CORE
Metadata, citation and similar papers at core.ac.uk
Provided by OAR@UM

Department of Informatics

Relaxing Goodness is Still Good for SPDIs

Research Report No. 372

Gordon J. Pace

Gerardo Schneider

ISBN 82-7368-331-1
ISSN 0806-3036

February 2008 (Updated: Aug. 2008)

Relaxing Goodness is Still Good for SPDIs

Gordon J. Pace* Gerardo Schneider†

February 2008

Abstract

Polygonal hybrid systems (SPDIs) are planar hybrid systems, whose dynamics are defined in terms of constant differential inclusions, one for each of a number of polygonal regions partitioning the plane. The reachability problem for SPDIs is known to be decidable, but depends on the *goodness* assumption — which states that the dynamics do not allow a trajectory to both enter and leave a region through the same edge. In this paper we extend the decidability result to *generalised SPDIs* (GSPDI), SPDIs not satisfying the goodness property, and give an algorithmic solution to decide reachability of such systems.

1 Introduction

A hybrid system is one in which discrete and continuous behaviours interact. Some systems are inherently hybrid — consider a robot, with differential equations determining, for instance, its speed, together with an embedded computer taking discrete decisions based on the continuous input values coming from sensors. In other cases, a system consisting only of continuous behaviour, can be *hybridised,* introducing discrete behaviour in order to facilitate the analysis. For example, exact solutions can be difficult to obtain for a non-linear differential equation, making a qualitative and approximative analysis necessary.

In general, the main problem with the analysis of hybrid systems is that for properties such as reachability, their verification is undecidable. For instance,

*Dept. of Computer Science and AI, University of Malta, Msida, Malta. E-mail: gordon.pace@um.edu.mt

†Dept. of Informatics – Univ. of Oslo, P.O. Box 1080 Blindern, N-0316 Oslo, Norway. E-mail: gerardo@ifi.uio.no

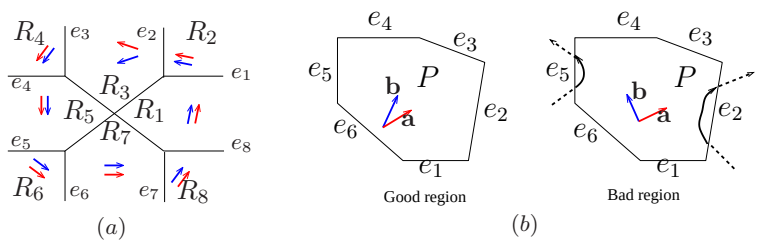

Figure 1: (a) Example of an SPDI; (b) Good and bad regions.

the reachability problem for planar PCDs (deterministic systems with Piecewise Constant Derivatives) is known to be decidable [MP93], but becomes undecidable in three and higher dimensions [AMP95]. Slight extensions of such decidable classes have been proved to be undecidable or equivalent to problems for which decidability or undecidability is not known [AS02, MP05]. An interesting class of hybrid systems for which the reachability question is known to be decidable, is the class of Polygonal Hybrid Systems (SPDIs) — a subclass of hybrid systems on the plane whose dynamics is defined by constant differential inclusions [ASY01, ASY07, Sch02]. Informally, an SPDI consists of a partition of the plane into polygonal regions, each of which enforces different dynamics given by two vectors determining the possible directions a trajectory might take; a simple SPDI is depicted in figure 1-(a). A constructive proof for deciding reachability on SPDIs can be found in [ASY07]. The proof is restricted to SPDIs which have the so-called *goodness* property — the dynamics of any region of the SPDI do not allow a trajectory to traverse any edge of the polygonal region in opposite directions (see figure 1-(b)). Technically this is equivalent to the property that the direction vector of each edge cannot be obtained as a positive linear combination of the vectors defining the dynamics. An SPDI without the goodness property is called *Generalised SPDI* (GSPDI).

In this paper, we present a constructive decidable algorithm for solving the reachability problem for GSPDIs. This decidability result contributes towards narrowing the undecidability frontier of low dimension hybrid systems [AS02, MP05]. At the same time our positive result allows GSPDIs to be used to approximate planar non-linear differential equations.

The paper is organised as follows. In the next section, we informally discuss the motivation behind relaxing goodness, and explain what technical problems arise when doing so. In section 3 we define the notation used, and outline definitions and results about SPDIs. Section 4 is concerned with the extension of these results to enable analysis of GSPDIs, including the decision algorithm for reachability. We conclude in the last section.

2 On Goodness

In this section we discuss informally why *goodness* is good for deciding the reachability problem of SPDI and what are the problems when relaxing it. More formal definitions will be given in section 3.
See figure 1-(b) for an example of a good and a 'bad' region (here 'bad' indicates that the region does not satisfy the goodness criterion). In the figure on the left we can see a good region, where the two vectors **a** and **b** make it impossible for a trajectory to enter and leave the region P through the same edge of the polygon delimiting the region. On the other hand, the figure on the right shows a bad region: Both e_2 and e_5 can be crossed in both directions by a trajectory entering and leaving P, as shown in the figure.

2.1 On Why Goodness is Good

The algorithm presented in [ASY07] for deciding reachability on SPDI depends on pre-processing of trajectory segments and a qualitative analysis to guarantee that it is possible to review the behaviour of all the possible *signatures*[1], by looking at only a finite set of abstract signatures. Informally, this is achieved as follows:

1. Trajectory segments are simplified — it is sufficient to look at trajectories made up of straight segments across regions, and which do not cross themselves.

2. Trajectory segments are abstracted into signatures, consisting of just the sequence of traversed edges. This result is based on the *Poincaré map* [HS74], that relates n-dimensional continuous-time systems with $(n-1)$-dimensional discrete-time systems.

3. It is shown that it is sufficient to look at signatures which consist only of sequences of edges and simple cycles.

4. Such signatures can be abstracted into *types of signatures* — signatures which do not take into account the number of times each simple cycle is iterated.

Many of the lemmas for proving that the above guarantee the finiteness of types of signatures critically depend on the goodness assumption, which propagate this dependency to the constructive proof given for deciding reachability of SPDIs which satisfies the goodness assumption.

[1] We call *signature* the sequence of traversed edges by the trajectory. A more formal definition will be given in a later section.

2.2 On Why We'd Rather Not be Good

Restricting oneself only to SPDIs satisfying the goodness assumption makes it very difficult to model real-life examples. Unfortunately, extending the SPDI model in most ways, such as allowing jumps with resets (from one edge to another remote one), increasing the number of dimensions and allowing non-linear differential inclusions, have been shown to make the model undecidable [AS02].

A potentially interesting and useful application of SPDIs is that of the approximation and analysis of two-dimensional non-linear differential equations. By splitting the plane into polygons, and by setting the dynamics of each polygon to be over-approximations of the non-linear differential equation in that region, one can ask reachability questions about the equation, and obtain answers accordingly. When over-approximating the dynamics, a negative reachability answer implies a negative answer in the exact equation. Using more and smaller polygons enables more precise approximations.

The problem with using this approach is that for most differential equations, using a fixed partition breaks the goodness assumption, since almost invariably, some edges of some regions will lie within the differential inclusion of that region. One solution would be to try to derive an intelligent partition of the plane which maintains goodness, which in some cases may be impossible, or by extending the SPDI analysis algorithms by relaxing the goodness assumption, thus enabling the modelling of non-linear differential equations in a straightforward manner.

As a simple example, consider a pendulum with friction coefficient k, mass M, pendulum length R and gravitational constant g. If θ is the angle subtended with the vertical, the behaviour of such a pendulum is described by the differential equation: $MR^2\ddot{\theta} + k\dot{\theta} + MgR\sin\theta = 0$. By taking $x = \theta$, and $y = \dot{\theta}$, we get $\dot{x} = y$ and $\dot{y} = -\frac{ky}{MR^2} - \frac{g\sin(x)}{R}$.

Using these formulae, we can produce SPDIs expressing these constraints, possibly with different plane partitions. Figure 2 gives two such partitions for $k = 1$, $R = 10$, $M = 10$, and $g = -10$. Visual inspection of the SPDIs, shows that various polygons fail the goodness assumption. By presenting an algorithm showing the decidability of reachability on Generalised SPDIs, we can automatically analyse such systems.

The main question is how much do we need to depend on the goodness assumption to prove decidability of reachability of SPDIs? In other words, let us consider GSPDIs, which are SPDIs without the goodness assumption. Is reachability still decidable? Reusing the decidability argument and algorithm for SPDIs to use directly on GSPDIs may appear to be the most straightforward approach to solve the problem. However, as it was proved in [Sch07]

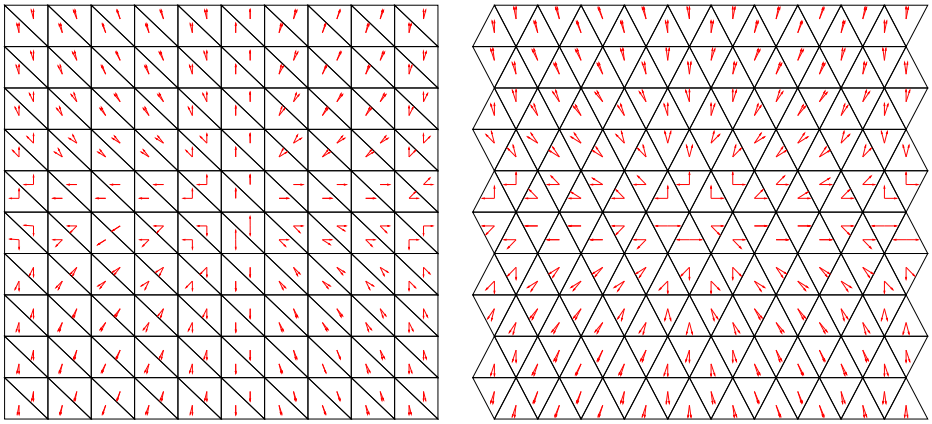

Figure 2: Approximating a non-linear differential equation describing a pendulum using different partitioning of the plane.

it is not possible to answer GSPDI reachability simply by using the SPDI reachability algorithm, since the algorithm is not complete and may give false negatives. The counter-example given in [Sch07] demonstrates that the SPDI algorithm fails to take into account certain trajectories entering and leaving a given region through the same edge. This indicates that simply by relaxing the goodness assumption and extending the proof for decidability of SPDIs cannot be achieved through such an approach.

One of the main problems when relaxing goodness is that a region can no longer be bi-partitioned into two sub-polygons were all the edges of the region in one sub-polygon can only be traversed in one direction, and all the edges in the other sub-polygon can be traversed only in the other direction. The goodness assumption permits a certain 'contiguity' of *entry-only* edges and *exit-only* edges belonging to two disjoint sub-regions. Certain lemmas and proofs of soundness of the reachability algorithm depend on this contiguity. If we relax goodness, we should be able to re-prove all such results without assuming the contiguity of entry-only and exit-only edges. Following this approach Schneider [Sch07] gives a terminating semi-decision algorithm for reachability analysis on GSPDIs.

In this paper, we take a different approach. Rather than try to reduce GSPDI reachability to SPDI reachability, augment the proof (and hence the algorithm) to work for cases not found in SPDIs. We give techniques to deal with these problematic cases, and prove that the number of abstract signatures (cases) to analyse can still be reduced to a finite one.

3 Polygonal Hybrid Systems (SPDIs)

In this section we recall the main definitions and concepts required in the rest of the paper, and give an outline of the results for SPDIs, upon which the results presented in this paper are built. For a more detailed presentation see [ASY07, Sch02].

In the rest of this section, we will use $\mathbf{a} = (a_1, a_2)$ and $\mathbf{x} = (x_1, x_2)$ to represent 2-dimensional vectors ($\mathbf{a}, \mathbf{x} \in \mathbb{R}^2$). An *angle* $\angle_{\mathbf{a}}^{\mathbf{b}}$ on the plane, defined by two non-zero vectors $\mathbf{a}$ and $\mathbf{b}$ is the set of all vectors lying between vector $\mathbf{a}$ and vector $\mathbf{b}$ in a counter-clockwise direction, and defined to be $\{\mathbf{x} \mid \theta(\mathbf{a}, \mathbf{x}) \leq \theta(\mathbf{a}, \mathbf{b})\}$, where $\theta(\mathbf{a}, \mathbf{b})$ is the angle between non-zero vectors $\mathbf{a}$ and $\mathbf{b}$ in the counter-clockwise direction.

Definition 1. *A* polygonal hybrid system *(SPDI) is a pair $\mathcal{H} = \langle \mathbb{P}, \mathbb{F} \rangle$, where $\mathbb{P}$ is a finite partition of the plane (with each $P \in \mathbb{P}$ being a convex polygon), called the* regions *of the SPDI, and $\mathbb{F}$ is a function which associates a pair of vectors to each polygon: $\mathbb{F}(P) = (\mathbf{a}_P, \mathbf{b}_P)$.*
In an SPDI every point on the plane has its dynamics defined according to which polygon it belongs to: if $\mathbf{x} \in P$, then $\dot{\mathbf{x}} \in \angle_{\mathbf{a}_P}^{\mathbf{b}_P}$. ■

Example 1. Consider the SPDI illustrated in figure 1-(a), with eight regions $R_1, R_2, \ldots, R_8$. A pair of vectors $(\mathbf{a}_i, \mathbf{b}_i)$ is also associated to each region R_i: $\mathbf{a}_1 = \mathbf{b}_1 = (1, 5)$, $\mathbf{a}_2 = \mathbf{b}_2 = (-1, \frac{1}{2})$, $\mathbf{a}_3 = (-1, \frac{11}{60})$ and $\mathbf{b}_3 = (-1, -\frac{1}{4})$, $\mathbf{a}_4 = \mathbf{b}_4 = (-1, -1)$, $\mathbf{a}_5 = \mathbf{b}_5 = (0, -1)$, $\mathbf{a}_6 = \mathbf{b}_6 = (1, -1)$, $\mathbf{a}_7 = \mathbf{b}_7 = (1, 0)$, $\mathbf{a}_8 = \mathbf{b}_8 = (1, 1)$. ■

We define $E(P)$ to be the set of edges of region P. We say that an edge e ($e \in E(P)$) is an *entry-only* of P if for all $\mathbf{x} \in e$ and for all $\mathbf{c} \in \angle_{\mathbf{a}_P}^{\mathbf{b}_P}$, $\mathbf{x} + \mathbf{c}\epsilon \in P$ for some $\epsilon > 0$. We say that e is an *exit-only* of P if the same condition holds for some $\epsilon < 0$. Intuitively, an entry-only (exit-only) edge of a region P allows at least a trajectory in P starting (terminating) on edge e, but allows no trajectories in P terminating (starting) on edge e. We write $In(P)$ ($In(P) \subseteq E(P)$) to denote the set of all entry-only edges of P and $Out(P)$($Out(P) \subseteq E(P)$) to denote the set of exit-only edges of P. From the definition, it follows immediately that no edge can be both an entry-only and an exit-only edge of a region: $In(P) \cap Out(P) = \emptyset$.

A region P is said to be *good*, if all the edges of that region are either entry-only or exit-only: $E(P) = In(P) \cup Out(P)$. An SPDI is said to be *good*, or satisfy the *goodness* property, if it consists of only good regions: $\forall P \in \mathbb{P} \cdot E(P) = In(P) \cup Out(P)$.

Assumption 1. *In the rest of this section, we will consider only good SPDIs.*

Example 2. In figure 1-(b), the region P shown on the left is good since all edges are either entry-only or exit-only. The region depicted on the right shows a region that is not good, since neither edge e_2 nor edge e_5 are in $In(P) \cup Out(P)$. ■

We will use the notation $e^P_{\circlearrowright}$ to indicate the directed edge e such that it follows a clockwise direction in region P, and similarly $e^P_{\circlearrowleft}$ to indicate the directed edge e following an anticlockwise direction in region P. Given a directed edge e, its inverse will be written as e^{-1}.

Definition 2. The set of directed edges of an SPDI $\mathcal{H}$ with partition $\mathbb{P}$, written $E_d(\mathcal{H})$, is defined to be:

$$E_d(\mathcal{H}) \;=\; \{e^P_{\circlearrowright} \mid P \in \mathbb{P},\; e \in In(P)\} \cup \{e^P_{\circlearrowleft} \mid P \in \mathbb{P},\; e \in Out(P)\}.$$

Similarly, we define $In_d(P)$ and $Out_d(P)$ to correspond to $In(P)$ and $Out(P)$ but with directed edges. ■

Since an edge typically appears in two adjacent regions, the direction induced in the two regions may be different. However, it was proved that edges which are entry-only in one region, and exit-only in the other result in matching induced directions: $e \in E_d(\mathcal{H})$ or $e^{-1} \in E_d(\mathcal{H})$, but not both [MP93, Sch02]. In an SPDI satisfying goodness, the only case where one can have both e and e^{-1} is when the dynamics of the regions result in e being either both an entry-only or an exit-only edge in the two adjacent regions it belongs to.
A *trajectory segment* of an SPDI $\mathcal{H}$, is a continuous function $\xi \in [0,T] \to \mathbb{R}^2$ such that for all $t \in [0,T]$, if $\xi(t) \in P$ and $\dot{\xi}(t)$ is defined then $\dot{\xi}(t) \in \angle^{\mathbf{b}_P}_{\mathbf{a}_P}$. The *signature* of a trajectory segment ξ, written $\mathsf{Sig}(\xi)$, is the ordered sequence of edges traversed by the trajectory, that is, $e_1, e_2, \ldots e_n$ resulting from $\xi \cap E_d(\mathcal{H})$.
One of the more important results presented in [ASY07] is that the behaviour of any trajectory is equivalent to the behaviour of some trajectory which does not cross itself and follows straight-line segments within regions.

Lemma 1. *Given a trajectory segment $\xi \in [0,T] \to \mathbb{R}^2$, there exists another trajectory segment $\xi' \in [0,T'] \to \mathbb{R}^2$ starting and finishing at the same points as ξ ($\xi(0) = \xi'(0)$ and $\xi(T) = \xi'(T')$) such that (i) ξ' does not cross itself (ξ is injective); and (ii) ξ' follows straight-line segments inside regions.* □

This result shows that to decide reachability, it is sufficient to look at non-self-crossing trajectories consisting of straight-line segments. In the rest of the discussion, we will restrict our use of trajectory to mean 'a non-self-crossing trajectory composed of straight-line segments between edges'. Similarly, the

term signature will be used to indicate the signature of a trajectory with these constraints. Note that the result is true of any SPDI, not only ones satisfying the goodness constraint.

3.1 Truncated Affine Multi-Valued Functions

An *affine* function $f \in \mathbb{R} \to \mathbb{R}$ is such that $f(x) = ax + b$. If $a > 0$ we say that f is *positive affine*, and if $a < 0$ we say that f is *negative affine*; we call this the parity of the affine function.
An *affine multivalued* function (AMF) $F \in \mathbb{R} \to 2^{\mathbb{R}}$, written $F = \langle f_l, f_u \rangle$, is defined by $F(x) = \langle f_l(x), f_u(x) \rangle$ where f_l and f_u are affine and $\langle \cdot, \cdot \rangle$ denotes an interval. For notational convenience, we do not make explicit whether intervals are open, closed, left-open or right-open, unless required for comprehension. For an interval $I = \langle l, u \rangle$ we have that $F(\langle l, u \rangle) = \langle f_l(l), f_u(u) \rangle$.
An *inverted affine multivalued* function $F \in \mathbb{R} \to 2^{\mathbb{R}}$, written $F = \langle f_l, f_u \rangle$, is defined by $F(x) = \langle f_u(x), f_l(x) \rangle$ where f_l and f_u are both negative affine and $\langle \cdot, \cdot \rangle$ denotes an interval.
Given an AMF F and two intervals $S \subseteq \mathbb{R}^+$ and $J \subseteq \mathbb{R}^+$, a *truncated affine multivalued* function (TAMF) $\mathcal{F}_{F,S,J} \in \mathbb{R} \to 2^{\mathbb{R}}$ is defined as follows: $\mathcal{F}_{F,S,J}(x) = F(x) \cap J$ if $x \in S$, otherwise $\mathcal{F}_{F,S,J}(x) = \emptyset$. In what follows we will write $\mathcal{F}$ instead of $\mathcal{F}_{F,S,J}$ whenever no confusion may arise. Moreover, in the rest of the paper F will always denote an AMF and $\mathcal{F}$ a TAMF. For convenience we write $\mathcal{F}(x) = F(\{x\} \cap S) \cap J$ instead of $\mathcal{F}(x) = F(x) \cap J$ if $x \in S$. We overload the application of a TAMF on an interval I: $\mathcal{F}(I) = F(I \cap S) \cap J$. We say that $\mathcal{F}$ is *normalised* if $S = \mathsf{Dom}(\mathcal{F}) = \{x \mid F(x) \cap J \neq \emptyset\}$ and $J = \mathsf{Im}(\mathcal{F}) = \mathcal{F}(S)$.
As in the case of affine multivalued functions, an *inverted truncated affine multivalued* function (inverted TAMF) is similar to a TAMF, but defined in terms of an inverted affine multivalued function as opposed to a normal one. An important result is that normal TAMFs are closed under composition.

Theorem 2. The composition of two normal TAMFs $\mathcal{F}_1(I) = F_1(I \cap S_1) \cap J_1$ and $\mathcal{F}_2(I) = F_2(I \cap S_2) \cap J_2$, is the TAMF $(\mathcal{F}_2 \circ \mathcal{F}_1)(I) = \mathcal{F}(I) = F(I \cap S) \cap J$, where $F = F_2 \circ F_1$, $S = S_1 \cap F_1^{-1}(J_1 \cap S_2)$ and $J = J_2 \cap F_2(J_1 \cap S_2)$. □

The following corollary extends the above result.

Corollary 3. *The composition of two inverted TAMFs gives a normal TAMF. Conversely, the composition of one normal and one inverted TAMF (in either order) gives an inverted TAMF.* □

To avoid having to reason about the length of every edge, we normalise every edge e such that its TAMF has the domain $[0, 1]$ (that is, the normalised

version of e has length 1, with 0 corresponding to the starting point of the directed edge, and 1 to the end point).

3.2 Successors

Given an SPDI, we fix a one-dimensional coordinate system on each edge to represent points lying on edges. For notational convenience, we will use e to denote both the directed edge and its one-dimensional representation. Accordingly, we write $\mathbf{x} \in e$ and $x \in e$, to mean "point $\mathbf{x}$ lies on edge e" and "coordinate x in the one-dimensional coordinate system of e" respectively. The same convention applied to sets of points of e represented as intervals (for example, $\mathbf{x} \in I$ and $x \in I$, where $I \subseteq e$) and to trajectories (for example, "ξ starting at x" or "ξ starting at $\mathbf{x}$").
Consider a polygon $P \in \mathbb{P}$, with $e_0 \in In_d(P)$ and $e_1 \in Out_d(P)$. For $I \subseteq e_0$, $\mathsf{Succ}_{e_0e_1}(I)$ is defined to be the set of all points lying on e_1 reachable from some point in I by a trajectory segment $\xi \in [0,t] \to \mathbb{R}^2$ in P (that is, $\xi(0) \in I \wedge \xi(t) \in e_1 \wedge \mathsf{Sig}(\xi) = e_0e_1$). Given $I = [l,u]$, $\mathsf{Succ}_{e_0e_1}(I) = F(I \cap S_{e_0e_1}) \cap J_{e_0e_1}$, where $S_{e_0e_1}$ and $J_{e_0e_1}$ are intervals, $F([l,u]) = \langle f_l(l), f_u(u) \rangle$ and f_l and f_u are positive affine functions. Successors are thus normal TAMFs.

3.3 Qualitative analysis of simple edge-cycles

Let $\sigma = (e_1 \dots e_k)$ be a simple edge-cycle — that is, a signature that can be repeated a number of times, and such that all edges are distinct ($e_i \neq e_j$ for all $1 \leq i < j \leq k$). Let $\mathsf{Succ}_\sigma(I) = F(I \cap S_\sigma) \cap J_\sigma$ with $F = \langle f_l, f_u \rangle$.
We assume that neither of the two functions f_l, f_u is the identity function. The following analysis, taken from [ASY01], allows us to calculate the behaviour of cycles provided that the path along the cycle has a normal (not inverted) TAMF. Since, in good SPDIs, the TAMF between a pair of edges is normal, and the composition of two normal TAMFs is itself a normal TAMF, this approach is universally applicable as long as the goodness assumption holds.
Let σ be a simple cycle, and l^* and u^* be the fix-points[2] of f_l and f_u, respectively, and $S_\sigma \cap J_\sigma = \langle L, U \rangle$. It can be shown that σ is of one of the following types:

STAY. The cycle is not abandoned neither by the leftmost nor the rightmost trajectory, that is, $L \leq l^* \leq u^* \leq U$.

[2]The fix-point x^* is the solution of $f(x^*) = x^*$, where $f(\cdot)$ is positive affine. The existence and computation of such fix-points are detailed in [ASY07].

DIE. The rightmost trajectory exits the cycle through the left (consequently the leftmost one also exits) or the leftmost trajectory exits the cycle through the right (consequently the rightmost one also exits), that is, $u^* < L \vee l^* > U$.

EXIT-BOTH. The leftmost trajectory exits the cycle through the left and the rightmost one through the right, that is, $l^* < L \wedge u^* > U$.

EXIT-LEFT. The leftmost trajectory exits the cycle (through the left) but the rightmost one stays inside, that is, $l^* < L \leq u^* \leq U$.

EXIT-RIGHT. The rightmost trajectory exits the cycle (through the right) but the leftmost one stays inside, that is, $L \leq l^* \leq U < u^*$.

The classification above provides useful information about the qualitative behaviour of trajectories. Any trajectory that enters a cycle of type DIE will eventually leave it after a finite number of turns. In the case of a cycle is of type STAY, all trajectories that happen to enter it will keep turning inside it forever. In all other cases, some trajectories will turn for a while and then exit, and others will continue turning forever. This information is crucial for solving the reachability problem for SPDIs. Also note that the above analysis gives us a non-iterative solution of cycle behaviour for most cycles. An important result to prove the decidability of SPDIs is that any valid signature can be expressed in a normal form, consisting of alternating sequential paths and simple loops:

Theorem 4. *Given an SPDI with the goodness constraint, any edge signature $\sigma = e_1 \dots e_p$ can be written as $\sigma_{\mathcal{A}} = r_1 s_1^{k_1} \dots r_n s_n^{k_n} r_{n+1}$, where for any $1 \leq i \leq n+1$, r_i is a sequence of pairwise different edges and for all $1 \leq i \leq n$, s_i is a simple cycle (no edges are repeated within s_i).* □

This representation of signatures is the base to obtain *types of signatures* with the following properties:

Lemma 5. *Given a good SPDI, let $\sigma = e_0 \dots e_p$ be a feasible signature, then its type, $\texttt{type}(\sigma) = r_1, s_1, \dots, r_n, s_n, r_{n+1}$ satisfies the following properties:*

$\mathbf{P_1}$ *For every $1 \leq i < j \leq n+1$, r_i and r_j are disjoint;*

$\mathbf{P_2}$ *For every $1 \leq i < j \leq n$, s_i and s_j are different.* □

The finiteness of the different types of signatures is the basis of the proof of decidability of (good) SPDI reachability, and of the termination of the reachability algorithm (together with acceleration results for simple loops).

Theorem 6. *The reachability problem in SPDIs satisfying the goodness constraint is decidable.*

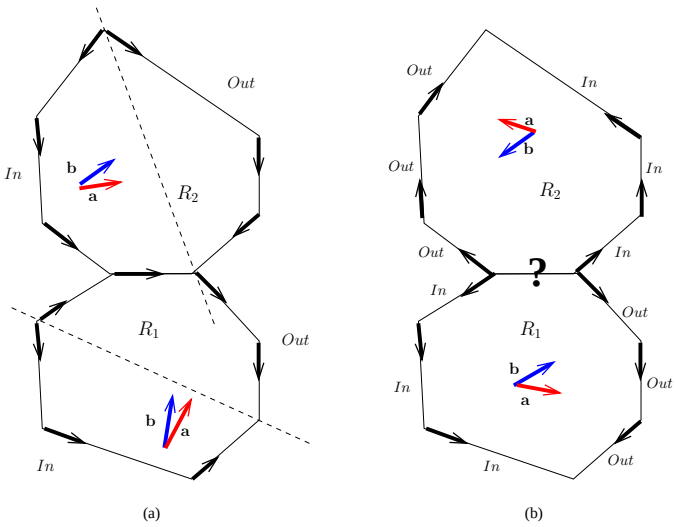

Figure 3: (a) An SPDI with matching order of edges; (b) a GSPDI showing that the order breaks the contiguity of the edge directions.

4 Relaxing Goodness: Generalised SPDIs

The original proof of the decidability of the reachability question for SPDIs, depended on the concept of monotonicity of TAMFs and their composition. Before starting the analysis, the algorithm fixed the direction of the edges separating regions. An interesting result guaranteed that the orientation of the edges resulted in each polygon split into two contiguous sequences of paths — one being the input edges, the other being the output edges. Furthermore, the orientation of an edge in one region is guaranteed to match the orientation of the same edge in the adjacent region[3], as shown in figure 3-(a). When one moves on to GSPDIs, *inout edges* (edges that may be traversed in both directions) break this result, since the direction of an edge when considered as an input edge clashes with the direction it is given when used as an output edge in the same region. The previous result however, still guaranteed that the entry-only edges and the exit-only edges can be assigned in one fixed direction (see figure 3-(b)).

To solve this problem, we use directed edges, and differentiate between the edge used as an input, and when it is used as an output, just as though they were two different edges in the GSPDI. Figure 4-(a) shows how an inout edge can be seen in this manner. Note that edge e_1 is an input edge in

[3]As we have already pointed out, there are special cases when an edge is an entry-only to a region and an exit-only to an adjacent regions. From the reachability point of view this does not cause any problem as these cases can be identified and treated accordingly.

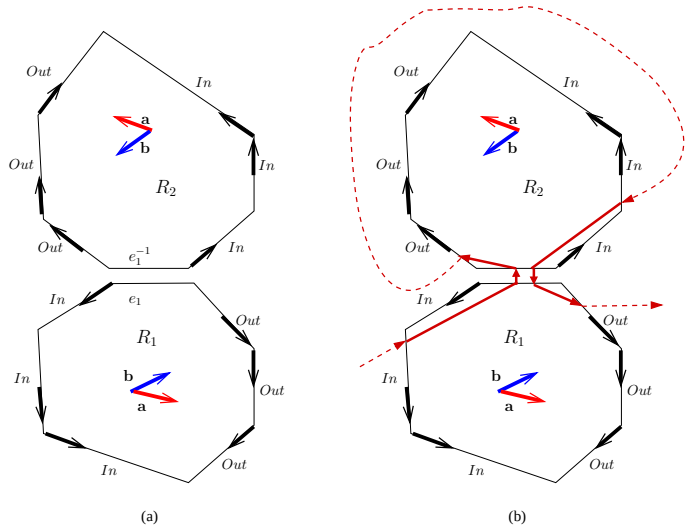

Figure 4: (a) A GSPDI with a duplicated inout edge; (b) a path through the GSPDI using edge e_1 in both directions.

region R_1, but an output edge in region R_2, and similarly, e_1^{-1} is an output edge in region R_1 and an input edge in region R_2. In other words, any path passing through the edge such as $\sigma = e_0 e_1 e_2 \dots e_3 e_1^{-1} e_4$ (see figure 4-(b)) can be analysed as before, and through monotonicity, one can deduce that Succ_σ is a positive TAMF. e_1 and e_1^{-1} are considered distinct edges, and the above path contains no loop.

It can be seen that the standard analysis for SPDIs works well for such cases. However, paths can now 'bounce' off an edge. Recall that any pair of edges $e_0 e_1$ is part of a path if e_0 is an input edge of a region, and e_1 is an output edge of the same region. One can calculate the TAMF for such a trajectory. However, ee^{-1} can now be a valid path, whose behaviour cannot be expressed as a normal TAMF. This breaks the analysis used in SPDIs, to accelerate the analysis of loops. The standard SPDI analysis thus needs to be extended to handle such 'bounces' in paths.

4.1 Preliminary Results

The *goodness* restriction (assumption 1) was originally introduced to simplify treatment of trajectories and to guarantee, amongst other things, that each region can be partitioned into *entry-only* and *exit-only* edges in an ordered way, a fact used in the proof of decidability of the reachability problem. We will introduce in this section further background, and provide new results concerning GSPDIs, needed to prove our decidability result.

Definition 3. *An edge $e \in P$ is an* inout *edge of P if e is neither an entry-only nor an exit-only edge of P.* ■

An SPDI without the goodness restriction is called a *Generalised SPDI (GSPDI)*. Thus, in GSPDIs there are three kinds of edges: inouts, entry-only and exit-only.
Self-crossing of trajectory segments of SPDIs can be eliminated which allow us to consider only non-crossing trajectory (segments). Lemma 1 (the full proof of which can be found in [ASY07]) also applies to GSPDIs. Therefore, in what follows, we will consider only trajectory segments without self-crossings.
Note that on GSPDIs, a trajectory can "intersect" an edge at an infinite number of points by *sliding* along it. A trace is thus no longer a sequence of points, but rather, a sequence of intervals.

Definition 4. *The* trace *of a trajectory ξ is the sequence $\texttt{trace}(\xi) = I_0 I_1 \dots I_n$ of the intersection intervals of ξ with the set of edges: $I_i \subseteq \xi \cap E_d(\mathcal{H})$.* ■

Definition 5. *An* edge signature *(or simply a* signature*) of a GSPDI is a sequence of edges. The* edge signature of a trajectory *ξ, $\mathsf{Sig}(\xi)$, is the ordered sequence of traversed edges by the trajectory segment, that is, $\mathsf{Sig}(\xi) = e_0 e_1 \dots e_n$, with $\texttt{trace}(\xi) = I_0 I_1 \dots I_n$ and $I_i \subseteq e_i$.* ■

Note that, in many cases, the intervals of a trace are in fact points. We say that a trajectory with edge signature $\mathsf{Sig}(\xi) = e_0 e_1 \dots e_n$ and trace $\texttt{trace}(\xi) = I_0 I_1 \dots I_n$ *interval-crosses* edge e_i if I_i is not a point.
Given a trajectory segment, we will distinguish between *proper inout* edges and *sliding* edges.

Definition 6. *Let ξ be a trajectory segment from point $\mathbf{x}_0 \in e_0$ to $\mathbf{x}_f \in e_f$, with edge signature $\mathsf{Sig}(\xi) = e_0 \dots e_i \dots e_n$, and $e_i \in E(P)$ be an edge of P. We say that e_i is a* sliding edge of P for ξ *if ξ interval-crosses e_i, otherwise e is said to be a* proper inout edge of P for ξ. ■

We say that a trajectory segment ξ *slides* along an edge e, if e is a sliding edge of P for ξ, and that ξ is a *sliding trajectory* if it contains at least one sliding edge.
The signatures that we will be analysing in GSPDIs are similar to ones in SPDIs, except that they may include consecutive inverted edges of the form ee^{-1}. The behaviour between such edges does not correspond to a normal TAMF, and thus has to be analysed separately.
One interesting property of inout edges is that the dynamics of the region they are in allow us to *slide* along the edge to one of the end-points of the edge.

Proposition 1. *If e is an inout edge, then any trajectory reaching the edge can always slide on the edge (in one or the other direction, or both).* $\square$

Note that in GSPDIs successors are not positive affine TAMFs as for SPDIs. A typical counter-example is a successor function $\mathsf{Succ}_{ee'}$ where e is an inout edge and e' is a parallel edge on the same region. It is clear that following at least one of the leftmost or rightmost vector will not intersect the edge e', thus giving an underlying AMF of the form $[ax+b, +\infty]$ or $[-\infty, ax+b]$ (or $[-\infty, +\infty]$). Extending successor to consider this kind of multi-valued functions would also need the extension of defining inverse AMFs and thus on how to compute the intervals S and J associated to the corresponding TAMFs. Note that the problem is also present in case only one of e or e' is inout.

In what follow we will show that though the above is true, it is always possible to find an equivalent successor expression (w.r.t. reachability) with the positive affine property. We assume in what follows that all edges are normalised (i.e., edges have local coordinates between 0 and 1).

The following lemma shows that the successor from e to e', whenever the dynamics is given by an angle greater than π, is indeed equal to the successor whenever the angle is 2π. This, in turn, can be expressed as a positive TAMF.

Lemma 7. *Let R be a region of a GSPDI $\mathcal{H}$ with dynamics given by $\angle_{\mathbf{a}}^{\mathbf{b}}$, with $\theta(\mathbf{a}, \mathbf{b}) > \pi$, and edges $e \in Inout(R)$ and $e' \in Out(R)$ (or $e' \in Inout(R)$). Then $\mathsf{Succ}_{ee'}^{\angle_{\mathbf{a}}^{\mathbf{b}}} = \mathsf{Succ}_{ee'}^{\angle_{\mathbf{0}}^{\mathbf{2}\pi}}$. Moreover, we can always express the successor as follows: $\mathsf{Succ}_{ee'}^{\angle_{\mathbf{a}}^{\mathbf{b}}}(x) = F(\{x\} \cap S) \cap J$, with $F = [x-\alpha, x+\alpha]$ (for any $\alpha > 1$), $S = [0, 1]$, and $J = [0, 1]$.* $\square$

The following result concerns the qualitative analysis of simple cycles containing at least one region with dynamics given by an angle greater than π.

Lemma 8. *Let R be a region of a GSPDI $\mathcal{H}$ with dynamics given by $\angle_{\mathbf{a}}^{\mathbf{b}}$, with $\theta(\mathbf{a}, \mathbf{b}) > \pi$. Then for any cyclic signature σ including edges in R , the cycle is EXIT-BOTH.* $\square$

The above follows directly from the definition of EXIT-BOTH and the previous lemma.

We now show how we can define a successor function as a positive TAMF which correctly describes the behavious between inout edges in regions with the vectors describing the dynamics within an angle π of each other. We start by defining a transformed successor with positive affine AMFs.

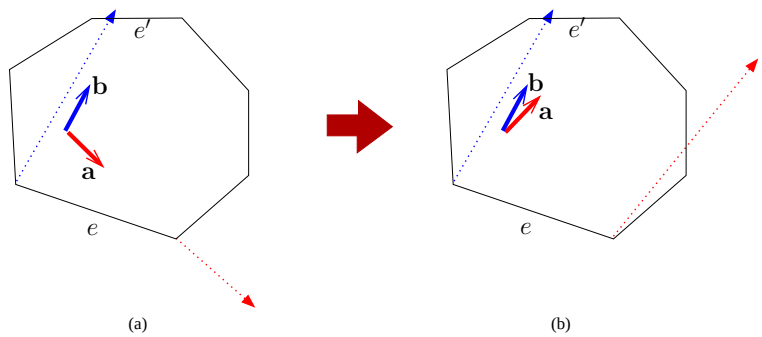

Figure 5: Transforming successors as in Def. 7: (a) $\mathsf{Succ}_{ee'}^{\angle_{\mathbf{a}}^{\mathbf{b}}}$; (b) $\mathsf{Succ}'^{\angle_{\mathbf{a}}^{\mathbf{b}}}_{ee'}$

Definition 7. *Let R be a region of a GSPDI $\mathcal{H}$ with dynamics given by $\angle_{\mathbf{a}}^{\mathbf{b}}$, with $\theta(\mathbf{a},\mathbf{b}) \leq \pi$, with at least one of the edges e and e' being an inout edge of R. We define $\mathsf{Succ}'^{\angle_{\mathbf{a}}^{\mathbf{b}}}_{ee'}(x)$, to be equal to $F(\{x\} \cap S) \cap J$, with S and J computed as for SPDIs, and F a positive affine AMF defined as follows:*

- $[f^{\mathbf{b}}(x), x+\alpha]$ *if* $\hat{\mathbf{a}} \cdot \mathbf{e} \leq 0$ *or* $\hat{\mathbf{b}} \cdot \mathbf{e}' \leq 0$ *(for any $\alpha > 1$ such that $x+\alpha$ defines a vector situated in a clock-wise position w.r.t. vector $\mathbf{b}$)*
- $[x-\alpha, f^{\mathbf{a}}(x)]$ *if* $\hat{\mathbf{b}} \cdot \mathbf{e} \leq 0$ *or* $\hat{\mathbf{a}} \cdot \mathbf{e}' \leq 0$ *(for any $\alpha > 1$ such that $x-\alpha$ defines a vector situated in an anticlock-wise position w.r.t. vector $\mathbf{a}$)*
- $[x-\alpha, x+\alpha]$ *if both conditions given in the previous items hold (for any $\alpha > 1$ such that $x-\alpha$ defines a vector situated in an anticlock-wise position w.r.t. the vector defined by $x+\alpha$)*
- $[f^{\mathbf{b}}(x), f^{\mathbf{a}}(x)]$ *otherwise.* □

See Fig. 5 for an example of the transformation done. Based on this definition, we can give a characterisation of successors in terms of positive affine AMFs:

Lemma 9. *Under the conditions in definition 7,* $\mathsf{Succ}_{ee'}^{\angle_{\mathbf{a}}^{\mathbf{b}}}(\{x\}) = \mathsf{Succ}'^{\angle_{\mathbf{a}}^{\mathbf{b}}}_{ee'}(\{x\})$. □

In the rest of the article we will always write $\mathsf{Succ}_{ee'}$ though it should be understood as being the primed version of the successor, in order to guarantee working always with positive affine underlying AMFs.
From all the above lemmas, we get that given a signature σ of a GSPDI $\mathcal{H}$, the successors can always be written in such a way that the underlying AMFs are positive affine. The argument is as follows. If the dynamics of region R is given by an angle $\angle_{\mathbf{a}}^{\mathbf{b}}$ greater than π, then we use lemma 7. If not, we use

lemma 9 which allows us to obtain positive affine AMFs by analysing all the different possible combinations of e and e' as inout edges with the dynamics under consideration.
We have then the following result.

Corollary 10. *In GSPDIs, for any two edges e_0 and e_1, $\mathsf{Succ}_{e_0 e_1}$ can always be written as a positive TAMF, whenever $e_1 \neq e_0^{-1}$.* □

A *bounce* is a part of a trajectory which crosses an edge twice in immediate succession. We define bounces formally within a signature as follows:

Definition 8. *Given a signature $\sigma = e_0 e_1 \dots e_n$, a pair of edges $e_i e_{i+1}$ is said to be a* bounce *if $e_{i+1} = e_i^{-1}$. We say that a signature $e_0 e_1 \dots e_n$ contains m bounces, if there are exactly m distinct indices $I = \{i_1,\ i_2,\ \dots i_m\}$ such for every $i \in I$, $e_i = e_{i+1}^{-1}$.* ■

Let $\mathsf{Flip}[l, u] = [1 - u,\ 1 - l]$ be an interval function. The following result establishes that the successor function for bounces can be defined in terms of the Flip function. The result follows directly from the definition of e^{-1}:

Lemma 11. *The behaviour of going from an edge e to its inverse e^{-1} is equivalent to* Flip*: $\mathsf{Succ}_{ee^{-1}} = \mathsf{Flip}$.* ■

One of the useful properties of SPDIs is that the successor function of any given signature is a normal TAMF. For GSPDIs, however, we need to take into account bounces, and hence analyse the composition of normal TAMFs with Flip:

Lemma 12. *Composing* Flip *with an inverted TAMF gives a normal TAMF and an inverted TAMF if we compose it with a normal TAMF.* □

The parity of the number of bounces occurring in a given signature influences the form of the underlying TAMF, as shown in the following result, whose proof follows immediately by induction on the number of bounces.

Corollary 13. *Any signature with an even number of bounces has its behaviour characterised by a normal TAMF, while a signature with an odd number of bounces is characterised by an inverted TAMF.* □

Given a simple cycle σ, let σ^+ be the cycle iterated one or more times. Recall that the analysis of simple cycle behaviour given for SPDIs depended only on the assumption that the TAMF of the cycle body is a normal one. From the previous result, it thus follows that whenever the number of bounces is even on a given cyclic signature, the composed TAMF is a normal one, meaning that the loop analysis can be conducted as for SPDIs:

Lemma 14. *Given a loop σ containing an even number of bounces, its iterated behaviour σ^+ can be calculated as for SPDIs.* □

Since we slide along inout edges, and can only bounce off inout edges, we can prove that loops which include at least one bounce are never STAY loops:

Lemma 15. *Loops which include bounces are not STAY loops.* □

This leaves only simple cycles with an odd number of bounces to be analysed. Considering the case when a bounce appears as the first pair of elements of a loop body, we can accelerate the analysis by running through the loop only once. The proof follows from the fact that the initial bounce enables a slide, thus allowing us to identify the limits through only one application of the loop body:

Lemma 16. *Given a signature $\sigma = e_0(e_1e_1^{-1}e_2 \dots e_n)^k e_1$ (i) with only one loop; (ii) with $k > 0$; (iii) which has an odd number of bounces; and (iv) starts with a bounce; the behaviour of signature is equivalent to following the loop only once as in $\sigma' = e_0e_1e_1^{-1}e_2 \dots e_ne_1$. In other words: $\mathsf{Succ}_\sigma = \mathsf{Succ}_{\sigma'}$.* □

Based on the above lemma, we can prove that any loop containing an odd number of bounces can be accelerated. The proof works by unwinding the loop body to push the first bounce to the beginning, and then applying the previous lemma:

Lemma 17. *Given a loop s with an odd number of bounces, we can calculate the limit of s^+ without iterating.* □

Therefore, we can now analyse any type of signature in GSPDIs using the results from corollary 10 (to deal with inout edges), and lemmas 14 and 17 (to deal with bounces).

Theorem 18. *We can compute the behaviour of a signature $r_1s_1^+\ r_2s_2^+ \dots r_n$.* □

4.2 Decidability

The following lemma guarantees that it is sufficient to consider simple cycles which occur in a type of signature only under certain patterns. Any type of signature containing two occurrences of the same simple cycle can be reduced to another type of signature where the simple cycle s occurs only once, provided the cycle with the edges in reverse order (denoted reverse(s)) does not occur between them. The proof is based on the fact that, assuming the path does not cross itself, between two instances of a repeated loop,

one can always find either (i) the reverse of the cycle; or (ii) a bounce. In the latter case, it can be shown that the bounce can be eliminated to avoid leaving the loop.

Lemma 19. *Given a GSPDI, and assuming only trajectories without self-crossing, if there is a type of signature where a simple cycle* $s = (e_0, e_1, \ldots, e_n)$ *appears twice, i.e.* $\mathtt{type}(\mathsf{Sig}(\xi)) = \sigma'\sigma''\sigma'''$ *with* $\sigma'' = s^k \ldots s^{k''}$*, then if there is no reverse*(s) *between the two occurrences of* s*, then* $\mathtt{type}(\mathsf{Sig}(\xi)) = \sigma' s^{k'''}\sigma'''$*.* □

We also prove that a trajectory which takes a loop (any number of times), then takes it again (once again any number of times) but in reverse order, and finally takes it a number of times in the forward direction, can be simulated by another trajectory which simply takes the loop a number of times. The proof is based on the fact that whichever direction the first edge of the simple cycle under consideration allows sliding in, it is possible to obtain a type of signature preserving reachability without such pattern.

Lemma 20. *Given a GSPDI, if there is a trajectory segment* $\xi : [0, T] \to \mathbb{R}^2$*, with* $\xi(0) = \mathbf{x}$ *and* $\xi(t) = \mathbf{x}'$ *for some* $t > 0$*, such that* $\mathtt{type}(\mathsf{Sig}(\xi)) = r_1 s_1^{k_1} r_2 s_2^{k_2} r_3 s_3^{k_3} r_4$*, with* $s_2 = s_1^{-1}$ *and* $s_3 = s_1$*, then it is always possible to find a trajectory segment* $\xi' : [0, T] \to \mathbb{R}^2$ *such that* $\xi'(0) = \mathbf{x}$ *and* $\xi'(t) = \mathbf{x}'$ *for some* $t > 0$*, and* $\mathtt{type}(\mathsf{Sig}(\xi)) = r_1 s_1^{k'_1} r'_4$*.* □

Based on these last two results, we can conclude now that for GSPDIs we can always transform a type of signature into one where simple loops are not repeated.

Corollary 21. *Given a GSPDI, let* σ *be an edge signature, then it can always be written as* $\sigma_{\mathcal{A}} = r_1 s_1^{k_1} \ldots r_n s_n^{k_n} r_{n+1}$*, where for any* $1 \leq i \leq n+1$*,* s_i *is a simple cycle (i.e., without repetition of edges), and for every* $1 \leq i < j \leq n$*,* s_i *and* s_j *are different.* □

The following lemma, ensuring that there are a finite number of types of signatures in GSPDIs, follows from the previous results and it is the basis for the termination proof of the reachability algorithm.

Corollary 22. *The number of different types of abstract signatures of a given GSPDI is finite.* □

4.3 Algorithm

Given a type of signature σ where each edge is traversed in exactly one direction, let $\mathbf{Reach}_\sigma(\mathbf{x}_0, \mathbf{x}_f)$ be the SPDI reachability algorithm presented

in [ASY07]. The reachability algorithm for a GSPDI $\mathcal{H}$, $\mathbf{Reach}(\mathcal{H}, \mathbf{x}_0, \mathbf{x}_f)$, consists of the following steps:

1. Generate the finite set of types of signatures $\Sigma = \{\sigma_0, \ldots, \sigma_n\}$ taking into account e and e^{-1} as different edges, and such that the loop signatures are all distinct;

2. Apply the function $\mathbf{Reach}_{\sigma_i}(\mathbf{x}_0, \mathbf{x}_f)$ for each $\sigma_i \in \Sigma$;

3. If for at least one $\sigma_i \in \Sigma$, $\mathbf{Reach}_{\sigma_i}(\mathbf{x}_0, \mathbf{x}_f) = \mathsf{Yes}$, then $\mathbf{Reach}(\mathcal{H}, \mathbf{x}_0, \mathbf{x}_f) = \mathsf{Yes}$, otherwise the answer is No.

We note that in step 2 we apply Succ progressively on the abstract signature, using lemmas 14 and 17 to compute the successor of a loop with bounces, and the Succ function as in the case of SPDIs for the rest. Based on these results, it is possible to show termination, correctness and completeness of GSPDI reachability.

Lemma 23. $\mathbf{Reach}(\mathcal{H}, \mathbf{x}_0, \mathbf{x}_f)$ *is a terminating, correct and complete algorithm calculating GSPDI reachability.* □

From this, the main theoretical result of our paper follows immediately:

Theorem 24. *The reachability problem for GSPDIs is decidable.* □

5 Conclusions

We have proved that the reachability question for GSPDIs is decidable. The proof is a constructive one, giving an algorithm which extends the one given in [ASY07] for SPDIs. The key lies in showing that the previous analysis works in all cases except when a loop contains an odd number of bounces. The algorithm is extended to deal with such cases. Inout edges enable sliding in one or both directions. Although the algorithm needs to be extended to deal with these cases, the overall effect of such edges is to accelerate the analysis of an SPDI, since (at least) one end of the edge is immediately covered once the edge is reached.
Reachability analysis of GSPDIs is not easy. In [Sch07] a semi-decision algorithm has been presented by reducing reachability of GSPDI to reachability of an exponential number of SPDIs. The main idea behind such an algorithm is that in most cases reachability is preserved when fixing inout edges as entry-only or exit-only edges, and then considering all the possible permutations of SPDIs generated from such a pre-processing, reducing then the problem

to SPDI reachability. The main problem with the approach is that there are cases where it is not possible to eliminate inout edges while preserving reachability.

The main contribution of our paper is interesting in a theoretical sense, since it extends the class of decidable hybrid systems, narrowing further the gap between what is known to be be decidable and what is known to be undecidable [AS02, MP05]. The result is, however, also interesting in a practical sense, since it provides a good foundation to approximate planar non-linear differential equations (as discussed in section 2.2). The next step is to implement the algorithm, extending the SPeeDI$^+$ tool [APSY02, Spe] to treat GSPDIs, and use in real case studies using non-linear differential equations.

Acknowledgements

We would like to thank Hallstein A. Hansen for finding an error in an earlier version of this paper.

References

[AMP95] E. Asarin, O. Maler, and A. Pnueli. Reachability analysis of dynamical systems having piecewise-constant derivatives. *TCS*, 138:35–65, 1995.

[APSY02] E. Asarin, G. Pace, G. Schneider, and S. Yovine. SPeeDI: a verification tool for polygonal hybrid systems. In *CAV'02*, volume 2404 of *LNCS*, pages 354–358. Springer-Verlag, July 2002.

[AS02] E. Asarin and G. Schneider. Widening the boundary between decidable and undecidable hybrid systems. In *CONCUR'02*, volume 2421 of *LNCS*, pages 193–208. Springer-Verlag, 2002.

[ASY01] E. Asarin, G. Schneider, and S. Yovine. On the decidability of the reachability problem for planar differential inclusions. In *HSCC'01*, number 2034 in LNCS, pages 89–104. Springer-Verlag, 2001.

[ASY07] E. Asarin, G. Schneider, and S. Yovine. Algorithmic Analysis of Polygonal Hybrid Systems. Part I: Reachability. *Theoretical Computer Science*, 379(1-2):231–265, 2007.

[HS74] M.W. Hirsch and S. Smale. *Differential Equations, Dynamical Systems and Linear Algebra*. Academic Press Inc., 1974.

[MP93] O. Maler and A. Pnueli. Reachability analysis of planar multilinear systems. In *CAV*, number 697 in LNCS, pages 194–209. Springer-Verlag, 1993.

[MP05] V. Mysore and A. Pnueli. Refining the undecidability frontier of hybrid automata. In *FSTTCS*, volume 3821 of *LNCS*. Springer-Verlag, 2005.

[Sch02] G. Schneider. *Algorithmic Analysis of Polygonal Hybrid Systems*. PhD thesis, VERIMAG – UJF, Grenoble, France, July 2002.

[Sch07] G. Schneider. On the decidability of the reachability problem for GSPDIs. Technical Report 359, Dept. of Informatics, Univ. of Oslo, June 2007.

[Spe] SpeeDI$^+$. `http://www.cs.um.edu.mt/speedi/`.

A Proofs of Lemmas, Theorems and Propositions

This appendix is for reviewing purposes only. It contains the detailed proofs of the results which could not be included for space reasons is being included at the end of the paper. Should the paper be accepted for publication, the main paper merged with the proofs will be published as a technical report for reference.

Section 3

Corollary 3 Composition of two TMAFs gives a TMAF. The composition of two inverted TAMFs gives a normal TAMF. Conversely, the composition of one normal and one inverted TAMF (in either order) gives an inverted TAMF.

Proof Sketch. The proof follows similarly to Theorem 2, where we note that the resulting affine functions are switched when one of the TAMFs is inverted, and the gradient is the product of the original two gradient values (hence positive when both positive or negative, and negative otherwise). □

Section 4.1

Proposition 1 If e is an inout edge, then any trajectory reaching the edge can always slide on the edge (in one or the other direction, or both).

Proof. The results follows from the fact that the director vector of e can be expressed as the positive linear combination of the two vectors of the region in consideration. □

Proposition 2. *If e_0 is an inout edge, then for any other edge e_1, and interval I, such that $\mathsf{Succ}_{e_1 e_0}(I)$ is not empty, all such applied successors include the left or all include the right end of the edge (equal to one of $(0, x\rangle$ or $\langle x, 1)$ for some value of x — depending on one of I's extremities).*

Proof. This is a direct consequence of Proposition 1. □

Lemma 12 Composition of the function Flip with an inverted TAMF results in a normal TAMF and in an inverted TAMF if we compose Flip with a normal TAMF.

Proof. Consider a normal TAMF f:

$$
\begin{aligned}
& (\mathsf{Flip}_\circ f)[x,y] \\
= \; & \{ \text{ by definition of TAMFs } \} \\
& \mathsf{Flip}([a_l x' + b_l, a_r y' + b_r] \cap J) \text{ where } [x',y'] = [x,y] \cap S \\
= \; & \{ \; J = [J_l, J_r] \text{ and by definition of intersection } \} \\
& \mathsf{Flip}[\max\{a_l x' + b_l, J_l\}, \min\{a_r y' + b_r, J_r\}] \text{ where } [x',y'] = [x,y] \cap S \\
= \; & \{ \text{ definition of } \mathsf{Flip} \; \} \\
& [1 - \min\{a_r y' + b_r, J_r\}, 1 - \max\{a_l x' + b_l, J_l\}] \text{ where } [x',y'] = [x,y] \cap S \\
= \; & \{ \text{ since } -\min\{x,y\} = \max\{-x,-y\}, \text{ similarly for max } \} \\
& [1 + \max\{-(a_r y' + b_r), -J_r\}, 1 + \min\{-(a_l x' + b_l), -J_l)] \text{ where } [x',y'] = [x,y] \cap S \\
= \; & \{ \text{ since } a + \max\{x,y\} = \max\{a+x, a+y\}, \text{ similarly for min } \} \\
& [\max\{1 - (a_r y' + b_r), 1 - J_r\}, \min\{1 - (a_l x' + b_l), 1 - J_l\}] \text{ where } [x',y'] = [x,y] \cap S \\
= \; & \text{ arithmetic } \} \\
& [\max\{-a_r y' - (1 + b_r), 1 - J_r\}, \min\{-a_l x' + (1 - b_l)), 1 - J_l\}] \text{ where } [x',y'] = [x,y] \cap S \\
= \; & \{ \text{ definition of intersection } \} \\
& [-a_r y' - (1 + b_r), -a_l x' + (1 - b_l)] \cap [1 - J_r, 1 - J_l] \text{ where } [x',y'] = [x,y] \cap S.
\end{aligned}
$$

Note that the result is also an inverted TAMF. The other result follows identically. □

Corollary 13 Any signature with an even number of bounces has its behaviour characterised by a normal TAMF, while a signature with an odd number of bounces is characterised by an inverted TAMF.

Proof. The proof follows by induction on the number of edges appearing in the signature.
The base case is when the signature consists of exactly two edges (shorter sequences of edges are not signatures by definition). Let the signature be $\sigma = e_0 e_1$. Now either (i) $e_1 = e_0^{-1}$, in which case we have an odd number (exactly one) bounce, and $\mathsf{Succ}_\sigma = \mathsf{Flip}$ (by definition 8) which is an inverted TAMF (by definition of Flip); or (ii) $e_1 \neq e_0^{-1}$, in which case we have an odd number of bounces (zero) and Succ_σ is a normal TAMF by the result in [ASY07]. In both cases, the result holds.
Now let us assume that the result holds for signatures of length n, and we will consider a signature of length $n+1$, namely: $\sigma = e_0 e_1 \ldots e_n$. Once again, either $e_n = e_{n-1}^{-1}$ or it is not. We will consider the cases separately:

- If $e_n = e_{n-1}^{-1}$, then the signature $e_0 e_1 \ldots e_{n-1}$ contains one bounce less that the original signature.

$$
\begin{array}{ll}
 & \mathsf{Succ}_{e_0 e_1 \ldots e_n} \\
= & \{ \text{ property of } \mathsf{Succ} \ \} \\
 & \mathsf{Succ} e_{n-1} e_n \circ \mathsf{Succ}_{e_0 e_1 \ldots e_{n-1}} \\
= & \{ \text{ definition of } \mathsf{Succ} \text{ on a bounce } \} \\
 & \mathsf{Flip} \circ \mathsf{Succ}_{e_0 e_1 \ldots e_{n-1}}
\end{array}
$$

Now, if $e_0 e_1 \ldots e_n$ has an even number of bounces, $e_0 e_1 \ldots e_{n-1}$ has an odd number of bounces (since the last pair were a bounce), and thus, by the inductive hypothesis, $\mathsf{Succ}_{e_0 e_1 \ldots e_{n-1}}$ is an inverted TAMF. But by the above equational reasoning, and lemma 12, it follows that $\mathsf{Succ}_{e_0 e_1 \ldots e_n}$ is a normal TAMF.

The case when $e_0 e_1 \ldots e_n$ has an odd number of bounces follows similarly.

- On the other hand, if $e_n \neq e_{n-1}^{-1}$, then the signature $e_0 e_1 \ldots e_{n-1}$ contains the same number of bounces as the original signature.

$$
\begin{array}{ll}
 & \mathsf{Succ}_{e_0 e_1 \ldots e_n} \\
= & \{ \text{ property of } \mathsf{Succ} \ \} \\
 & \mathsf{Succ} e_{n-1} e_n \circ \mathsf{Succ}_{e_0 e_1 \ldots e_{n-1}}
\end{array}
$$

As before, if $e_0 e_1 \ldots e_n$ contains an even number of bounces, so thus $e_0 e_1 \ldots e_{n-1}$ (since the last pair were not a bounce), and thus, by the inductive hypothesis, $\mathsf{Succ}_{e_0 e_1 \ldots e_{n-1}}$ is a normal TAMF. But by the above equational reasoning, and lemma 2, it follows that $\mathsf{Succ}_{e_0 e_1 \ldots e_n}$ is a normal TAMF.

The case when $e_0 e_1 \ldots e_n$ has an odd number of bounces follows similarly.

□

Lemma 15 Loops which include bounces are not STAY loops.

Proof. The proof follows from Proposition 1, which guarantees that once we reach the first inout edge, we can always slide to one end of the edge. Hence any loop containing such edge cannot be a STAY, by definition. □

Lemma 14 The behaviour of any loop σ containing an even number of bounces can be calculated as for SPDIs.

Proof. Corollary 13 ensures that Succ_σ is a normal TAMF. Earlier, in Section 3.3, we have summarised the analysis from [ASY01] which enables us to

calculate the behaviour of a cycle whose TAMF is not inverted, in a non-iterative manner. We can thus use this technique to calculate the iterated behaviour of σ in a non-iterative way. $\square$

Lemma 16 Given a signature with one loop $\sigma = e_0(e_1 e_1^{-1} e_2 \dots e_n)^k e_1$ (with $k > 0$, which has an odd number of bounces, and starts with a bounce), the behaviour of signature is equivalent to following the loop only once as in $\sigma' = e_0 e_1 e_1^{-1} e_2 \dots e_n e_1$. In other words: $\mathsf{Succ}_\sigma = \mathsf{Succ}_{\sigma'}$.

Proof. Since e_1 is an inout edge, by proposition 2, we know that we can slide in (at least) one direction. without loss of generality, let's assume that for any e, $\mathsf{Succ}_{ee_1}(I) = (0, x)$. Note that due to the definition of TAMFs, x is only dependant on the right bound of I.
Let $F = \mathsf{Succ}_{e_1^{-1} e_2 \dots e_n e_1}$. Since this includes an even number of bounces, composed TAMF (thus F) is a normal (non-inverted) TAMF. Moreover, since $F(I) = \mathsf{Succ}_{e_n e_1}(\mathsf{Succ}_{e_1^{-1} e_2 \dots e_n}(I))$, then $F(I) = (0, x)$ for some value of x. Finally, we note that since F is a normal TAMF, x is dependant only on the right bound of I, we can conclude that there exists α such that for any x, $F(x, 1) = (0, \alpha)$.
We can now proceed to prove the result by induction on k. Trivially, the result holds for $k = 1$. Now consider $k > 1$:

$$
\begin{aligned}
& \mathsf{Succ}_{e_0(e_1e_1^{-1}e_2\ldots e_n)^k e_1}(I) \\
= \; & \{\ k > 1\ \} \\
& \mathsf{Succ}_{e_0(e_1e_1^{-1}e_2\ldots e_n)(e_1e_1^{-1}e_2\ldots e_n)^{k-1} e_1}(I) \\
= \; & \{\ \text{by induction}\ \} \\
& \mathsf{Succ}_{e_0(e_1e_1^{-1}e_2\ldots e_n)(e_1e_1^{-1}e_2\ldots e_n)e_1}(I) \\
= \; & \{\ \text{by definition of } \mathsf{Succ} \text{ and } F\ \} \\
& F \circ \mathsf{Succ}_{e_1e_{-1}} \circ F\mathsf{Succ}_{e_1e_{-1}} \circ \mathsf{Succ}_{e_0e_1}(I) \\
= \; & \{\ \text{by definition of } \mathsf{Flip}\ \} \\
& F \circ \mathsf{Flip} \circ F \circ \mathsf{Flip} \circ \mathsf{Succ}_{e_0e_1}(I) \\
= \; & \{\ \text{by sliding argument given above}\ \} \\
& F \circ \mathsf{Flip} \circ F \circ \mathsf{Flip}(0, x) \\
= \; & \{\ \text{by definition of } \mathsf{Flip}\ \} \\
& F \circ \mathsf{Flip} \circ F(1-x, 1) \\
= \; & \{\ \text{by property of } F \text{ given above}\ \} \\
& F \circ \mathsf{Flip}(0, \alpha) \\
= \; & \{\ \text{by definition of } \mathsf{Flip}\ \} \\
& F(1-\alpha, 1) \\
= \; & \{\ \text{by property of } F \text{ given above}\ \} \\
& (0, \alpha) \\
= \; & \{\ \text{by property of } F \text{ given above}\ \} \\
& F(1-x', 1) \\
= \; & \{\ \text{by definition of } \mathsf{Flip}\ \} \\
& F \circ \mathsf{Flip}(0, x') \\
= \; & \{\ \text{by sliding argument given above}\ \} \\
& F \circ \mathsf{Flip} \circ \mathsf{Succ}_{e_0e_1}(I) \\
= \; & \{\ \text{by definition of } \mathsf{Flip}\ \} \\
& F \circ \mathsf{Succ}_{e_1e_1^{-1}} \circ \mathsf{Succ}_{e_0e_1}(I) \\
= \; & \{\ \text{by definition of } \mathsf{Succ} \text{ and } F\ \} \\
& \mathsf{Succ}_{e_0(e_1e_1^{-1}e_2\ldots e_n)e_1}(I)
\end{aligned}
$$

By induction the result thus follows.

□

Lemma 17 Given a loop σ with an odd number of bounces, we can calculate the limit of σ^+ without iterating.

Proof. Let $\sigma = e_0e_1 \ldots e_ie_i^{-1}e_{i+1} \ldots e_n$, where $e_ie_i^{-1}$ is the first bounce of the sequence. Since σ contains inout edges, it cannot be a STAY loop, and we only consider the case where the loop finally exits. Consider the exiting loop σ^+e'.

The case when σ is never repeated or repeated only once, can be easily handled. When the number of repetitions is at least twice, we can use the following reasoning:

$$
\begin{array}{ll}
 & \mathsf{Succ}_{\sigma^k e'} \\
= & \{ \text{ definition of } \sigma \ \} \\
 & \mathsf{Succ}_{(e_0 e_1 \ldots e_i e_i^{-1} e_{i+1} \ldots e_n)^k e'} \\
= & \{ \text{ definition of path repetition } \} \\
 & \mathsf{Succ}_{e_0 e_1 \ldots e_{i-1} (e_i e_i^{-1} e_{i+1} \ldots e_n e_0 e_q \ldots e_{i-1})^{k-1} e_i e_i^{-1} e_{i+1} \ldots e'} \\
= & \{ \text{ using Lemma 16 } \} \\
 & \mathsf{Succ}_{e_0 e_1 \ldots e_{i-1} (e_i e_i^{-1} e_{i+1} \ldots e_n e_0 e_q \ldots e_{i-1}) e_i e_i^{-1} e_{i+1} \ldots e_n e'}
\end{array}
$$

This reduces the analysis of such loops to a simple path analysis which we know how to perform. □

Theorem 18 We can (constructively) compute the behaviour of a signature $r_1 s_1^+ r_2 s_2^+ \ldots r_n$.

Proof. We use the standard techniques presented in [ASY07], but use Theorems 14 and 17 to analyse loops with bounces. □

Section 4.2

Lemma 19 Given a GSPDI, and assuming only trajectories without self-crossing, if there is a type of signature where a simple cycle $s = (e_0, e_1, \ldots, e_n)$ appears twice, i.e. $\mathtt{type}(\mathsf{Sig}(\xi)) = \sigma' \sigma'' \sigma'''$ with $\sigma'' = s^k \ldots s^{k''}$, then if there is no reverse(s) between the two occurrences of s, then $\mathtt{type}(\mathsf{Sig}(\xi)) = \sigma' s^{k'''} \sigma'''$.

Proof Sketch. There are two cases:

1. $\sigma'' = s^k r s^{k''}$**:** In this case r must be of the form $e_n^{-1} e_{n-1}^{-1} \ldots e_i^{-1}$ with $i > 0$. We must have a bouncing at e_i^{-1}, then we can slide and we get $\sigma'' = s^{k'''}$.

2. $\sigma'' = s^k \omega s^{k''}$**:** Here ω is any finite sequence of alternating r's and s's. It can be shown that either we reduce to the previous case, or ω must contain reverse(s), or there must be a self-crossing. □

Lemma 20 Given a GSPDI, if there is a trajectory segment $\xi : [0, T] \to \mathbb{R}^2$, with $\xi(0) = \mathbf{x}$ and $\xi(t) = \mathbf{x}'$ for some $t > 0$, such that $\mathtt{type}(\mathsf{Sig}(\xi)) = r_1 s_1^{k_1} r_2 s_2^{k_2} r_3 s_3^{k_3} r_4$, with $s_2 = \text{reverse}(s_1)$ and $s_3 = s_1$, then it is always possible to find a trajectory segment $\xi' : [0, T] \to \mathbb{R}^2$ such that $\xi'(0) = \mathbf{x}$ and

$\xi'(t) = \mathbf{x}'$ for some $t > 0$, and $\mathtt{type}(\mathsf{Sig}(\xi)) = r_1 s_1^{k'_1} r'_4$.

Proof Sketch. Let $s_1 = (e_0, e_1, \ldots, e_n)$ be a simple cycle where ξ is a clockwise spiral turning inwards. Due to Proposition 1, we have the following two cases:

1. e_0 **allows sliding inwards.** We can always eliminate the first s, i.e., $\mathtt{type}(\mathsf{Sig}(\xi)) = r'_1 s_2^{k_2} r_3 s_3^{k_3} r_4$. See Fig. 8.

2. e_0 **allows sliding outwards.** Two cases:

   (a) **reverse**(s_1) **loops outwards.** In this case we can eliminate s_2 since once we start that loop, we can slide outwards till s_3 starts, and we get $\mathtt{type}(\mathsf{Sig}(\xi)) = r_1 s_1^{k_1} s_3^{k_3} r_4$, which is $\mathtt{type}(\mathsf{Sig}(\xi)) = r_1 s_1^{k'} r_4$. See Fig. 6-(a).

   (b) **reverse**(s_1) **loops inwards.** Two cases:

      i. r_2 **contains only edges in** s_1 **and** s_2**.** This implies bouncing. Two cases.

         A. **Bouncing inwards.** Implies sliding inwards, which contradicts the assumption. See Fig. 7-(a).

         B. **Bouncing outwards.** Implies reverse(s_1) must loop outwards, contradicts the assumption that reverse(s_1) loops inwards. See Fig. 7-(b).

      ii. r_2 **contains edges not in** s_1 **and** s_2**.** This means that the trajectory exit s_1 through the 'right'. Let us assume the last visited point in s_1 is $\mathbf{x} \in e_n$, and that $\mathbf{x}' \in e$ such that $\xi(t) = \mathbf{x}$ and $\xi(t') = \mathbf{x}'$ with $\mathsf{Sig}(\xi[t..t']) = e_n e$, where $e \in first(r_2)$. Then the segment of line $\overline{\mathbf{x}\mathbf{x}'}$ partition the region R into two subregions R_1 and R_2. Clearly the only way to have $r_2 s_2$ with s_2 going inwards is from a trajectory segment from region R_1 to R_2 by crossing $\overline{\mathbf{x}\mathbf{x}'}$, which breaks the assumption of non-crossing trajectories. Thus the pattern $s_2 r_3 s_3$ is not possible in this case. See Fig. 6-(b). □

Corollary 21 Given a GSPDI, let $\sigma = e_1 \ldots e_p$ be an edge signature, then it can always be written as $\sigma_{\mathcal{A}} = r_1 s_1^{k_1} \ldots r_n s_n^{k_n} r_{n+1}$, where for any $1 \leq i \leq n+1$, s_i is a simple cycle (i.e., without repetition of edges), and for every $1 \leq i \neq j \leq n$, s_i and s_j are different.

Proof. If there are $i \neq j$ such that $s_i = s_j$, the only possibility is to satisfy or the assumptions of Lemma 19 or Lemma 20. In both cases we can always obtain a signature without repeating s_i. □

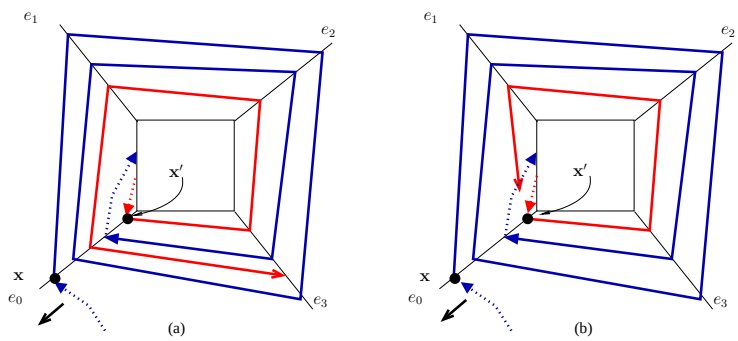

Figure 6: Proof of Lemma 20 - Case sliding outwards: (a) case reverse(s) looping outwards; (b) case reverse(s) looping inwards and exiting.

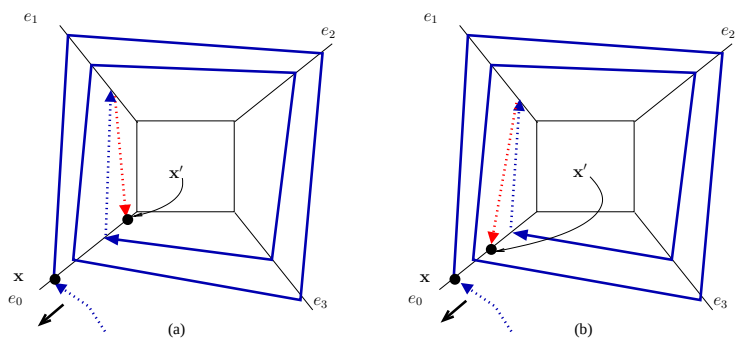

Figure 7: Proof of Lemma 20 - Case sliding outwards: (a) case bouncing inwards; (b) case bouncing outwards.

Corollary 22 The number of different types of abstract signatures of a given GSPDI is finite.

Proof. Based on Lemma 21, it suffices to analyse signatures of the form $\sigma_{\mathcal{A}} = r_1 s_1^+ \dots r_n s_n^+ r_{n+1}$ such that provided that $i \neq j$, $s_i \neq s_j$ and with each r_k containing no repeated edges. Hence, since the number of edges is finite, the number of possible values each r_k can take is finite. Similarly, the number of distinct simple loops is finite. Therefore, the number of abstract signatures to analyse is finite. □

Section 4.3

Lemma 23 **Reach**$(\mathcal{H}, \mathbf{x}_0, \mathbf{x}_f)$ is a terminating, correct and complete algorithm calculating GSPDI reachability.

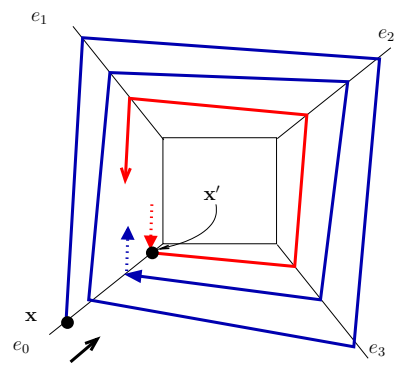

Figure 8: Proof of Lemma 20 –Case sliding inwards.

Proof. Termination of step 1 follows from the fact that GSPDIs have finite partitions. Step 2 terminates by corollary 22. Using Theorems 17 and 14 we can also compute steps 3 and 4, hence guaranteeing termination of the algorithm.
Correctness of the algorithm follows from Theorems 17 and 14 (on accelerating loops with bounces) and the results in [Sch02, ASY07] on the correctness of SPDI reachability checking.
Finally, completeness is guaranteed by Theorem 22.
Therefore, $\mathbf{Reach}(\mathcal{H}_i, \mathbf{x}_0, \mathbf{x}_f)$ (for all $\mathcal{H}_i \in \mathcal{H}_{red}$, $1 \leq i \leq n$), is a terminating complete and sound algorithm for deciding GSPDI reachability. □

View publication stats