UNIVERSITY OF OSLO
Department of Informatics

On the Decidability of the Reachability Problem for GSPDIs

Research Report No. 359

Gerardo Schneider

Isbn 82-7368-317-6
Issn 0806-3036

June 2007

On the Decidability of the Reachability Problem for GSPDIs

Gerardo Schneider

Department of Informatics, University of Oslo
PO Box 1080 Blindern, N-0316 Oslo, Norway
E-mail: {gerardo at ifi.uio.no}

Abstract

Polygonal hybrid systems (SPDIs) are a subclass of hybrid systems whose dynamics is defined by constant differential inclusions, for which the reachability problem is decidable. The decidability result is based, among other things, on the fact that a trajectory cannot enter and leave a given region through the same edge. An SPDI satisfying the above restriction is said to have the *goodness* property. In a previous work we have given a misleading proof sketch of decidability of reachability for SPDIs when relaxing goodness. In this work we give a counter-example to such proof and we give an algorithm for semi-deciding reachability of such class of systems.

1 Introduction

An interesting and still decidable (w.r.t reachability) class of hybrid systems is the so-called Polygonal Hybrid System (SPDI for short, [ASY01, ASY07, Sch02]) which is a subclass of hybrid systems on the plane whose dynamics is defined by constant differential inclusions. SPDIs are a generalization of PCDs (deterministic systems with Piece-wise Constant Derivatives) for which it has been shown that the reachability problem is decidable for the planar case [MP93] but undecidable for three and higher dimensions [AMP95]. Slight extensions of such decidable classes have been proved to be undecidable or equivalent to a problem for which decidability or undecidability is not known [AS02, MP05].
The constructive proof for deciding reachability on SPDI given in [ASY01] (see also [ASY07] and [Sch02, Chap. 5]) relies, among other things, on the

fact that SPDIs have the *goodness* property, i.e. the dynamics of any region of the SPDI (location of the corresponding automaton) does not allow a trajectory to traverse any edge of the polygon defining the region in both directions. Technically this means that the director 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 *General SPDI* –or GSPDI for short. We have wrongly claimed in [Sch02, Chap. 9] that the reachability problem for GSPDI is decidable. The proof sketch was conducted by proving that any GSPDI can be reduced to a set of SPDIs, preserving reachability. The proof sketch, as presented, is not completely wrong but incomplete, letting the decidability conclusion to be still inconclusive. Unfortunately we have discovered such mistake in September 2002, just few months after the final print of the thesis. We considered it was not worth publishing a refutation of the result at that moment since there was no research being conducted in that direction then. We revived our interest on the subject again only recently due to the publication of the paper [MP05], in which the frontier between decidable and undecidable hybrid systems is revisited, to refine previous result given in [AS02]. The decidability of reachability of GSPDIs would have contributed to narrow the undecidability frontier; with the result presented here we let it still open, unfortunately.

In this paper we provide a counter-example to the claim of the decidability of the reachability problem for GSPDIs given in [Sch02, Chap. 9], which remain thus an open problem. We prove, indeed, that GSPDI reachability cannot be reduced to SPDI reachability. We rephrase the results given in [Sch02] to give a semi-decidable algorithm for solving the reachability problem for GSPDIs.

The paper is organized as follows. In next section we explain informally the problems arising when relaxing goodness while in Section 3 we give some preliminaries, providing useful notation and definitions and recalling the definition of SPDI. In Section 4 we present GSPDIs. Section 5 is concerned with the analysis of trajectories, providing some results needed to establish the semi-decision algorithm for reachability presented in Section 6. 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 Fig. 1 for an example of a good and a 'bad' region (here 'bad' stands for a region not satisfying the goodness criteria). In the left side of the figure we can see a good region, where the two vectors **a** and **b** determine the impossibility of 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.

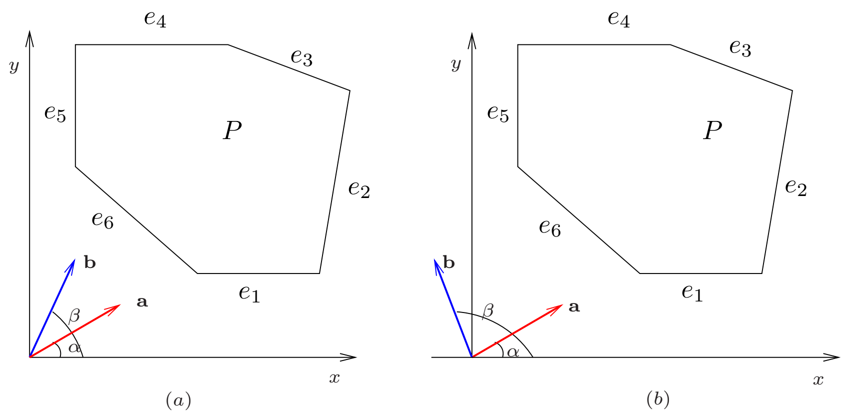

Figure 1: a) A good region. b) A bad region.

2.1 Why Goodness is Good?

The algorithm presented in [Sch02] for deciding reachability on SPDI heavily depends on the pre-processing of trajectory segments to guarantee that it is possible to list all the possible sets of *signatures*, i.e., those sequences of edges of the SPDI traversed by all the possible trajectories between two points. This is of course not possible in general as there are infinitely many such trajectories. However, a qualitative analysis allows to prove that indeed there are a finite number of *types of signatures*, that are kind of abstract signatures that preserve the reachability property.
Briefly, the above is achieved by performing the following steps.

1. Simplification of trajectory segments: straightening them and removing self-crossings. Given an arbitrary trajectory segment from one point to another, we show how to get a piecewise constant derivative trajectory segment without self-crossing.

2. Abstraction of trajectory segments into *signatures*, considering the sequence of traversed edges. This result is based on the *Poincaré map* [HS74,

NS60], that relates n-dim continuous-time systems with $(n-1)$-dim discrete-time systems.

3. *Factorization* of signatures in a convenient way, having only sequences of edges and simple cycles. This factorization allows to have a nice representation of signatures.

4. Abstraction of factorized signatures into *types of signatures*, that are signatures without taking into account the number of times each simple cycle is iterated.

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

2.2 Why Relaxing Goodness is not so Good?

The main question now 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 the new class of polygonal hybrid systems, GSPDI, obtained by relaxing goodness in SPDI. Is reachability still decidable? From the above discussion we are let with the following two alternatives:

1. Adapt the proofs of decidability for SPDIs to GSPDIs. This would imply to restate the proofs to make them independent of the goodness assumption.

2. Provide a completely new decidability proof for GSPDI. This will probably need to use different techniques and results than the ones used for SPDIs.

The first alternative above seems the most straightforward and easy to do. However, as we will show later it is not possible to reduce GSPDI reachability to SPDI reachability. This is done by proving that it is not in general possible to simplify certain trajectories entering and leaving a given region through the same edge, to trajectories behaving as in SPDIs. One of the main problems when relaxing goodness is that a region cannot be bi-partitioned anymore into two semi-planes were all the edges in one semi-plane can be traversed only in one direction, w.r.t. the region, and all the edges in the other semi-plane can be traversed only in the other direction. That is, the goodness assumption permit a certain 'contiguity' of *entry* edges and *exit* edges belonging to two

disjoint sub-regions (see Fig. 8). Some 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 and exit edges.
This let us with the second alternative. Unfortunately, to date we have not succeeded in providing a proof of decidability (nor of undecidability) to the reachability problem on GSPDIs.
On the other hand and as stated in the introduction, we will show that we can relax the goodness assumption as to give a terminating semi-decision algorithm for reachability analysis on GSPDIs.

3 Preliminaries

This section is more technical, recalling the main definitions and concepts needed to understand the rest of the paper. For a more detailed presentation see [ASY07, Sch02].

3.1 SPDI

Let $\mathbf{a} = (a_1, a_2), \mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ and $\alpha, \beta \in \mathbb{R}$. The *inner product* of two vectors $\mathbf{a} = (a_1, a_2)$ and $\mathbf{x} = (x_1, x_2)$ is defined as $\mathbf{a} \cdot \mathbf{x} = a_1x_1 + a_2x_2$. We denote by $\hat{\mathbf{x}}$ the vector $(x_2, -x_1)$ obtained from $\mathbf{x}$ by rotating clockwise by the angle $\pi/2$. Notice that $\mathbf{x} \cdot \hat{\mathbf{x}} = 0$.
An *angle* $\angle_{\mathbf{a}}^{\mathbf{b}}$ on the plane, defined by two non-zero vectors $\mathbf{a}, \mathbf{b}$ is the set of all positive linear combinations $\mathbf{x} = \alpha\ \mathbf{a} + \beta\ \mathbf{b}$, with $\alpha, \beta \geq 0$, and $\alpha + \beta > 0$. We can always assume that $\mathbf{b}$ is situated in the counter-clockwise direction from $\mathbf{a}$.

Definition 1. *A* polygonal differential inclusion system *(SPDI) is defined by giving a finite partition $\mathbb{P}$ of the plane into convex polygonal sets (called* regions*), and associating with each $P \in \mathbb{P}$ a couple of vectors $\mathbf{a}_P$ and $\mathbf{b}_P$. Let $\phi(P) = \angle_{\mathbf{a}_P}^{\mathbf{b}_P}$, we have that for each $\mathbf{x} \in P$, $\dot{\mathbf{x}} \in \phi(P)$.*

Let $E(P)$ be the set of edges of P. We say that $e \in E(P)$ is an *entry* of P if for all $\mathbf{x} \in e$ and for all $\mathbf{c} \in \phi(P)$, $\mathbf{x} + \mathbf{c}\epsilon \in P$ for some $\epsilon > 0$. We say that e is an *exit* of P if the same condition holds for some $\epsilon < 0$. We denote by $In(P) \subseteq E(P)$ the set of all entries of P and by $Out(P) \subseteq E(P)$ the set of all exits of P.

Assumption 1. *All the edges in $E(P)$ are either entries or exits, that is, $E(P) = In(P) \cup Out(P)$. We say then that all the regions in an SPDI are good or that they have the goodness property.*

Example 1. In Fig. 1-(a), region P (with $\phi(P) = \angle_{\mathbf{a}}^{\mathbf{b}}$) is good, since all are entry or exit edges. Fig. 1-(b) shows a region that is not good: edges e_2 and e_5 are not in $In(P) \cup Out(P)$. ■

A *trajectory segment* of an SPDI is a continuous function $\xi : [0, T] \to \mathbb{R}^2$ which is smooth everywhere except in a discrete set of points, and such that for all $t \in [0, T]$, if $\xi(t) \in P$ and $\dot{\xi}(t)$ is defined then $\dot{\xi}(t) \in \phi(P)$. The *signature*, denoted $\mathsf{Sig}(\xi)$, is the ordered sequence of edges traversed by the trajectory segment, that is, $e_1, e_2, \ldots$, where $\xi(t_i) \in e_i$ and $t_i < t_{i+1}$. If $T = \infty$, a trajectory segment is called a *trajectory*.
The following is a very simple example of an SPDI: a swimmer trying to escape from a whirlpool in a river.

Example. The dynamics $\dot{\mathbf{x}}$ of the swimmer around the whirlpool is approximated by the piece-wise differential inclusion defined as follows. The zone of the river nearby the whirlpool is divided into 8 regions $R_1, \ldots, R_8$. To each region R_i we associate a pair of vectors $(\mathbf{a}_i, \mathbf{b}_i)$ meaning that $\dot{\mathbf{x}}$ belongs to their positive hull: $\mathbf{a}_1 = \mathbf{b}_1 = (1, 5)$, $\mathbf{a}_2 = \mathbf{b}_2 = (-1, \frac{1}{2})$, $\mathbf{a}_3 = (-1, \frac{11}{60})$ and $\mathbf{b}_3 = (-1, -\frac{1}{4})$, $\mathbf{a}_4 = \mathbf{b}_4 = (-1, -1)$, $\mathbf{a}_5 = \mathbf{b}_5 = (0, -1)$, $\mathbf{a}_6 = \mathbf{b}_6 = (1, -1)$, $\mathbf{a}_7 = \mathbf{b}_7 = (1, 0)$, $\mathbf{a}_8 = \mathbf{b}_8 = (1, 1)$. The corresponding SPDI is illustrated in Fig. 2. □

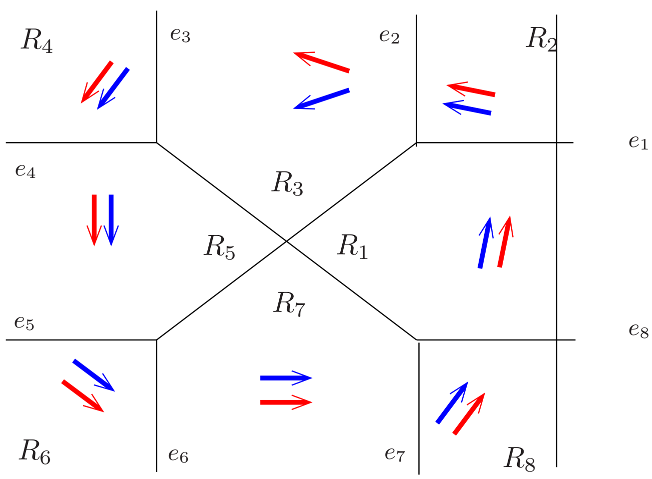

Figure 2: The SPDI of the swimmer.

3.1.1 Successors and predecessors

Given an SPDI, we fix a one-dimensional coordinate system on each edge to represent points laying on edges. For notational convenience, we will use e to denote both the edge and its one-dimensional representation. Accordingly, we write $\mathbf{x} \in e$ or $x \in e$, to mean "point $\mathbf{x}$ in edge e with coordinate x in the one-dimensional coordinate system of e". The same convention is applied to sets of points of e represented as intervals (e.g., $\mathbf{x} \in I$ or $x \in I$, where $I \subseteq e$) and to trajectories (e.g., "ξ starting in x" or "ξ starting in $\mathbf{x}$").

Now, let $P \in \mathbb{P}$, $e \in In(P)$ and $e' \in Out(P)$. For $I \subseteq e$, $\mathsf{Succ}_{ee'}(I)$ is the set of all points in e' reachable from some point in I by a trajectory segment $\xi : [0,t] \to \mathbb{R}^2$ in P (i.e., $\xi(0) \in I \wedge \xi(t) \in e' \wedge \mathsf{Sig}(\xi) = ee'$). Given $I = [l,u]$, $\mathsf{Succ}_{ee'}(I) = F(I \cap S) \cap J$, where S and J are intervals, $F([l,u]) = \langle f_l(l), f_u(u) \rangle$ and f_l and f_u are affine functions (a function $f : \mathbb{R} \to \mathbb{R}$ is affine iff $f(x) = ax + b$ with $a > 0$).

For $I \subseteq e'$, $\mathsf{Pre}_{ee'}(I)$ is the set of points in e that can reach a point in I by a trajectory segment in P. We have that: $\mathsf{Pre}_{ee'} = \mathsf{Succ}_{ee'}^{-1}$ and $\mathsf{Pre}_{\sigma} = \mathsf{Succ}_{\sigma}^{-1}$.

3.1.2 Qualitative analysis of simple edge-cycles

Let $\sigma = e_1 \cdots e_k e_1$ be a simple edge-cycle, i.e., $e_i \neq e_j$ for all $1 \leq i \neq j \leq k$. Let $\mathsf{Succ}_{\sigma}(I) = F(I \cap S) \cap J$ with $F = \langle f_l, f_u \rangle$.

Assumption 2. *None of the two functions f_l, f_u is the identity.*

Let l^* and u^* be the fix-points[1] of f_l and f_u, respectively, and $S \cap J = \langle L, U \rangle$. It can be shown that a simple cycle is of one of the following types:

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

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, $l^* < L \leq u^* \leq U$.

[1] The fix-point x^* is computed by solving a linear equation $f(x^*) = x^*$, which can be finite or infinite.

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 behavior of trajectories. Any trajectory that enters a cycle of type DIE will eventually quit it after a finite number of turns. If the 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.

To finish this section we recall the representation theorem for SPDIs that allows to factorize the signatures (step 3 in Section 2.1) in a convenient way. Given a sequence w, ε denotes the empty sequence whereas $first(w)$ and $last(w)$ are the first and last elements of the sequence respectively. An edge signature σ can be expressed as a sequence of edges and cycles of the form $r_1 s_1^{k_1} r_2 s_2^{k_2} \ldots r_n s_n^{k_n} r_{n+1}$, where

1. For all $1 \leq i \leq n+1$, r_i is a sequence of pairwise different edges;
2. For all $1 \leq i \leq n$, s_i is a simple cycle (i.e., without repetition of edges) repeated k_i times;

This is summarized by the following *representation theorem for SPDIs* that not only guarantees the existence of the above representation for SPDIs but also provides a constructive way of doing so [Sch02, Theorem 17].

Theorem 1. *Given an SPDI, 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 $1 \leq i \leq n+1$, r_i is a sequence of pairwise different edges and for all $1 \leq i \leq n$, s_i is a simple cycle (i.e., without repetition of edges).* □

This representation of signatures is the base to obtain *types of signatures* (step 4 in Section 2.1) with the following good properties [Sch02, Lemma 20].

Lemma 2. *Given an SPDI, let $\sigma = e_0 \ldots e_p$ be a feasible signature, then its type, $\texttt{type}(\sigma) = r_1, s_1, \ldots, r_n, s_n, r_{n+1}$ satisfies the following properties.*

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

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

The above is the base for the argument on the finiteness of different types of signatures to take into account in the reachability algorithm and thus to termination of SPDI reachability.

4 GSPDI

The *goodness* restriction (Assumption 1) was originally introduced to simplify treatment of trajectories to guarantee, among other things, that each region can be partitioned into *entry* and *exit* edges in an ordered way, fact used in the proof of decidability of the reachability problem. We will study in this section what happens when goodness is relaxed. First notice that without goodness there are edges that are neither of entry nor of exit as shown in Fig. 1. This naturally leads to the following definition.

Definition 2. *An edge $e \in P$ is an* inout *edge of P if e is neither an entry nor an exit edge of P.* ■

As already explained in previous sections, the above definition is the base for obtaining a new class of polygonal hybrid systems which generalizes SPDI.

Definition 3. *An SPDI without the goodness restriction is called a* general SPDI (GSPDI). ■

Thus, in GSPDIs there are three kinds of edges: inouts, entries and exits.
Self-crossing of trajectory segments of SPDIs can be eliminated which allow us to consider only non-crossing trajectory (segments). The proof given in [Sch02, Chap. 4, Sec. 4.2.2] can be extended to deal with the case when the self-crossing trajectories involve inout edges, so the result still holds for GSPDIs. Thus in what follows we will consider only trajectory segments without self-crossings.
Notice that on GSPDIs a trajectory can "intersect" an edge at an infinite number of points because it can *slide* at it. Thus, a trace is not anymore a sequence of points but rather a sequence of intervals.

Definition 4. *The* trace *of a trajectory ξ is the sequence* $\texttt{trace}(\xi) = I_0 I_1 \ldots$ *of the intersection intervals of ξ with the set of edges, that is, $I_i \subseteq (\xi \cap \mathcal{E})$.* ■

A point interval $I = [\mathbf{x}, \mathbf{x}]$ will be sometimes written as $\mathbf{x}$ whenever no confusion might arise.

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,* $\mathsf{Sig}(\xi) = e_0 e_1 \ldots$, *with* $\texttt{trace}(\xi) = I_0 I_1 \ldots$ *and $I_i \subseteq e_i$. The* region signature *of ξ is the sequence* $\mathsf{RSig}(\xi) = P_0 P_1 \ldots$ *of traversed regions, that is, $e_i \in In(P_i)$.* ■

Notice that in many cases the intervals of a trace are in fact points. We say that a trajectory with edge signature $\mathsf{Sig}(\xi) = e_0 e_1 \ldots e_i \ldots$ and trace $\texttt{trace}(\xi) = I_0 I_1 \ldots I_i \ldots$ *interval-crosses* edge e_i if I_i is not a point.

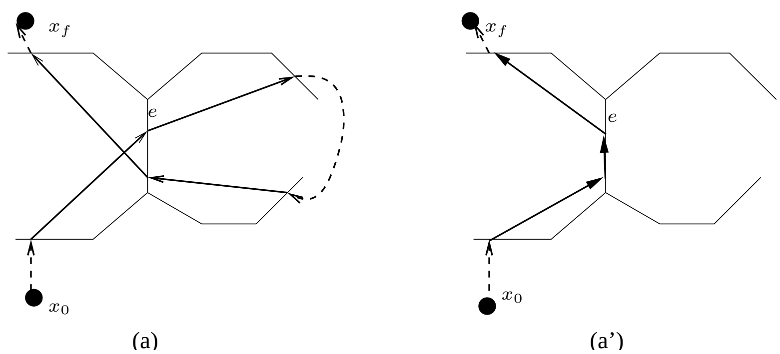

Figure 3: (a): A proper inout edge; (b): A sliding edge.

Given a trajectory segment, we will make the difference 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 $\mathsf{Sig}(\xi) = e_0 \ldots e_i \ldots e_n$, 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* on an edge e if e is a sliding edge of P for ξ and ξ is said to be a *sliding trajectory* if there is at least one sliding edge $e \in \mathsf{Sig}(\xi)$.

Example 2. In Fig. 3-(a), e is a proper inout edge. Edge e on Fig. 3-(b) is a sliding edge. ■

5 Simplification of GSPDI's Trajectory Segments

In this section we show that in many cases it is possible to simplify trajectory segments eliminating inout edges, but not always. We first start by showing that the good properties of the representation theorem for SPDIs are not valid any longer for GSPDIs, explaining why inouts edges are not desirable in a reachability analysis.

Proposition 1. *Property $\mathbf{P_2}$ of the representation theorem for SPDIs (Lemma 2) does not hold in general for GSPDIs.*

Proof: Let ξ be a trajectory with signature $\mathsf{Sig}(\xi) = \sigma = e_0 \ldots e_i \ldots e_n \ldots$ of a given GSPDI. The proposition states that it is not possible in general to write σ in the form $\sigma_{\mathcal{A}} = r_1 s_1^{k_1} \ldots r_n s_n^{k_n} r_{n+1}$ with the properties stated in Lemma 2. The proof is done by providing a counter-example. A typical counter-example should allow to obtain a signature consisting of a clockwise spiral followed by a counter-clockwise spiral (or vice-versa) and then back to the first spiral. In such a case it is possible to find two simple cycles which are repeated in the type of signature. Let us consider the GSPDI of Fig. 4. To let it simple we do not write down the dynamics of the regions and we assume that they are as to allow the segments of trajectories shown in the picture to be well-defined. In such a GSPDI it is possible to obtain the following type of signature: $r_1 s_1 r_2 s_2 r_3 s_3 \ldots$, where $s_1 = (abcd)$, $s_2 = (dcba)$, and $s_3 = (abcd)$. Since $s_1 = s_3$, then property $\mathbf{P_2}$ of Lemma 2 is not satisfied. $\square$

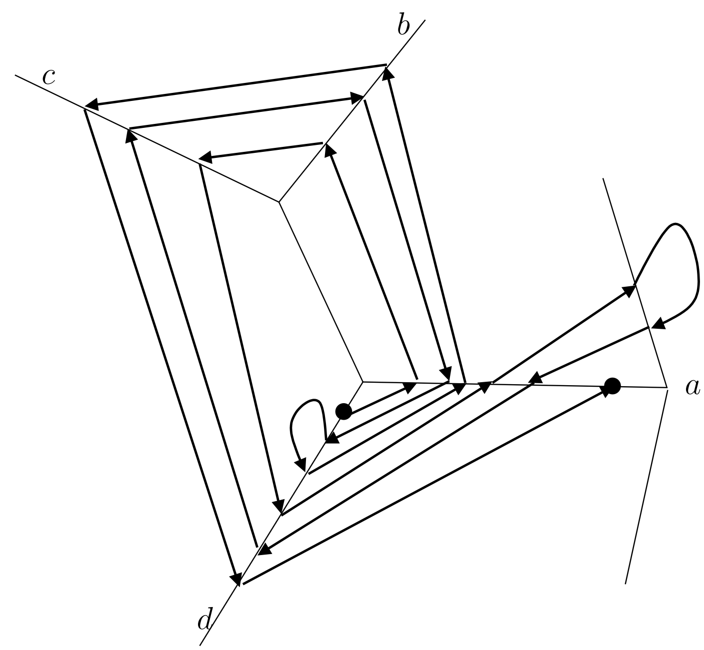

Figure 4: Counter-example for Proposition 1.

The following lemma presents some typical cases where it is possible to eliminate proper inout edges.

Lemma 3. *Let ξ be a trajectory segment $\mathbf{x}_0 \in e_0$ to $\mathbf{x}_f \in e_f$ with edge signature $\mathsf{Sig}(\xi) = e_0 \ldots e_i \ldots e_n$. If e_i is a proper inout edge then in some cases there exists a trajectory segment ξ' from $\mathbf{x}_0$ to $\mathbf{x}_f$ that traverses e_i in at most one sense (that is, e_i is either an entry or an exit, but no both).*

Proof Sketch: In Fig. 5-(a) we illustrate a typical case where edge e_i is a proper inout edge. After a straightforward algebraic vector manipulation, on the same lines of elimination of self-crossings, the trajectory segment shown in Fig. 5-(a') is obtained. □

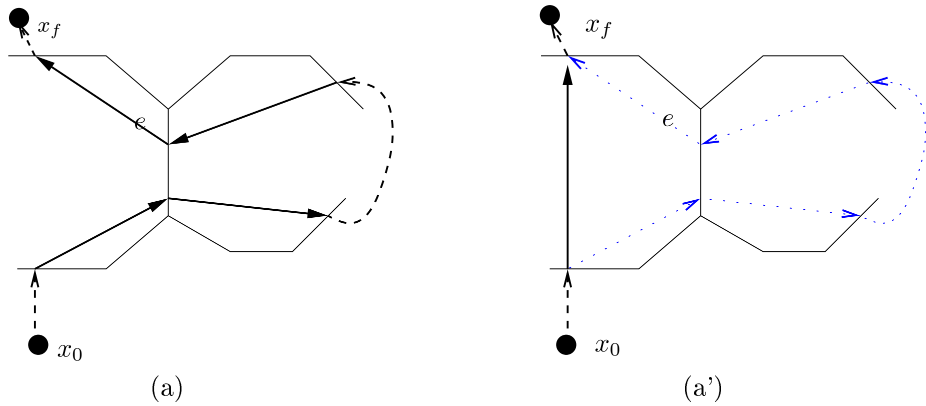

Figure 5: Inout case.

Note that the above does not establish completeness of a reduction from GSPDIs into SPDIs reachability since there are cases where the above is not possible. We have then the following result.

Proposition 2. *Given a GSPDI, assume there exists a trajectory segment from points* $\mathbf{x}_0 \in e_0$ *to* $\mathbf{x}_f \in e_f$*, traversing inout edges in both directions. Then it is, in general, not possible to find a trajectory segment whose edge signature contains no proper inout edges (traversed in both directions), between them.*

Proof: The GSPDI of Fig. 6 presents a typical example of an inout edge (e_2) which cannot be directly eliminated as to preserve that x_f is reachable from x_0. To keep the explanation simple we do not present here a formal GSPDI as counter-example. The example, however, sheds some light on the kind of GSPDI regions serving as counter-examples. It suffices to take any trajectory with a dynamics such that the angle is slightly less than 180 degrees. The trajectory must traverse an inout edge following the **b** vector and enters into the region by following the **a** vector. The trajectory must not cross itself. □

We show now how to eliminate sliding edges.

Lemma 4. *Let* ξ *be a trajectory segment* $\mathbf{x}_0 \in e_0$ *to* $\mathbf{x}_f \in e_f$ *with edge signature* $\mathsf{Sig}(\xi) = e_0 \dots e_i \dots e_n$*. If* e_i *is a sliding edge for* ξ *then there exists a trajectory segment* ξ' *from* $\mathbf{x}_0$ *to* $\mathbf{x}_f$ *that does not slide on edge* e_i*.*

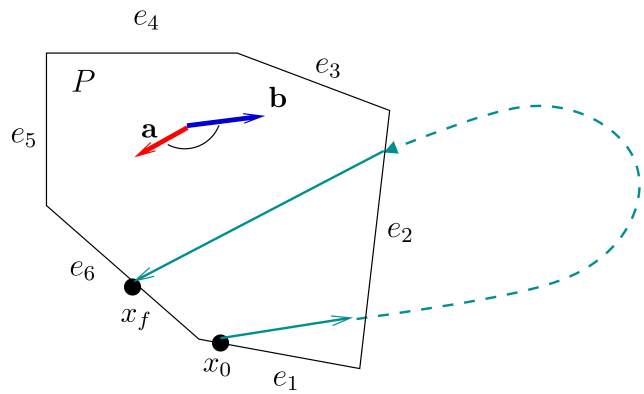

Figure 6: A GSPDI with a non-eliminating inout edge.

Proof Sketch: Sliding edges can arise in four different cases (without taking into account the symmetric cases); they are shown in Fig. 14-(a) to (d). The corresponding primed figures (Fig. 14-(a') to (d')) show the transformation done in order to avoid sliding on edge e. The reason why the above transformation is possible is because in all the cases the new obtained segment of trajectory can be expressed as a positive linear combination of two suitable existing segments of trajectory. Such two segments are the sliding segment, and another segment of trajectory with starting point at the beginning or the end of the sliding segment. □

As a consequence we have the following result.

Proposition 3 (Existence of a non-sliding trajectory)**.** *If there exists a sliding trajectory segment from points $\mathbf{x}_0 \in e_0$ to $\mathbf{x}_f \in e_f$ then there always exists a non-sliding trajectory segment between them.*

Proof: By induction on the number n of sliding edges of the signature of the trajectory segment using Lemma 4 in the induction step. □

We usually eliminate first proper inout edges (when possible) and next sliding. In fact, the number of sliding edges is not guaranteed to decrease if sliding edges are eliminated before proper inout edges as shown in the following example.

Example 3. In Fig. 7-(a) a trajectory segment that slides at edge e' is shown. After eliminating the sliding at edge e', a new sliding edge is introduced (e). This is shown in Fig. 7-(b). However, if proper inout edges are eliminated first, we do not introduce new proper inout edges as shown in part (c) of the same figure. ∎

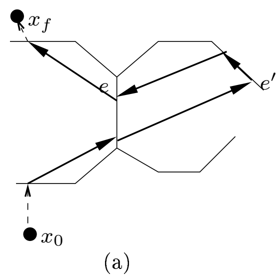

Figure 7: Elimination order of inout edges.

Remark. Sliding is not easy to treat in general since an edge always belong to two different regions with different dynamics. Thus a trajectory may be 'allowed' to slide by one of the dynamics but not by the other. We do not analyze this in more detail, for our purposes we assume that at an inout edge a trajectory can slide if at least one of the dynamics allows so. This assumption does not affect the reachability analysis.

About the ordering between edges. We finish this section with an informal discussion about the importance of the 'contiguous' order between entry and exit edges on SPDIs.
In SPDIs edges of a region can be bi-partitioned into entry and exit edges in a contiguous way (see Fig. 8) having as a consequence an ordering between edges. This is not longer the case in GSPDIs.

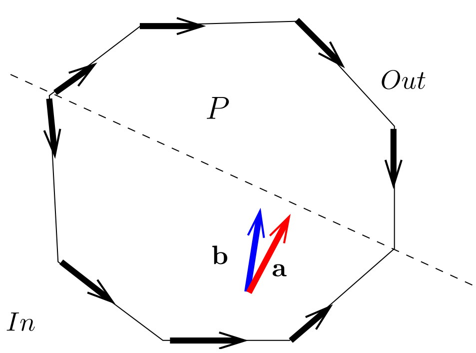

Figure 8: Ordering of edges on an SPDI (all the edges e satisfy $\hat{\mathbf{a}}\,\mathbf{e} > 0$).

First of all, notice that the ordering of edges on an SPDI were chosen in order to preserve the 'positive affinity' (and hence the monotonicity) of the successor functions. Given a region R with differential inclusion $\angle_{\mathbf{a}}^{\mathbf{b}}$, let e be an entry edge and e_1 and e_2 two exit edges of R. For e we chose the direction (given by a director vector $\mathbf{e}$) that satisfies the inequality $\hat{\mathbf{a}}\ \mathbf{e} > 0$ (see Fig. 11). The same for e_1 and e_2. As a consequence we obtain an ordering like the one shown in Fig. 8.
Note that on a GSPDI (see Fig. 9(a)), the property that for any edge e, $\hat{\mathbf{a}}\ \mathbf{e} > 0$ is not longer valid since an edge can be of entry and of exit and then the ordering can change. In spite of that, once an inout edge is 'converted' into an entry (or exit) then we can have the notation of considering the ordering of entry edges going counter-clockwise and clockwise for exit edges (see Fig. 9(b)).

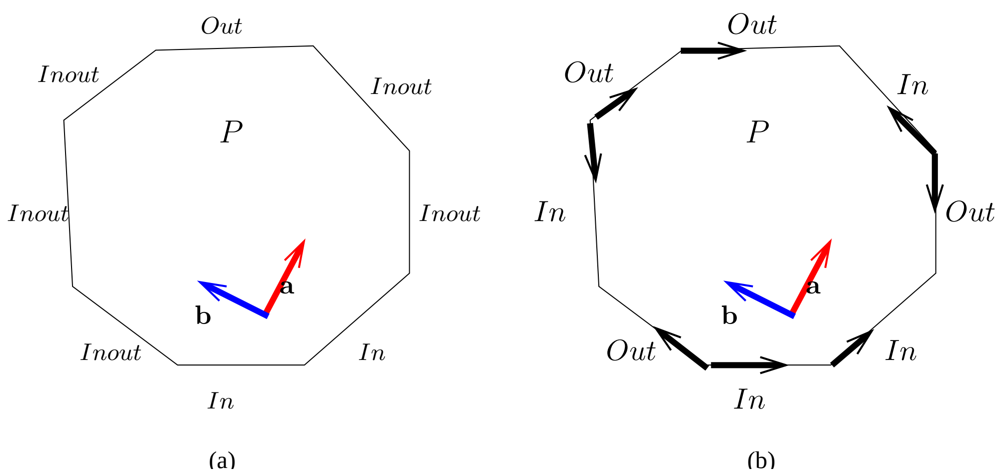

Figure 9: (a) A GSPDI; (b) Ordering after fixing input and output edges.

Even though the definition of edge and region signatures as well as edge cycle continue to hold, it is not the case for region cycle. We can have a region signature $P_1 \cdots P_i \cdots P_k P_1$ that is not a region cycle. The reason is that in GSPDIs a trajectory can enter a region through two different edges without forming a cycle.
Thus we have that a region signature $P_1 \cdots P_i \cdots P_k P_1$ is a *region cycle* if the edge signature $e_1 \cdots e_k e_1$, with $e_i \in Out(P_i)$ for all $1 \leq i \leq k$, forms an edge cycle.
In Fig. 10 the following is a region cycle: $P_1P_2P_3P_4P_2P_5P_1$. Notice that $P_2P_3P_4P_2$ is region cycle for SPDIs but not for the given GSPDI.

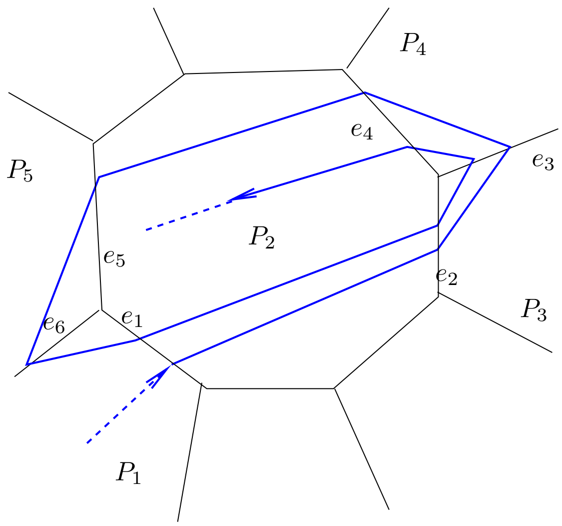

Figure 10: A region cycle.

6 Reachability Analysis for GSPDIs

In this section we 'topologically' rephrase and prove the results of [Sch02, Chap. 4,5] that use the contiguity between entry and exit edges in their proofs. We also re-prove soundness of Exit-LEFT and Exit-STAY algorithms and at the end we give a semi-decision algorithm for GPSDI reachability. We have informally explained in Section 2.2 why we need to do so.

6.1 Proof of Lemmas without using the Contiguity Assumption

The only results that use the contiguity order between entry and exit edges are Lemmas 20, Lemma 26 and Corollary 27 of [Sch02]. Lemma 20 has been repeated here in Section 3 as Lemma 2, which as we have seen does not hold in general for GSPDIs (Proposition 1). However, after fixing all the edges as either of entry or exit, we can prove the result holds since it behaves as an SPDI, modulo the contiguity of entry and exit edges.

We prove then these three results without using the order between entry and exit edges. We restate Lemma 2 ([Sch02, Lemma 20]) for property $\mathbf{P_2}$, for the case when GPSDI is transformed as to fix inout edges as entries or exits.

Lemma 5. *Given a GSPDI where edges has been fixed as entry or exit, let $\sigma = e_0 \dots e_p$ be a feasible signature, then its type,* $\texttt{type}(\sigma) = r_1, s_1, \dots, r_n,$

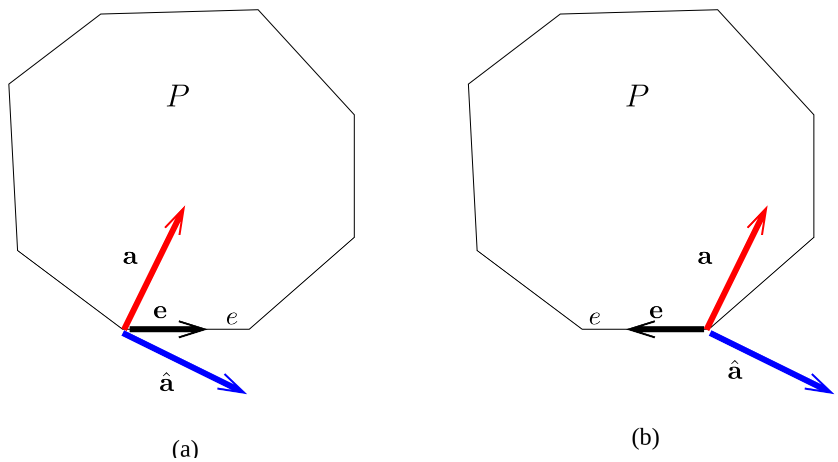

Figure 11: (a) $\hat{\mathbf{a}}\,\mathbf{e} > 0$; (b) $\hat{\mathbf{a}}\,\mathbf{e} < 0$.

s_n, r_{n+1} *satisfies the following property,* $\mathbf{P_2}$*: For every* $1 \leq i \neq j \leq n$*,* s_i *and* s_j *are different.*

Proof: In order to prove property $\mathbf{P_2}$ we prove that, given a simple cycle $s_i = e', \ldots, e$, the sequence of edges ee' cannot occur after leaving s_i (hence it cannot occur in any other simple cycle s_j, with $1 \leq i < j \leq n$). After cycling k_i times cycle s_i is abandoned by edge e (guaranteed by construction). Let P be a region s.t. $e \in In(P)$ and consider the unfolding of the last iteration and its continuation (see Fig. 12-(a)):

$$\ldots, e, e', \ldots, e, e'', \ldots$$

where $e'' = first(r_{i+1})$, $e \in In(P)$ and $e', e'' \in Out(P)$ ($e' \neq e''$). Let x_2 be the last point visited on edge e before leaving cycle s_i and x_2'' be the first point on edge e'' after leaving s_i (see Fig. 12-(b)). Segment $\overline{\mathbf{x_2 x_2''}}$ of the trajectory segment divides region P into two subregions P_1 and P_2 and edge e into two segments $\overline{e^l x_2}$ and $\overline{x_2 e^u}$. By the non-crossing hypothesis (and monotonicity on edges) after leaving s_i the only accessible part of edge e is the segment $\overline{x_2 e^u} \in e$. By Jordan's curve theorem the only way to reach edge e' from any point in $\overline{x_2 e^u} \in e$ is by crossing $\overline{\mathbf{x_2 x_2''}}$ or by crossing one of the edges of region P_2. The first case is not possible since it would contradict the hypothesis of non-crossing trajectory and in the second case the sequence ee' would not belong to the trajectory segment. $\square$

Remark. Note that for our purposes it is irrelevant whether property $\mathbf{P_1}$ holds or not, since it does not affect the finiteness argument. This is due to

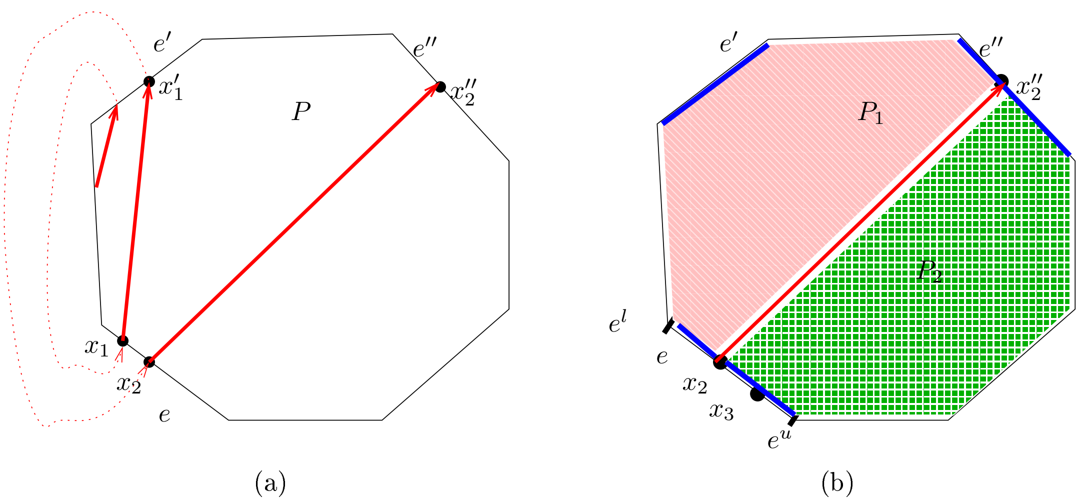

Figure 12: (a): Simple cycle s_i and its continuation through edge e; (b) Edge e' cannot be reached from point x_3 without intersecting $\overline{\mathbf{x_2x''_2}}$

the fact that a type of signature is finite if the number of simple cycles are not repeated, which is stated in $\mathbf{P_2}$.

In what follows we use the following notation. Whenever we partition the space into two regions P_L and P_R by the line defined by a segment of line $\overline{xy}$, P_L is the semi-space of all the points that are a left rotation of $\vec{xy}$ and P_R is the semi-space corresponding to the points that are a right rotation of the same vector. With $f(x)\downarrow$ we mean that f is defined at x and $f(x)\uparrow$ will mean that f is undefined at x.
Next we will (topologically) rephrase [Sch02, Lemma 26] and [Sch02, Corollary 27] and we prove them both.

Lemma 6. *Let P be a region, $e \in In(P)$, $e_1, e_2 \in Out(P)$, $\langle l_i, u_i\rangle$ be any subinterval of $\langle e_i^l, e_i^u\rangle$ and $f_i(x) = F^{\mathbf{c}}_{e,e_i}(x)$.*

1. *Let P be partitioned into two regions P_L and P_R by the line defined by $\overline{xl_1}$, then the following holds: if $e_2 \in P_L$, $f_2(x)\downarrow$ and $l_1 < f_1(x)$ then $u_2 < f_2(x)$;*

2. *Let the plane be partitioned into two subspaces P_L and P_R by the line defined by $\overline{xl_2}$, then the following holds: if $e_1 \in P_R$, $f_1(x)\downarrow$ and $f_2(x) < u_2$ then $f_1(x) < l_1$.*

Proof:

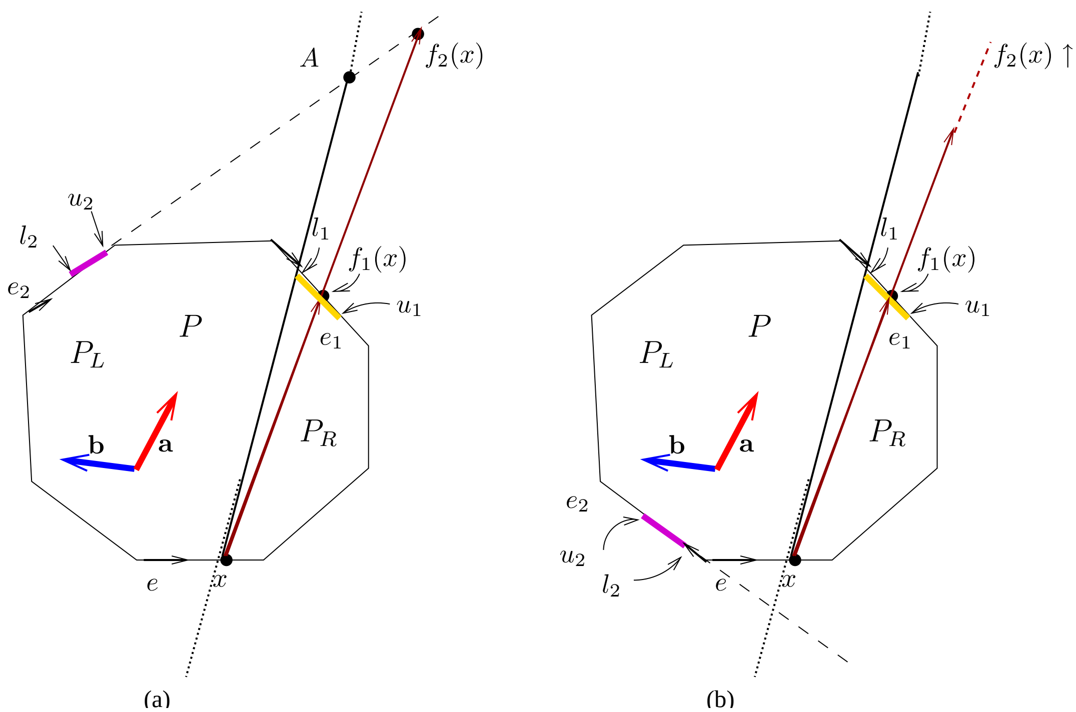

Figure 13: Lemma 6'-1. (a) When $f_2^l(x)\downarrow$; (b) The case $f_2^l(x)\uparrow$.

1. Remember that the line defined by e_2 is ordered and that u_2, A and $f_2(x)$ belongs to it. We have then that $e_2 \in P_L$ (and hence $u_2 \in P_L$) and that $f_2(x) \in P_R$ (by construction of the partition). We have then that $u_2 < A$ and $A < f_2(x)$, that implies $u_2 < f_2(x)$. See Fig. 13(a).

2. This case is symmetric to the previous one. □

Corollary 7. *Let P be a region, $e \in In(P)$, $e_1, e_2 \in Out(P)$, $f_i(x) = F^{\mathbf{c}}_{e,e_i}(x)$ be an affine function and $\mathcal{F}_i(\langle x, y\rangle) = F_i(\langle x, y\rangle \cap S_i) \cap J_i$ be a truncated affine multi-valued function (with $F_i = [f_i^l, f_i^u]$ and $J_i = \langle L_i, U_i\rangle$).*

1. *Let P be partitioned into two regions P_L and P_R by the line defined by $\overline{xL_1}$, then the following holds: If $e_2 \in P_L$ and $L_1 < f_1^l(x)$ then $\mathcal{F}_2(\langle x, y\rangle) = \emptyset$;*

2. *Let P be partitioned into two regions P_L and P_R by the line defined by $\overline{xL_2}$, then the following holds: if $e_1 \in P_R$ and $f_2^u(y) < U_2$ then $\mathcal{F}_1(\langle x, y\rangle) = \emptyset$.*

Proof:

1. If $f_2^l(x)$ is undefined, then it is obvious that $\mathcal{F}_2(\langle x, y\rangle) = \emptyset$. If $f_1^l(x)$ is defined, then the result follows directly from Lemma 6-1 and definition of $\mathcal{F}_i(\langle x, y\rangle)$.
2. Symmetric to the above case using Lemma 6-2. $\square$

6.2 Soundness of Exit-STAY and Exit-LEFT

We prove now soundness of the Exit-STAY and Exit-LEFT algorithm whose proofs rely on the results proved in the previous section.
Let $A = \mathsf{Succ}_s^{\mathbf{b}}(L)$ and consider the line defined by $\overline{AL}$. This line partition the space into P_L and P_R as before.

Exit-STAY

function $Exit_{STAY}(I, s, ex)$ $\longleftarrow \emptyset$

Soundness By hypothesis, $L < l^* < u^* < U$. Hence, for all i, $\tilde{I}_i = \langle \tilde{l}_i, \tilde{u}_i\rangle \subseteq \langle L, U\rangle$, hence $I_i = \tilde{I}_i$ and by Corollary 7 we have that $\mathsf{Succ}_{s,ex}^i(I) = \emptyset$.

Termination Trivial. $\square$

Exit-LEFT:

function $Exit_{LEFT}(I, s, ex)$ $\longleftarrow \mathsf{Succ}_{s,ex}(\mathsf{Succ}_{s,f}(\langle L, \max\{u, u^*\}\rangle))$

Soundness By hypothesis, $l^* < L < u^* \leq U$. Thus, there exists a natural number n s.t. $\tilde{l_n} \leq L$ and for all i, $u_i = \tilde{u}_i \leq U$. Let's consider the following two cases:

1. If $ex \in P_R$ then $Ex = \emptyset$ (by definition of Exit-LEFT) and $\mathsf{Succ}_{s,ex}(I_i) = \emptyset$ for any i (by Corollary 7-2), so $\mathsf{Succ}_{s,ex}(\mathsf{Succ}_{s,f}(\langle L, \max\{u, u^*\}\rangle)) = \emptyset$;
2. If $ex \in P_L$, we consider two cases:
   (a) If $u < u^*$ then for all i, $u_i = \tilde{u}_i \leq u^*$ and then $\cup_{m>0}\mathsf{Succ}_{s,f}^m(I) = \mathsf{Succ}_{s,f}(L, u^*)$, thus $Ex = \mathsf{Succ}_{s,ex}(\mathsf{Succ}_{s,f}(L, u^*))$;

(b) If $u^* < u$ then for all i, $u_i = \tilde{u}_i \leq u$ and $\cup_{m>0}\mathsf{Succ}^m_{s,f}(I) = \mathsf{Succ}_{s,f}(L, u)$. Consequently, $Ex = \mathsf{Succ}_{s,ex}(\mathsf{Succ}_{s,f}(L, u))$;

From both cases we have that $Ex = \mathsf{Succ}_{s,ex}(\mathsf{Succ}_{s,f}(\langle L, \max\{u, u^*\}\rangle))$.

Termination Trivial. □

6.3 A semi-decision algorithm for reachability analysis of GSPDIs

From the above results we have that the main algorithm for reachability may be applied to GSPDIs after performing certain pre-processing steps.
Before presenting a sound (but incomplete) algorithm for reachability analysis of GSPDIs we need the following notation. Given a GSPDI $\mathcal{H}$, we denote by $\mathcal{H}_{red} = \{\mathcal{H}_1, \ldots, H_n\}$ the set of all the SPDIs obtained after fixing all the inout edges of $\mathcal{H}$ as inputs or outputs, considering all the possible permutations.
The reachability algorithm for a GSPDI $\mathcal{H}$, $\mathbf{Reach}(\mathcal{H}, \mathbf{x}_0, \mathbf{x}_f)$, consists of the following steps:

1. Detect all the inout edges;
2. Generate the set of SPDIs $\mathcal{H}_{red} = \{\mathcal{H}_1, \ldots, H_n\}$;
3. Apply the reachability algorithm for SPDIs to each $\mathcal{H}_i$ $(1 \leq i \leq n)$.
4. If there exists at least one $\mathcal{H}_i \in \mathcal{H}_{red}$ such that $\mathbf{Reach}(\mathcal{H}_i, \mathbf{x}_0, \mathbf{x}_f) = \mathsf{Yes}$ then $\mathbf{Reach}(\mathcal{H}, \mathbf{x}_0, \mathbf{x}_f) = \mathsf{Yes}$, otherwise we do not know.

We have then the following result about termination of GSPDI reachability.

Lemma 8. $\mathbf{Reach}(\mathcal{H}, \mathbf{x}_0, \mathbf{x}_f)$ *always terminate.*

Proof: The result follows from the termination of steps 1 and 2 of the above algorithm, as well as from that of $\mathbf{Reach}(\mathcal{H}_i, \mathbf{x}_0, \mathbf{x}_f)$ (for all $\mathcal{H}_i \in \mathcal{H}_{red}$, $1 \leq i \leq n$). □

We finish this section with the main result of our paper, which follows from all the previous results, stating that we can semi-decide reachability for GSPDIs.

Theorem 9. *Given a GSPDI $\mathcal{H}$, if* $\mathbf{Reach}(\mathcal{H}_i, \mathbf{x}_0, \mathbf{x}_f) = \mathsf{Yes}$ *for some $\mathcal{H}_i \in \mathcal{H}_{red}$, then* $\mathbf{Reach}(\mathcal{H}, \mathbf{x}_0, \mathbf{x}_f) = \mathsf{Yes}$. *On the other hand, if for all $\mathcal{H}_i \in \mathcal{H}_{red}$,* $\mathbf{Reach}(\mathcal{H}_i, \mathbf{x}_0, \mathbf{x}_f) = \mathsf{No}$*, then* $\mathbf{Reach}(\mathcal{H}, \mathbf{x}_0, \mathbf{x}_f)$ *is inconclusive.*

Proof: Termination is guaranteed by Lemma 8. Soundness follows from soundness of the algorithm for SPDIs [Sch02, Sec. 5.2], including the new proof given in Section 6.2 considering the use of non-contiguous entry and exit edges. The fact that reachability is inconclusive whenever $\mathbf{Reach}(\mathcal{H}_i, \mathbf{x}_0, \mathbf{x}_f) = \mathsf{No}$ for all $\mathcal{H}_i \in \mathcal{H}_{red}$ follows from Proposition 2. □

7 Final Discussion

In this work we have provided a counter-example to a previous proof of the decidability of the reachability problem for GSPDIs given in [Sch02, Chap. 9], which remain thus an open problem. We have rephrased the results given in above mentioned work in order to give a semi-decidable algorithm for solving the reachability problem for such class of systems.

References

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

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

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

[ASY07] Eugene Asarin, Gerardo Schneider, and Sergio Yovine. Algorithmic Analysis of Polygonal Hybrid Systems. Part I: Reachability. *Theoretical Computer Science*, 379(1-2):231–265, 2007. doi:10.1016/j.tcs.2007.03.055.

[HS74] Morris W. Hirsch and Stephen 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] Venkatesh Mysore and Amir Pnueli. Refining the undecidability frontier of hybrid automata. In *FSTTCS*, volume 3821 of *LNCS*, pages 261–272. Springer-Verlag, 2005.

[NS60] V.V. Nemytskii and V.V. Stepanov. *Qualitative theory of differential equations.* Princeton University Press, 1960.

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

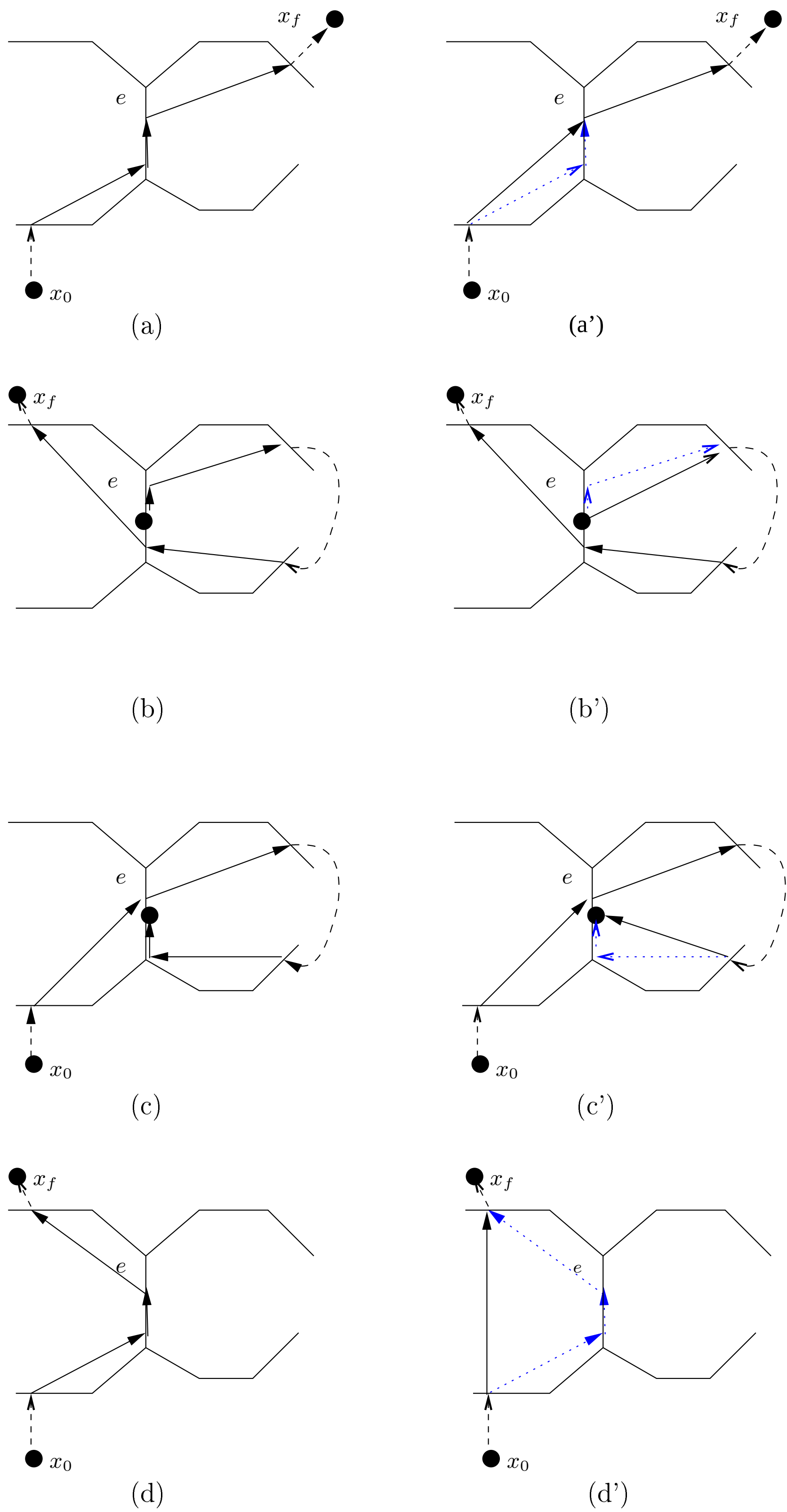

Figure 14: Sliding cases.