite sequence of alternations between time steps and discrete steps of the following form*

$$(e_0, v_0) \xrightarrow{d_1} (e_0, v_1) \xrightarrow{\sigma_1} (e_1, v_2) \xrightarrow{d_2} \dots, \tag{40}$$

where $d_i \in \mathbb{R}_{\geq 0}$ and $\sigma_i \in \Sigma$.

By forcing alternation of time and discrete steps in Definition 17, the time step d_i is the maximal time step between the discrete steps σ_{i-1} and σ_i.

A.1 Trajectory of a Timed Automaton

A vital object for studying the behavior of any dynamical system is its trajectory. Therefore, we define a trajectory of a timed automaton [9]. At the outset, we introduce a concept of a time domain.

In the following, we denote sets of the form $\{a, \dots\}$ with $a \in \mathbb{Z}_{\geq 0}$ as $\{a, \dots, \infty\}$. Let $k \in \mathbb{N} \cup \{\infty\}$; a subset $\mathscr{T}_k \subset \mathbb{Z}_{\geq 0} \times \mathbb{R}_{\geq 0}$ with disjoint (union) topology will be called a time domain if there exists an increasing sequence $\{t_i\}_{i \in \{0, \dots, k\}}$ in $\mathbb{R}_{\geq 0} \cup \{\infty\}$ such that

$$\mathscr{T}_k = \bigcup_{i \in \{1, \dots, k\}} \{i\} \times T_i,$$

where

$$T_i = \begin{cases} [t_{i-1}, t_i] & \text{if } t_i < \infty \\ [t_{i-1}, \infty[& \text{if } t_i = \infty. \end{cases}$$

Note that $T_i = [t_{i-1}, t_i]$ for all i if $k = \infty$. We say that the time domain is infinite if $k = \infty$ or $t_k = \infty$. The sequence $\{t_i\}_{i \in \{0, \dots, k\}}$ corresponding to a time domain will be called a switching sequence.

We define two projections $\pi_1 : E \times \mathbb{R}^{C}_{\geq 0} \to E$ and $\pi_2 : E \times \mathbb{R}^{C}_{\geq 0} \to \mathbb{R}^{C}_{\geq 0}$ by $\pi_1(e, v) = e$ and $\pi_2(e, v) = v$.

Definition 18 (Trajectory)**.** *A trajectory of the timed automaton $\mathscr{A}$ is a pair $(\mathscr{T}_k, \gamma)$ where $k \in \mathbb{N} \cup \{\infty\}$ is fixed, and*

- *$\mathscr{T}_k \subset \mathbb{Z}_{\geq 0} \times \mathbb{R}_{\geq 0}$ is a time domain with corresponding switching sequence $\{t_i\}_{i \in \{0, \dots, k\}}$,*
- *$\gamma : \mathscr{T}_k \to S$ and recall that S is the (topological) space of joint continuous and discrete states, see Definition 16 and Remark 1. The map γ satisfies:*
  1. *For each $i \in \{1, \dots, k-1\}$, there exists $\sigma \in \Sigma$ such that*
  $$\gamma(i, t_i) \xrightarrow{\sigma} \gamma(i+1, t_i) \in T_{\mathrm{s}}.$$
  2. *Let $\mathbf{0}$ be a vector of zeros and $\mathbf{1}$ be a vector of ones in $\mathbb{R}^{C}$. For each $i \in \{1, \dots, k\}$*
  $$\pi_2(\gamma(i, t_{i-1} + d)) = \pi_2(\gamma(i, t_{i-1})) + d\mathbf{1} \quad \forall d \in \begin{cases} [0, t_i - t_{i-1}] & \text{if } t_i < \infty \\ [0, \infty[& \text{if } t_i = \infty \end{cases},$$
  where $\pi_2(\gamma(i, t_{i-1} + d)) \in [\![I(\pi_1(\gamma(i, t_i)))]\!]$ and $\pi_2(\gamma(1, t_0)) = \mathbf{0}$. (Item 2 ensures that the time derivative of the valuation of each clock is one, between the discrete transitions.)

Note that γ is continuous by construction. Recall the definition of v_0 from Definition 14. A trajectory at (e, v_0) (with $v_0 \models I(e)$) is a trajectory $(\mathscr{T}_k, \gamma)$ with $\gamma(1, t_0) = (e, v_0)$.

We define a discrete counterpart of the flow map.

Definition 19 (Flow Map of Timed Automaton)**.** *The flow map of a timed automaton $\mathscr{A}$ is a multivalued map*

$$\phi_{\mathscr{A}} : \mathbb{R}_{\geq 0} \times S_0 \to 2^S,$$

defined by $(e', v') \in \phi_{\mathscr{A}}(t; e, v_0)$ if and only if there exists a trajectory $(\mathscr{T}_k, \gamma)$ at (e, v_0) such that $t = t_k - t_0$ and $(e', v') = \gamma(k, t_k)$.

It will be instrumental to define a discrete flow map $\Phi_{\mathscr{A}} : \mathbb{R}_{\geq 0} \times E_0 \to 2^E$, which forgets the valuation of the clocks

$$\Phi_{\mathscr{A}}(t, e) = \pi_1 \circ \phi_{\mathscr{A}}(t; e, v_0) \tag{41}$$

In other words, $\Phi_{\mathscr{A}}$ is defined by: $e' \in \Phi_{\mathscr{A}}(t, e)$ if and only if there exists a run (40) of $[\![\mathscr{A}]\!]$ initialized in (e, v_0) that reaches the location e' at time $t = \sum_i d_i$.

The reachable set of a timed automaton is defined as follows.

Definition 20 (Reachable set of Timed Automaton)**.** *The reachable locations of a system $\mathscr{A}$ from a set of initial locations $E_0 \subseteq E$ on the time interval $[t_1, t_2]$ is defined as*

$$\Phi_{\mathscr{A}}([t_1, t_2], E_0) \equiv \bigcup_{(t,e) \in [t_1, t_2] \times E_0} \Phi_{\mathscr{A}}(t, e). \tag{42}$$