On the Decidability of the Reachability Problem for Planar Differential Inclusions

E. Asarin *† G. Schneider‡† S. Yovine †

Eugene.Asarin@imag.fr Gerardo.Schneider@imag.fr Sergio.Yovine@imag.fr

VERIMAG
Centre Equation
2, Ave de Vignate
38610 Gières, France
Phone: (33) (0) 476 63 48 43
Fax: (33) (0) 476 63 48 50

Abstract

In this paper we develop an algorithm for solving the reachability problem of two-dimensional piece-wise rectangular differential inclusions. Our procedure is not based on the computation of the reach-set but rather on the computation of the limit of individual trajectories. A key idea is the use of one-dimensional affine Poincaré maps for which we can easily compute the fixpoints. As a first step, we show that between any two points linked by an arbitrary trajectory there always exists a trajectory without self-crossings. Thus, solving the reachability problem requires considering only those. We prove that, indeed, there are only finitely many "qualitative types" of those trajectories. The last step consists in giving a decision procedure for each of them. These procedures are essentially based on the analysis of the limits of extreme trajectories. We illustrate our algorithm on a simple model of a swimmer spinning around a whirlpool.

1 Introduction

One of the main research areas in hybrid systems is reachability analysis. It comprises two (closely related) issues, namely, the study of decidability and the development of algorithms. Most of the proved decidability results are based on the existence of a finite and computable partition of the state space into classes of states which are equivalent with respect to reachability. This is the case for timed automata [2], and classes of rectangular automata [11] and hybrid automata with linear vector fields [14]. Except for timed automata, these results rely on stringent hypothesis such as the resetting of variables along transitions.

Although analysis techniques based on the construction of a finite partition have been proposed [6], mainly all implemented computational procedures resort to (forward or backward) propagation of constraints, typically (unions of convex) polyhedra or ellipsoids [1, 3, 5, 8, 10, 13]. In general, these techniques provide semi-decision procedures, that is, if the given final set of states is reachable, they will terminate, otherwise they may fail to. This is a property of the techniques, not of the problem, that is, it does not imply that the reachability problem itself is undecidable, but only that they do not implement a decision procedure for it. In other words, these algorithms may be unsuccessful (i.e., not terminate) for certain classes of systems for which the reachability problem is indeed decidable (by other means). Nevertheless, they provide tools for computing

*Partially supported by the NATO under grant CRG-961115.
†Partially supported by Projet IMAG MASH "Modélisation et Analyse de Systèmes Hybrides".
‡Supported by ESPRIT-LTR Project 26270 VHS "Verification of Hybrid Systems".

(approximations of) the reach-set for large classes of hybrid systems with linear and non-linear vector fields.

Maybe the major drawback of set-propagation, reach-set approximation procedures is that they pay little attention to the geometric properties of the specific (class of) systems under analysis. To our knowledge, in the context of hybrid systems there are two lines of work in the direction of developing more "geometric" approaches. One is based on the existence of (enough) integrals and the ability to compute them all [6, 9]. These methods, however, do not necessarily result in decision procedures (they are actually not meant to). The other, applicable to two-dimensional dynamical systems, relies on the topological properties of the plane, and explicitly focuses on decidability issues. This approach has been proposed in [15]. There, it is shown that the reachability problem for two-dimensional systems with piece-wise constant derivatives (PCD) is decidable. This result has been extended in [7] for planar piece-wise Hamiltonian systems. In [4] it has been shown that the reachability problem for PCD is undecidable for dimensions higher than two.

In this paper we develop an algorithm for solving the reachability problem of two-dimensional piece-wise rectangular differential inclusions. As in [15], our procedure is not based on the computation of the reach-set but rather on the computation of the limit of individual trajectories. A key idea is the use of one-dimensional affine Poincaré maps for which we can easily compute the fixpoints. The decidability result of [15] fundamentally relies on the determinism of PCD which implies that planar trajectories do not intersect themselves. This property is no longer true for differential inclusions. As a first step, we show that between any two points linked by an arbitrary trajectory there always exists a trajectory without self-crossings. Thus, solving the reachability problem requires considering only those. We prove that, indeed, there are only finitely many "qualitative types" of those trajectories. The last step consists in giving a decision procedure for each of them. These procedures are essentially based on the analysis of the limits of extreme trajectories (which do not cut themselves).

2 Preliminaries

Let $\mathbf{a} = (a_1, a_2), \mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ and $\lambda \in \mathbb{R}$. We define: $\mathbf{a}\,\mathbf{x} = a_1x_1 + a_2x_2$ (inner product), $\lambda\mathbf{x} = (\lambda x_1, \lambda x_2)$, $|\mathbf{x}| = \sqrt{\mathbf{x}\,\mathbf{x}}$ (2-norm). 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 $\mathbf{x}\,\hat{\mathbf{x}} = 0$.

The *distance* between two points $\mathbf{x}$ and $\mathbf{y}$ is defined to be $|\mathbf{x}-\mathbf{y}|$. For $\epsilon > 0$, the *ϵ-neighborhood* of $\mathbf{x}$ is $B_\epsilon(\mathbf{x}) = \{\mathbf{y} \mid |\mathbf{x}-\mathbf{y}| < \epsilon\}$. The *interior* of $X \subseteq \mathbb{R}^2$, denoted by $int(X)$, is the set of $\mathbf{x} \in X$ for which there exists $\epsilon > 0$ such that $B_\epsilon(\mathbf{x}) \subseteq X$.

For $\mathbf{x}_1, \ldots \mathbf{x}_n \in \mathbb{R}^2$ a *linear* combination is a vector $\mathbf{x} = \sum_{i=1}^{n} \lambda_i\mathbf{x}_i$ for some $\lambda_i \in \mathbb{R}$. A *positive combination* is a linear combination with $\lambda_i \geq 0$ for every i. The positive *hull* of a set $X \subseteq \mathbb{R}^2$ is the set of all positive combinations of points in X. A (closed) *half-space* is the set of all points $\mathbf{x}$ satisfying $\mathbf{a}\,\mathbf{x} \leq b$. A *convex polyhedral set* P is the intersection of finitely many half-spaces. An *edge* is a segment of line in $\mathbb{R}^2$.

For $A = \{\mathbf{a}, \mathbf{b}\}$, with $\hat{\mathbf{a}}\,\mathbf{b} < 0$, we denote by $\angle_{\mathbf{a}}^{\mathbf{b}}$ the positive hull of A (Fig. 1).

3 Simple planar differential inclusions

Let S be a finite index set and $\mathcal{P} = \{P_s\}_{s\in S}$ be a finite set of convex polyhedral sets, called *regions*, such that:

1. For all $s \in S$, $int(P_s) \neq \emptyset$;
2. For all $s \neq r \in S$, $int(P_s \cap P_r) = \emptyset$;
3. $\bigcup_{s\in S} P_s = \mathbb{R}^2$.

Condition 1 states that regions are full dimensional. Condition 2 says that the intersection between two regions is either empty, an edge, or a point, whereas the third condition states that the regions

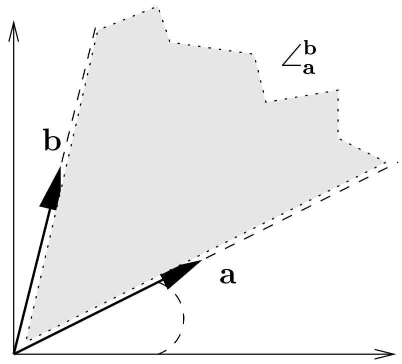


Figure 1: Positive hull of $\{\mathbf{a}, \mathbf{b}\}$ with $\hat{\mathbf{a}}\ \mathbf{b} < 0$.

covers the whole space. Thus, $\mathcal{P}$ is a partition of the plane. We denote by $E(P)$ the set of edges of the form $e = P \cap P'$ with $P \neq P'$ and by $V(P)$ the set of vertices of the form $v = e \cap e'$ with $e, e' \in E(P)$. Let $EV(P) = int(E(P)) \cup V(P)$.

Let $\mathcal{F} = \{\phi_s\}_{s \in S}$ be such that ϕ_s is the positive hull of two vectors $\mathbf{a}_s$ and $\mathbf{b}_s$ with $\hat{\mathbf{a}}_s\ \mathbf{b}_s < 0$.

A *simple planar differential inclusion system* (SPDI) is a pair $\mathcal{H} = (\mathcal{P}, \mathcal{F})$. Each region P_s has dynamics $\dot{\mathbf{x}} \in \phi_s$ for $\mathbf{x} \in P_s$ (given a generic region P we will also use the notation $\phi(P)$). A *trajectory* in some interval $[0, T] \subseteq \mathbb{R}$, with initial condition $\mathbf{x} = \mathbf{x}_0$, is a continuous and almost-everywhere (everywhere except on finitely many points) derivable function $\xi(\cdot)$ such that $\xi(0) = \mathbf{x}_0$ and for all $t \in (0, T)$, if $\xi(t) \in P \setminus EV(P)$, then $\dot{\xi}(t)$ is defined and $\dot{\xi}(t) \in \phi(P)$.

Consider a region P and an edge $e \in E(P)$. We say that e is an *entry* of P if for all $\mathbf{x} \in e \setminus V(P)$ 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.

Let $\mathbf{x} \in V(P)$ be a common vertex of two edges e and e'. $\mathbf{x}$ is an *entry point* to P if both e and e' are entry edges; it is an exit point if both e and e' are exit edges. In fact, vertices can be seen as a particular kind of edges, with exactly one point and in what follows we will consider them as edges, hence the term *edge* will be understood as belonging to the set EV. If needed, the difference between edges in V and E will be explicitely specified.

In general, $E(P) \neq in(P) \cup out(P)$. We say that P is a *good* region iff all the edges in $E(P)$ are entries or exits, that is, $E(P) = in(P) \cup out(P)$. Notice that, if P is a good region, then for all $e \in E(P)$, $e \notin \phi(P)$ (Fig. 2 with $\phi(P) = \angle_{\mathbf{a}}^{\mathbf{b}}$). Hereinafter, we assume that all regions are good regions.

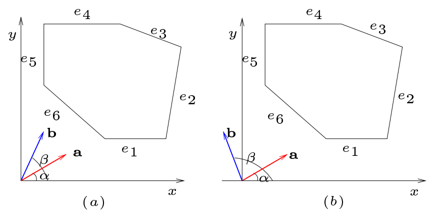


Figure 2: a) A good region. b) A bad region ($e_5 \notin in(P) \cup out(P)$).

The point-to-point reachability problem for $\mathcal{H}$, is the following: Given $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^2$, is there a trajectory ξ and $t \geq 0$ such that $\xi(0) = \mathbf{x}$ and $\xi(t) = \mathbf{x}'$?. If the answer is yes, we say that $\mathbf{x}'$ is *reachable* from $\mathbf{x}$. The edge-to-edge reachability problem is the following: Given two edges e and e' of $\mathcal{H}$, is there $\mathbf{x} \in e$ and $\mathbf{x}' \in e'$ such that $\mathbf{x}'$ is reachable from $\mathbf{x}$? The region-to-region reachability problem is defined straightforwardly.

Example. The dynamics $\dot{\mathbf{x}}$ of the swimmer around the whirlpool is approximated by the piecewise differential inclusion defined as follows. The zone of the river nearby the whirlpool is divided into 8 regions $R_1, \dots, R_8$. 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. 3. □

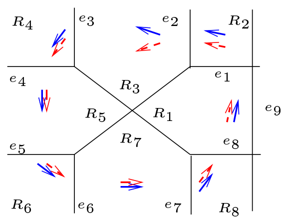


Figure 3: The SPDI of the swimmer.

4 Properties of trajectories

4.1 Rectilinear trajectories

W.l.o.g. we will consider in what follows that $\xi(0) \in e$ for some edge $e \in EV$. The *trace* of a trajectory ξ is the sequence $\tau(\xi) = \mathbf{x}_0\mathbf{x}_1 \dots$ of the intersection points of ξ with the set of edges, that is, $\mathbf{x}_i \in \xi \cap \bigcup_{s \in S} EV(P_s)$. The *edge signature* of ξ is the sequence $\sigma(\xi) = e_0 e_1 \dots$ of traversed edges, that is, $\mathbf{x}_i \in e_i$. The *region signature* of ξ is the sequence $\rho(\xi) = P_0 P_1 \dots$ of traversed regions, that is, $e_i \in in(P_i)$.

Let ξ be a trajectory whose trace is $\tau(\xi) = \mathbf{x}_0 \dots \mathbf{x}_k$. Let $0 = t_0 < t_1 < \dots < t_k$ be such that $\xi(t_i) = \mathbf{x}_i$. Since ξ is continuous and derivable in the interval (t_i, t_{i+1}), there exists a unique trajectory ξ' with $\xi'(t_i) = \xi(t_i)$ for all $i \in [0, k-1]$, such that the derivative $\dot{\xi}'$ is constant in the interval (t_i, t_{i+1}).

Proposition 1 *For every trajectory ξ there exists a trajectory ξ' with the same initial and final points, and edge and region signatures, such that for each P_i in the region signature, there exists $\mathbf{c}_i \in \phi(P_i)$, such that $\dot{\xi}'(t) = \mathbf{c}_i$ for all $t \in (t_i, t_{i+1})$.*

Hence, in order to solve the reachability problem it is enough to consider trajectories having piecewise constant slopes. Notice that, however, such slopes need not be the same for each occurrence of the same region in the region signature. Hereinafter, we use the word "trajectory" to mean trajectories whose derivatives are piecewise constant.

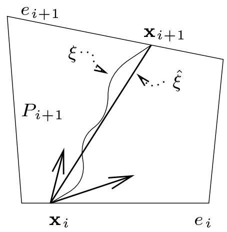


Figure 4: Piecewise constant trajectory.

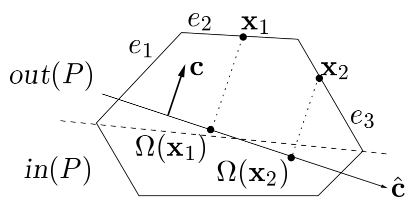


Figure 5: Ordering: $\mathbf{x}_1 \preceq \mathbf{x}_2$.

4.2 Order relation

Consider a region P and let $\mathbf{c} \in \phi(P)$. The mapping $\Omega : \mathbb{R}^2 \to \mathbb{R}$, defined as $\Omega(\mathbf{x}) = \mathbf{x}\ \hat{\mathbf{c}}$, assigns to every $\mathbf{x} \in \mathbb{R}^2$ a value proportional to the length of the projection of the vector $\mathbf{x}$ on the right rotation of $\mathbf{c}$. Indeed, the ordering is given by the direction of $\hat{\mathbf{c}}$ and one can easily see that the relation $\preceq$, defined as $\mathbf{x}_1 \preceq \mathbf{x}_2$ if $\Omega(\mathbf{x}_1) \leq \Omega(\mathbf{x}_2)$, is a dense linear order on $in(P)$ and $out(P)$ (Fig. 5). We use $\prec$ to denote the strict variant of $\preceq$ and say that $e_1 \prec e_2$ iff $e_1 \neq e_2$ and $\mathbf{x}_1 \preceq \mathbf{x}_2$ for every $\mathbf{x}_1 \in e_1, \mathbf{x}_2 \in e_2$. For example, in Fig. 5 we have $e_1 \prec e_2 \prec e_3 \prec e_4$. Notice that the order does not depend on the choice of $\mathbf{c}$.

4.3 Removing self-crossings

We say that a trajectory ξ crosses itself if there exist $t \neq t'$ such that $\xi(t) = \xi(t')$. If a trajectory does not cross itself, the sequence of consecutive intersection points with $in(P)$ or $out(P)$ is monotone with respect to $\preceq$. That is, for every three points $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$ (visited in this order), if $\mathbf{x}_1 \prec \mathbf{x}_2 \prec \mathbf{x}_3$ the trajectory is a "counterclockwise expanding spiral"(Fig. 6(a)) or a "clockwise contracting spiral" (Fig. 6(b)) and if $\mathbf{x}_3 \prec \mathbf{x}_2 \prec \mathbf{x}_1$, the trajectory is a "counterclockwise contracting spiral" (Fig. 6(c)) or a "clockwise expanding spiral" (Fig. 6(d)). On the other hand, if the sequence of intersections points with $in(P)$ or $out(P)$ is monotone (both increasing or both decreasing), the trajectory does not cross itself. Hence,

Lemma 1 *For every trajectory ξ, if ξ does not cross itself, then for every edge e, the sequence $\tau(\xi) \cap e$ is monotone (with respect to $\prec$).*

Now suppose that the trajectory ξ with trace $\tau(\xi) = \mathbf{x}_0 \ldots \mathbf{x}_f$ crosses itself *once* inside the region P. Let $e_1, e_2 \in in(P)$ be the input edges and $e'_1, e'_2 \in out(P)$ be the output ones. Let $\mathbf{x} = \mathbf{x}_i \in e_1$ and $\mathbf{y} = \mathbf{x}_j \in e_2$, with $i < j$, be the points in $\tau(\xi)$ the first and the second times ξ enters P, and let $\mathbf{x}' = \mathbf{x}_{i+1} \in e'_2$ and $\mathbf{y}' = \mathbf{x}_{j+1} \in e'_1$ be the corresponding output points. Let $\mathbf{c}_x, \mathbf{c}_y \in \phi(P) = \angle_{\mathbf{a}}^{\mathbf{b}}$ be the derivatives of ξ in the time intervals (t_i, t_{i+1}) and (t_j, t_{j+1}), respectively. Indeed, $\mathbf{c}_x$ and $\mathbf{c}_y$ are the director vectors of the segments $\overline{\mathbf{x}\mathbf{x}'}$ and $\overline{\mathbf{y}\mathbf{y}'}$, respectively (Fig. 7(a)).

Consider now the segment $\overline{\mathbf{x}\mathbf{y}'}$. Notice that the director vector $\mathbf{c}'_x$ of this segment can be obtained as a positive combination of the vectors $\mathbf{c}_x$ and $\mathbf{c}_y$. That is, there exist $\alpha_1, \alpha_2 > 0$ such that $\mathbf{c}'_x = \alpha_1 \mathbf{c}_x + \alpha_2 \mathbf{c}_y$ (see Fig. 7(b)). Since $\mathbf{c}_x, \mathbf{c}_y \in \phi(P) = \angle_{\mathbf{a}}^{\mathbf{b}}$, there exist $\gamma_1, \gamma_2, \delta_1, \delta_2 > 0$ such

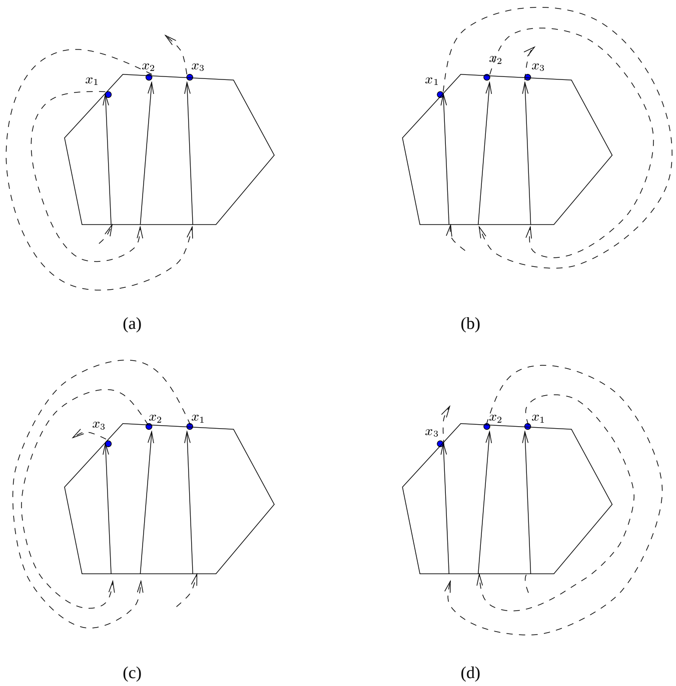


Figure 6: (a) $\mathbf{x_1} \prec \mathbf{x_2} \prec \mathbf{x_3}$: counterclockwise expanding spiral; (b) $\mathbf{x_1} \prec \mathbf{x_2} \prec \mathbf{x_3}$: clockwise contracting spiral; (c) $\mathbf{x_3} \prec \mathbf{x_2} \prec \mathbf{x_1}$: counterclockwise contracting spiral; (d) $\mathbf{x_3} \prec \mathbf{x_2} \prec \mathbf{x_1}$: clockwise expanding spiral.

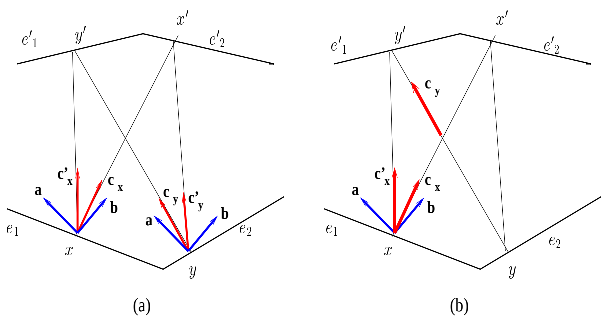


Figure 7: A trajectory that crosses itself.

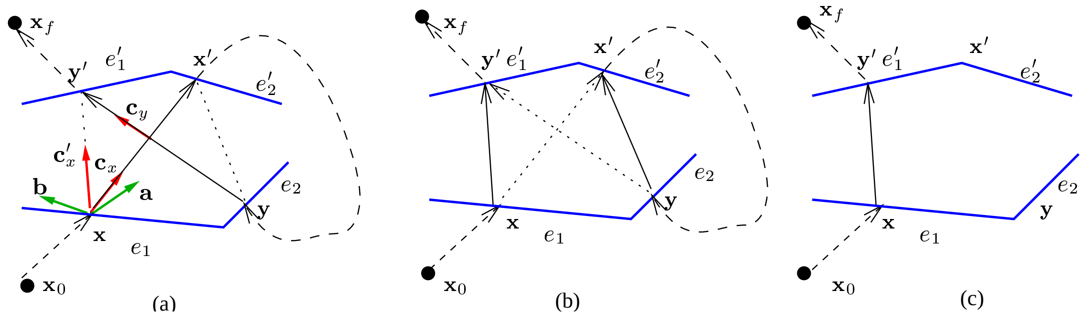


Figure 8: Obtaining a non-crossing trajectory

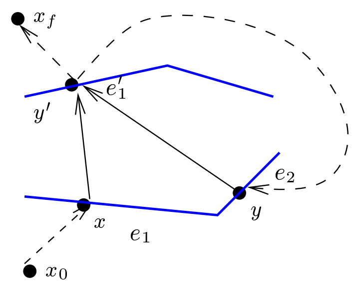


Figure 9: A "degenerate" self crossing.

that $\mathbf{c}_x' = \alpha_1(\gamma_1\mathbf{a} + \gamma_2\mathbf{b}) + \alpha_2(\delta_1\mathbf{a} + \delta_2\mathbf{b}) = (\alpha_1\gamma_1 + \alpha_2\delta_1)\mathbf{a} + (\alpha_1\gamma_2 + \alpha_2\delta_2)\mathbf{b}$. Thus, $\mathbf{c}_x' \in \phi(P)$. Similarly we can prove that $\mathbf{c}_y'$ is a positive combination of $\mathbf{a}$ and $\mathbf{b}$. Hence, there exists a trajectory ξ' that does not cross itself in P having a trace $\tau(\xi') = \mathbf{x}_1 \ldots \mathbf{x}\mathbf{y}' \ldots \mathbf{x}_f$ (Fig. 8). Notice that the result also works for the *degenerate* case when the trajectory crosses itself at an edge (or vertex) (see Fig. 9).

If the trajectory ξ crosses itself more than once in region P, then the number of times the trajectory ξ', obtained by cutting away the loop (Fig. 8(c)), crosses itself in P is strictly smaller than the number of times ξ does it (see Fig. 10). After replacing $\overline{\mathbf{x}\mathbf{x}'}$ and $\overline{\mathbf{y}\mathbf{y}'}$ by $\overline{\mathbf{x}\mathbf{y}'}$, the intersection q of $\overline{\mathbf{x}\mathbf{x}'}$ and $\overline{\mathbf{y}\mathbf{y}'}$ disappears. If the new segment of line $\overline{\mathbf{x}\mathbf{y}'}$ crosses another segment $\overline{\mathbf{z}\mathbf{z}'}$ (say at a point t), then $\overline{\mathbf{z}\mathbf{z}'}$ necessarily crosses either $\overline{\mathbf{x}\mathbf{x}'}$ (at r) or $\overline{\mathbf{y}\mathbf{y}'}$ (at s) -or both-, before the transformation. The above is due to the fact that if $\overline{\mathbf{z}\mathbf{z}'}$ crosses one side of the triangle $\mathbf{x}\mathbf{y}'q$ then it must also cross one of the other sides of the triangle, say at r. Thus, no new crossing can appear and the number of crossings in the new configuration is always less than in the old one.

Lemma 2 *For every trajectory ξ that crosses itself at least once, there exists a trajectory ξ' with the same initial and final points of ξ having a number of self-crossings strictly smaller.*

Proposition 2 (Existence of a non-crossing trajectory) *If there exists an arbitrary trajectory from points $\mathbf{x}_0 \in e_0$ to $\mathbf{x}_f \in e_f$ then there always exists a non-crossing trajectory between them.*

Proof: By induction on the number n of times the trajectory crosses itself using Lemma 2 in the induction step. □

Hence, in order to solve the reachability problem we only need to consider non-crossing trajectories with piecewise constant derivatives. In what follows, we only deal with trajectories of this kind.

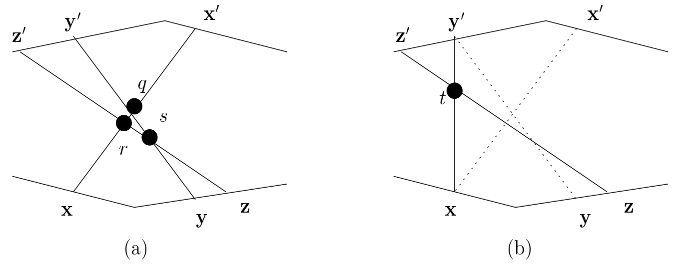


Figure 10: The number of crossings decreases after eliminating the self crossings. (a) Before (3 crossings); (b) After (1 crossing).

4.4 Properties of edge signatures

Let ξ be a trajectory with trace $\tau(\xi) = \mathbf{x}_0 \dots \mathbf{x}_p$, edge signature $\sigma(\xi) = e_0 \dots e_p$, and region signature $\rho(\xi) = P_0 \dots P_p$. An edge e is said to be *abandoned* by ξ after position i, if $e_i = e$ and for some j, k, $i \leq j < k$, $P_j \dots P_k$ forms a region cycle and $e \notin \{e_{i+1}, \dots, e_k\}$. Since trajectories are *finite* we should add the trivial case when $e \neq e_j$ for all $j, j > i$.

Lemma 3 (Abandonment is Irreversible) *For every trajectory ξ and edge e, if e is abandoned by ξ after position i, e will not appear in $\sigma(\xi)$ at any position $j > i$.*

Proof: It follows from Lemma 1. See Claim 2 in [4]. □

Intuitively, the abandonment lemma above guarantees that any edge that occurs in a prefix of an edge signature but not appears in a cycle following this prefix, then it cannot occur anymore in any postfix (starting with the cycle) of the edge signature.

Given a sequence s, ε denotes the empty sequence whereas $first(s)$ and $last(s)$ are the first and last elements of the sequence respectively. An edge signature $\sigma(\xi)$ can be canonically expressed as a sequence of edges and cycles of the form $\sigma_c(\xi) = r_1 s_1^{k_1} r_2 s_2^{k_2} \dots r_n s_n^{k_n} r_{n+1}$, 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;
3. For all $i, 1 \leq i \leq n$, s_i and r_{i+1} are disjoint;
4. $r_{n+1} \neq \varepsilon$. Moreover, $r_{n+1} = first(s_n)$ if $\sigma(\xi)$ ends in a loop and $first(r_{n+1}) \neq first(s_n)$ otherwise.

This canonical representation can be obtained as follows.

Algorithm $\mathcal{A}$
Let $\sigma(\xi) = e_1 \dots e_{p-1} e_p$ be an edge signature. Starting from e_{p-1} and traversing backwards, take the first edge that occurs the second time. If there is no such edge, then trivially the signature can be expressed in a canonical way and we are done. Otherwise, suppose that the edge e_j occurs again at position i (i.e. $e_i = e_j$ with $i < j$), thus $\sigma_c(\xi) = wsr$, where w, s and r are obtained as follows, depending on the repeated edge:

$$w = e_0 \dots e_i$$
$$s = e_{i+1} \dots e_j$$
$$r = e_{j+1} \dots e_{p-1}$$

Clearly r is not a cycle and s is a simple cycle with no repeated edges. We continue the analysis with w. Let $k_m = \max\{l \mid s^l \text{ is a suffix of } w\}$. Thus, $\sigma_c(\xi) = w' s^k r$ with $w' = e_0 \dots e_h$ (a prefix of w) and $k = k_m + 1$. We repeat recursively the procedure above with w'. Adding the edge e_p to the last r (at the end) we obtain $\sigma_c(\xi) = r_1 s_1^{k_1} \ldots r_n s_n^{k_n} r_{n+1}$ that is a canonical representation of signature σ. $\square$

The representation obtained by the above algorithm give rises to the following theorem.

Theorem 4 (Representation Theorem) *Let $\sigma(\xi) = e_1 \ldots e_p$ be a (concrete) edge signature, then it can always be written as $\sigma_c(\xi) = r_1 s_1^{k_1} \ldots r_n s_n^{k_n} r_{n+1}$, 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).* $\square$

Given a representation of a signature, obtained as before, we have the following definition.

Definition 1 *Let $\sigma(\xi) = e_1 \ldots e_p$ be a (concrete) edge signature and $\sigma_c(\xi) = r_1 s_1^{k_1} \ldots r_n s_n^{k_n} r_{n+1}$ be its representation (obtained by Algorithm $\mathcal{A}$). Then we define the* type of a signature σ *as type$(\sigma(\xi)) = r_1, s_1, \ldots, r_n, s_n, r_{n+1}$.* $\square$

Moreover, from Theorem 4 we conclude that we can obtain an abstraction of a concrete signature as shown in the following corollary.

Corollary 5 *For each (concrete) edge signature $\sigma(\xi) = e_1 \ldots e_i \ldots e_p$ it exists a type of signature of the form type$(\sigma(\xi)) = r_1, s_1, \ldots, r_n, s_n, r_{n+1}$.*

Proof: The result follows from Theorem 4 constructing $\sigma_c(\xi) = r_1 s_1^{k_1} \ldots r_n s_n^{k_n} r_{n+1}$ and then abstracting away the number of times each cycle is iterated. $\square$

Notice that the "preprocessing" (taking away the last edge e_p) is done in order to differentiate edges signatures that end with a cycle from those that do not. There exists many other (maybe easier) ways of decomposing a signature σ in a canonical form (in particular, traversing forward instead of backwards), but the one chosen here permits a clearer and simpler presentation of the reachability algorithm. In fact in this canonical form, the last visited edge in a cycle $e_1 \ldots e_k$ is always the last one (e_k). In what follows and when no confusion may arise, we will denote the type of a given signature $\sigma_c(\xi) = r_1 s_1^{k_1} \ldots r_n s_n^{k_n} r_{n+1}$ as *type*$(\sigma) = r_1 s_1 \ldots r_n s_n r_{n+1}$ or even just as $\sigma = r_1 s_1 \ldots r_n s_n r_{n+1}$.

Example. Let us consider the following examples. Suppose that $\sigma = abcdbcefgefgefgefhi$. Then, after applying once the above procedure we obtain that $\sigma_c = w(s_2)^3 r_1$, with $w = abcdbcef$; $s_2 = gef$; $r_1 = h$. Applying the procedure once more to w we obtain $w = w'(s_3)^1 r_2$ with $w' = r_3 = abc$; $s_3 = dbc$; $r_2 = ef$. Putting all together and adding the last edge (i) gives $\sigma_c = abc(dbc)^1 ef(gef)^3 hi$ with type *type*$(\sigma) = abc, dbc, ef, gef, hi$. Suppose now, that the signature ends with a cycle: $\sigma = abcdbcefgefgefgefgef$. In this case we apply the preprocessing obtaining $\sigma_c = w(s_2)^4 r_1$ with $w = abcdbce$; $s_2 = fge$; $r_1 = \varepsilon$. Applying the procedure to w we finally obtain $w = w'(s_3)^1 r_2$ with $w' = r_3 = abc$; $s_3 = dbc$; $r_2 = e$ and that gives (adding f to the end) $\sigma_c = abc(dbc)^1 e(fge)^4 f$. $\square$

The following lemma shows some important properties of types of signatures.

Lemma 6 *The type of a signature σ, type(σ), has the following properties:*

1. *For every $1 \leq i \neq j \leq n+1$, r_i and r_j are disjoint;*
2. *For every $1 \leq i \neq j \leq n$, s_i and s_j are different;*
3. *If v is a vertex appearing in type(σ), then it can only occur exactly once in r_i for some $1 \leq i \leq n+1$ in σ. Moreover, $v \notin last(r_i)$ unless $i = n+1$.*

Proof:

1. Let $e \in r_i$; we consider two cases:

   (a) $e \notin s_i$: The result follows immediately from Lemma 3 (e cannot occur in any r_j, $j > i$);

   (b) $e \in s_i$: Suppose that $e \in r_{i+1}$. Then we have $s_i = e_1 \cdots e_i \cdots e_k$ and $r_{i+1} = e_{k+1} \cdots e_j \cdots e_l$, with $e_i = e_j$, that contradicts the hypothesis of σ being canonical: the construction of σ was doing backwards, and in this case we should have a cycle $s = e_{i+1} \cdots e_k e_{k+1} \cdots e_j$. If $e \in r_j$ (for any $j > i+1$) then again we have two cases: $e \in s_{j-1}$ or $e \notin s_{j-1}$; the first case contradicts canonicity of σ and the latter contradicts Lemma 3.

2. Let $s_i = e_1, \ldots, e_k$ be a simple cycle. After cycling k_i times the cycle is abandoned by edge e_k (guaranteed by being in a canonical form). Let P be a region s.t. $e_k \in in(P)$ and consider the unfolding of the last iteration and its continuation:

$$\ldots, e_1, e_2, \ldots, e_k, e, \ldots$$

   where $e = first(r_{i+1})$, $e_k \in in(P)$ and $e_1, e \in out(P)$ ($e_1 \neq e$). By the ordering between edges we have that either $e \prec e_1$ or $e_1 \prec e$. By the monotonicity of the trajectory, in both cases e_1 cannot occur after e in σ. Thus, any other cycle s_j, with $i < j$, differs from s_i at least on e_1. Hence, all the cycles are different.

3. Direct from Lemma 3. □

As a direct consequence of the lemma above we have the following proposition.

Proposition 3 *The set of types of edge signatures is finite.*

Thus, to solve the reachability problem we can proceed by examining one by one the types of signatures.

5 Affine operators

5.1 Truncated affine multivalued operators

Let us introduce some useful notions. An *affine function* $f : \mathbb{R} \to \mathbb{R}$ is defined by a formula $f(x) = ax + b$ with $a > 0$. An *affine multivalued operator* (AMF) $F : \mathbb{R} \to 2^{\mathbb{R}}$ is determined by two affine functions $f_l(x)$ and $f_u(x)$ and maps x to the interval $\langle f_l(x), f_u(x) \rangle$, where $\langle a, b \rangle$ means (a, b), $[a, b]$, $(a, b]$ or $[a, b)$:

$$F(x) = \langle f_l(x), f_u(x) \rangle$$

with $dom(F) = \{x \mid f_l(x) \leq f_u(x)\}$. We use the notation $F = \langle f_l, f_u \rangle$.
Such an operator can be naturally extended to subsets of $\mathbb{R}$:

$$F(S) = \bigcup_{x \in S} F(x)$$

In particular, if $S = \langle l, u \rangle$:

$$F(\langle l, u \rangle) = \langle f_l(l), f_u(u) \rangle$$

where $dom(F) = \{\langle l, u \rangle \mid f_l(l) \leq f_u(u)\}$. A *truncated affine multivalued operator* (TAMF) $\mathcal{F} : \mathbb{R} \to 2^{\mathbb{R}}$ is determined by an affine multivalued operator F and intervals S and J as follows:

$$\mathcal{F}_{F,S,J}(x) = \begin{cases} F(x) \cap J & \text{if } x \in S \\ \emptyset & \text{otherwise} \end{cases}$$

or, what is the same

$$\mathcal{F}_{F,S,J}(x) = F(\{x\} \cap S) \cap J.$$

We use cal style to denote TAMFs operators and in general we will write $\mathcal{F}$ instead of $\mathcal{F}_{F,S,J}$.[1]

Such operators can be also extended to sets and in particular to intervals, as shown in what follows.

$$\begin{array}{lll}
\mathcal{F}(I) & = \bigcup_{x \in I} \mathcal{F}(x) & \text{by definition} \\
 & = \bigcup_{x \in I} F(\{x\} \cap S) \cap J & \text{by definition of } \mathcal{F} \\
 & = F(\cup_{x \in I} \{x\} \cap S) \cap J & \\
 & = F(I \cap S) \cap J &
\end{array}$$

A TAMF operator is in *normal form* if $S \subset F^{-1}(J)$. Indeed, every TAMF can be represented in normal form. Let $S_{norm} = S \cap F^{-1}(J)$ (where $F^{-1} = \langle f_u^{-1}, f_l^{-1} \rangle$).

In what follows, we consider just TAMFs in normal form. Before showing that the class of functions above defined are closed under composition we prove the following lemma.

Lemma 7 *Let $F = [f_1, f_2]$ be a multivalued affine operator. If $I \cap H \neq \emptyset$ or if $I = \emptyset \vee H = \emptyset$ then $F(I \cap H) = F(I) \cap F(H)$.*

Proof: Let $I = [i_l, i_u]$ and $H = [h_l, h_u]$. First, suppose that $I = \emptyset$. Then $F(I \cap H) = F(\emptyset) = \emptyset$ and $F(I) \cap F(H) = \emptyset \cap F(H) = \emptyset$.
Suppose now that $I \cap H \neq \emptyset$, then

$$\begin{array}{lll}
F(I \cap H) & = F([\max\{i_l, h_l\}, \min\{i_u, h_u\}]) & \text{by def of } \cap \\
 & = [f_1(\max\{i_l, h_l\}), f_2(\min\{i_u, h_u\})]) & \text{by def of } F \\
 & = [\max\{f_1(i_l), f_1(h_l)\}), \min\{f_2(i_u), f_2(h_u)\})]) & \text{by monotonicity of } F \\
 & = [f_1(i_l), f_2(i_u)] \cap [f_1(h_l), f_2(h_u)] & \text{by def of intersection} \\
 & = F([i_l, i_u]) \cap F([h_l, h_u]) & \text{by def of } F \\
 & = F(I) \cap F(H) &
\end{array}$$

□

Lemma 8 (composition of affine operations) *Affine functions, affine multivalued operators, and truncated affine multivalued operators are closed under composition.*

Proof:

Affine functions: For $f(x) = ax + b$ and $g(x) = cx + d$ the composition $g \circ f(x) = c(ax + b) + d = (ca)x + (cb + d)$ has the required form. Notice, that the coefficient ca is positive since c and a are positive.

Affine multi-valued operators For $F = \langle f_l, f_u \rangle$ and $H = \langle h_l, h_u \rangle$, the composition $H \circ F$ is nothing other than $\langle h_l \circ f_l, h_u \circ f_u \rangle$.

[1] The intersection $\langle_1 a_1, b_1 \rangle_1 \cap \langle_2 a_2, b_2 \rangle_2 = \langle_3 a_3, b_3 \rangle_3$ is defined as follows:

$$\begin{array}{l}
a_3 = \max\{a_1, a_2\} \\
b_3 = \min\{b_1, b_2\} \\
\langle_3 = \begin{cases} \langle_1 & \text{if } a_1 > a_2 \\ \langle_2 & \text{if } a_1 < a_2 \\ \max\{\langle_1, \langle_2\} & \text{if } a_1 = a_2 \end{cases} \\
\rangle_3 = \begin{cases} \rangle_1 & \text{if } b_1 < b_2 \\ \rangle_2 & \text{if } b_1 > b_2 \\ \min\{\rangle_1, \rangle_2\} & \text{if } b_1 = b_2 \end{cases}
\end{array}$$

where $'[' < '('$ and $')' < ']'$.

Truncated affine multi-valued operators For $\mathcal{F}_1(x) = F_1(\{x\} \cap S_1) \cap J_1$ and $\mathcal{F}_2(x) = F_2(\{x\} \cap S_2) \cap J_2$, $\mathcal{F}_2 \circ \mathcal{F}_1(x) = \mathcal{F}_{F',S',J'}(x)$ with $F' = F_2 \circ F_1$, $J' = J_2 \cap F_2(J_1 \cap S_2)$ and $S' = S_1 \cap F_1^{-1}(J_1 \cap S_2)$:

$$\begin{aligned} \mathcal{F}_2 \circ \mathcal{F}_1(x) &= F_2(F_1(\{x\} \cap S_1) \cap J_1) \\ &= F_2((F_1(\{x\} \cap S_1) \cap J_1) \cap S_2) \cap J_2; \end{aligned} \tag{1}$$

we split the proof into two cases

1. $x \in S_1$: in this case $F_1(\{x\} \cap S_1) = F_1(x)$ and then expression (1) is equal to $F_2((F_1(x) \cap J_1) \cap S_2) \cap J_2$ that is equal to

$$F_2((F_1(x) \cap (J_1 \cap S_2)) \cap J_2 \tag{2}$$

   We consider two cases

   (a) $F_1(x) \cap (J_1 \cap S_2) \neq \emptyset$: In this case the distributivity holds (see Lemma 7) and expression (2) is equal to $F_2(F_1(x)) \cap F_2(J_1 \cap S_2) \cap J_2$. But $F_1(x) \cap (J_1 \cap S_2) \neq \emptyset$ is equivalent to $x \in F_1^{-1}(J_1 \cap S_2)$. Hence,

$$\mathcal{F}_2(\mathcal{F}_1(x)) = F_2 \circ F_1(x) \cap F_2(J_1 \cap S_2) \cap J_2$$

   under the condition $x \in S_1 \wedge x \in F_1^{-1}(J_1 \cap S_2)$, i.e. $x \in S_1 \cap F_1^{-1}(J_1 \cap S_2)$.

   (b) $F_1(x) \cap (J_1 \cap S_2) = \emptyset$: here (2) is equal to the empty set and then $\mathcal{F}_2 \circ \mathcal{F}_1 = \emptyset$

2. $x \notin S_1$: that is, $\{x\} \cap S_1 = \emptyset$ and then $F_1(\{x\} \cap S_1) = \emptyset$, that implies $\mathcal{F}_2 \circ \mathcal{F}_1 = \emptyset$.

From the cases above we have

$$\mathcal{F}_2 \circ \mathcal{F}_1(x) = \mathcal{F}_{F',S',J'}(x) = \begin{cases} F'(x) \cap J' & \text{if } x \in S' \\ \emptyset & \text{otherwise} \end{cases}$$

with $F' = F_2 \circ F_1$, $J' = J_2 \cap F_2(J_1 \cap S_2)$ and $S' = S_1 \cap F_1^{-1}(J_1 \cap S_2)$. $\square$

Example. Let $x \in J_0$ (where $J_0 = [0,1]$), $F_1(x) = (2x - \frac{3}{5}, 3x+5]$ and $F_2(x) = [5x+2, 7x+6]$ be two (non-truncated) affine multi-valued functions, $\mathcal{F}_1 = F_1 \cap J_1$ (with $J_1 = (1, 6]$), and $\mathcal{F}_2 = F_2 \cap J_2$ (with $J_2 = [6, 10)$) their truncated versions. We have that $F_1^{-1}(\langle y_1, y_2 \rangle) = \langle \frac{y_1-5}{3}, \frac{5y_2+3}{10} \rangle$ and $F_2^{-1}(\langle y_1, y_2 \rangle) = \langle \frac{y_1-6}{7}, \frac{y_2-2}{5} \rangle$. To obtain $\mathcal{F}_2 \circ \mathcal{F}_1(x)$ we need to compute F', S' and J' as in Lemma 8 but first we compute S_1 and S_2:

$$\begin{aligned} S_1 &= F_1^{-1}(J_1) \cap J_0 \\ &= F_1^{-1}((1,6]) \cap [0,1] \\ &= (-\frac{4}{3}, \frac{33}{10}) \cap [0,1] \\ &= [0,1] \end{aligned}$$

$$\begin{aligned} S_2 &= F_2^{-1}(J_2) \cap J_1 \\ &= F_2^{-1}([6,10)) \cap (1,6] \\ &= [0, \frac{8}{5}) \cap (1,6] \\ &= (1, \frac{8}{5}) \end{aligned}$$

$$\begin{aligned}
S' &= S_1 \cap F_1^{-1}(J_1 \cap S_2) \\
&= [0,1] \cap F_1^{-1}((1,6] \cap (1, \frac{8}{5})) \\
&= [0,1] \cap F_1^{-1}((1, \frac{8}{5})) \\
&= [0,1] \cap (-\frac{4}{3}, \frac{11}{10}) \\
&= [0,1]
\end{aligned}$$

$$\begin{aligned}
J' &= J_2 \cap F_2(J_1 \cap S_2) \\
&= [6,10) \cap F_2((1,6] \cap (1, \frac{8}{5})) \\
&= [6,10) \cap F_2((1, \frac{8}{5})) \\
&= [6,10) \cap (7,10) \\
&= (7,10)
\end{aligned}$$

$$\begin{aligned}
F'(x) &= F_2 \circ F_1(x) \\
&= (5(2x - \frac{3}{5}) + 2,\ 7(3x+5) + 6] \\
&= (10x - 1,\ 21x + 41]
\end{aligned}$$

Hence, the truncated affine multi-valued operator $\mathcal{F}_2 \circ \mathcal{F}_1(x)$ is

$$\mathcal{F}_2 \circ \mathcal{F}_1(x) = \begin{cases} (10x-1,\ 21x+41] \cap (7,10) & \text{if } x \in [0,1] \\ \emptyset & \text{otherwise} \end{cases}$$

Notice that the result can be extended to any interval $\langle l, u \rangle$. $\square$

In the sequel we need some properties of iterations of affine operations. We start with affine functions.

Lemma 9 *Let f be an affine function, x_0 be any initial point and $x_n = f^n(x_0)$. The following properties hold*

1. *The sequence x_n is monotonous;*
2. *It converges to a limit x^* (finite or infinite), which can be effectively computed knowing a, b and x_0.*

Proof: Monotonicity of x_n follows from the identity $x_{n+1} - x_n = a^n(x_1 - x_0)^2$. Existence of limit is immediate from the monotonicity. To calculate the limit several cases should be considered (see Fig. 11):

[2] $x_n = f^n(x_0) = a^n x_0 + a^{n-1} b + \ldots + ab + b$. Thus, $x_{n+1} - x_n = (a^{n+1} x_0 + a^n b + \ldots + ab + b) - (a^n x_0 + a^{n-1} b + \ldots + ab + b) = (a^{n+1} x_0 + a^n b - a^n x_0) = a^n(a x_0 + b - x_0)$, from where we conclude that $x_{n+1} - x_n = a^n(x_1 - x_0)$.

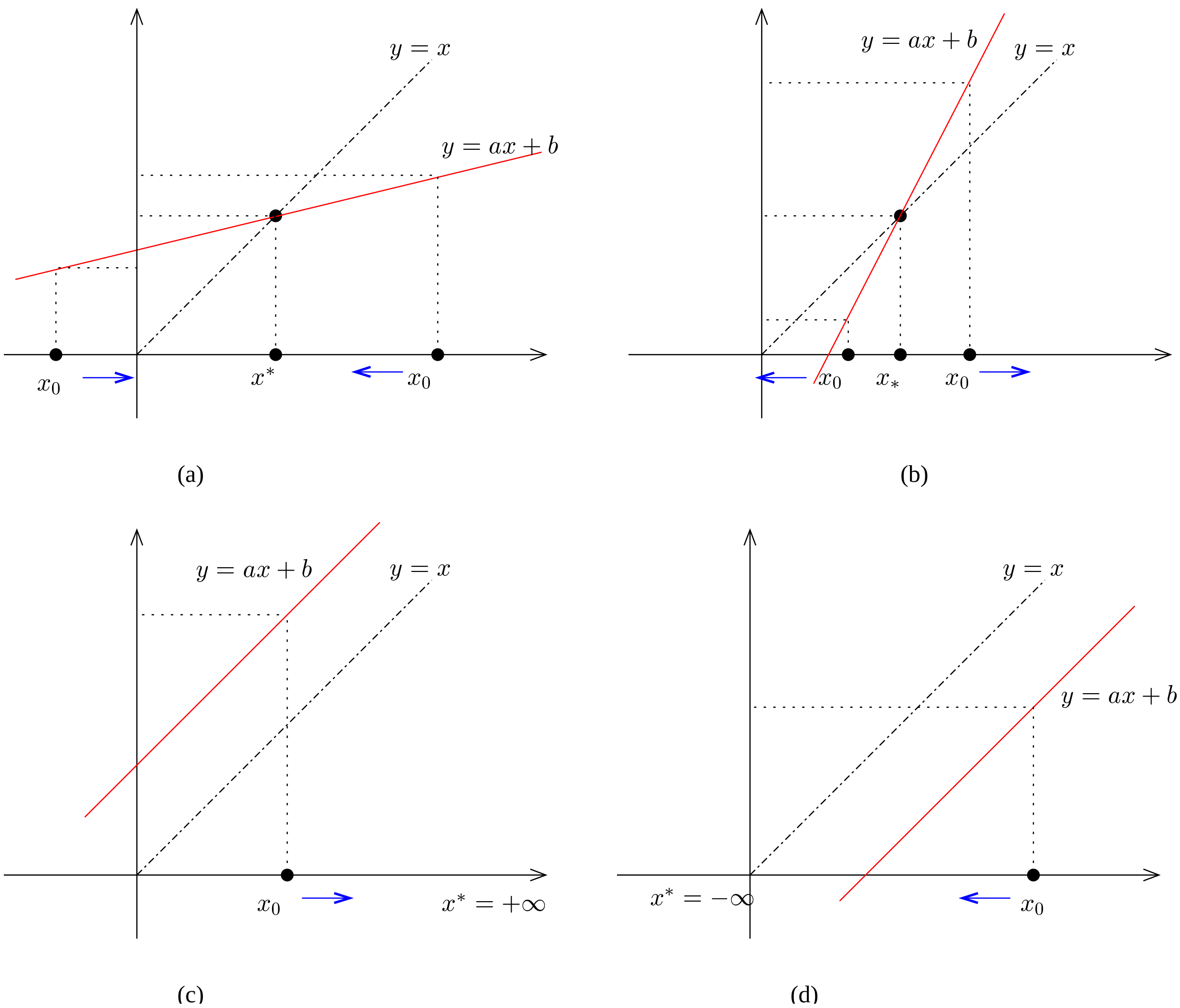


Figure 11: (a): $a < 1$, $x^* = b/(1-a)$; (b):$a > 1$, $x^* = -\infty$ if $x_0 < x_*$,$x^* = x_0$ if $x_0 = x_*$, $x^* = +\infty$ if $x_0 > x_*$; (c): $a = 1$ and $b > 0$, $x^* = +\infty$; (d): $a = 1$ and $b < 0$, $x^* = -\infty$

$a < 1$: In this case the limit is finite and it is the unique fixpoint of the function f: $ax^* + b = x^*$, and hence $x^* = b/(1-a)$.

$a = 1$: In this case
$$x^* = \begin{cases} -\infty & \text{if} \quad b < 0 \\ x_0 & \text{if} \quad b = 0 \\ \infty & \text{if} \quad b > 0 \end{cases}$$

$a > 1$: In this case we should calculate first the (unstable) fixpoint $x_* = b/(1-a)$. However in this case the limit is not necessary equal to x_* . Namely,
$$x^* = \begin{cases} -\infty & \text{if} \quad x_0 < x_* \\ x_0 & \text{if} \quad x_0 = x_* \\ \infty & \text{if} \quad x_0 > x_* \end{cases}$$

□

This result can be easily extended to intervals and affine multi-valued operators.

Lemma 10 *Let $\langle l_0, u_0 \rangle$ be any initial point and $\langle l_n, u_n \rangle = F^n(\langle l_0, u_0 \rangle)$. The following properties hold*

1. *The sequence l_n and u_n are monotonous;*
2. *They converge to limits l^* and u^* (finite or infinite), which can be effectively computed.*

Proof: Apply the previous lemma to the sequences l_n and u_n. □

5.2 Some special properties

Lemma 11 *Let $f(x) = ax + b$ be an affine function ($a > 0$). Then, the following holds (for $n > 0$):*

1. *If $x \le y$ then $f^n(x) \le f^n(y)$;*
2. *If $x \le f(x)$ then $f(x) \le f^n(x)$;*
3. *If $f(y) \le y$ then $f(y)^n \le f(y)$;*
4. *If $f(x) \le x$ and $f(x) \le f^n(y)$ then $x \le y$;*
5. *If $x \le f(x)$ and $x \le y$ then $f(x) \le f^n(y)$;*
6. *If $f(y) \le y$ and $x \le y$ then $f^n(x) \le f(y)$;*
7. *If $y \le f(y)$ and $f^n(x) \le f(y)$ then $x \le y$.*

Proof: The result follows directly from simple algebraic properties of partial order and monotonicity. □

Lemma 12 *Let $I = \langle l_I, u_I \rangle$ and $H = \langle l_H, u_H \rangle$ be two intervals and $F(I) = \langle f_l(l_I), f_u(u_I) \rangle$ an AMVF. Then,*

1. *F is monotonic;*
2. *F is continuous.*

Proof:

1. We have to prove that if $I \subseteq H$ then $F(I) \subseteq F(H)$. If $I = \emptyset$ or $H = \emptyset$ the result holds trivially, hence we will consider $I \neq \emptyset$ and $H \neq \emptyset$. Suppose that $I \subseteq H$ (i.e. $l_H \leq l_I \leq u_I \leq u_H$), then

$$\begin{array}{lll} x \in F(I) & \Longrightarrow x \in \langle f_l(l_I), f_u(u_I)\rangle & \text{(by def. of } F) \\ & \Longrightarrow x \in \langle f_l(l_H), f_u(u_H)\rangle & \text{(by hypothesis and monotonicity of } f_l \text{ and } f_u) \\ & \Longrightarrow x \in F(H) & \text{(by def. of } F) \end{array}$$

Thus, $F(I) \subseteq F(H)$, e.e. F is monotonic.

2. We have to prove that $F(\cup_i I_i) = \bigcup_i (F(I_i))$. Let $A = \cup_i I_i$.

$\subseteq$: Suppose that $y \in F(A)$, then $y \in \bigcup_{x \in A} F(x)$ (by def. of F applied to sets). Thus, for some $x \in A$, there exist an interval I_j s.t. $x \in I_j$ and then $y \in F(x)$. By definition and monotonicity of F we have that $F(x) \subseteq F(I_j)$, that implies $y \in F(I_j)$ and then $y \in \bigcup_i F(I_i)$. Hence, $F(\cup_i I_i) \subseteq \bigcup_i (F(I_i))$.

$\supseteq$: Suppose that $y \in \bigcup_i (F(I_i))$, then exist an interval I_j s.t. $y \in F(I_j)$ and by monotonicity of F, $y \in F(\cup_i I_i)$. Hence $\bigcup_i (F(I_i)) \subseteq F(\cup_i I_i)$.

From both results we have that $F(\cup_i I_i) = \bigcup_i (F(I_i))$, i.e. F is continuous.

□

We have the same result for TAMVF operators.

Lemma 13 *Let I and H be two intervals and $\mathcal{F}(I) = F(I \cap S) \cap J$ a TAMVF. Then,*

1. *$\mathcal{F}$ is monotonic;*
2. *$\mathcal{F}$ is continuous.*

Proof:

1. We have to prove that if $I \subseteq H$ then $\mathcal{F}(I) \subseteq \mathcal{F}(H)$. If $I = \emptyset$ or $H = \emptyset$ the result holds trivially. Suppose that I and H are non-empty and that $I \subseteq H$. Then,

$$\begin{array}{lll} x \in \mathcal{F}(I) & \Longrightarrow x \in F(I \cap S) \cap J & \text{(by def. of } \mathcal{F}) \\ & \Longrightarrow x \in F(I \cap S) \wedge x \in J & \\ & \Longrightarrow x \in F(H \cap S) \wedge x \in J & \text{(by hypothesis and monotonicity of } F) \\ & \Longrightarrow x \in F(H \cap S) \cap J & \text{(by def. of } \mathcal{F}) \\ & \Longrightarrow x \in \mathcal{F}(H) & \end{array}$$

Hence, $\mathcal{F}(I) \subseteq \mathcal{F}(H)$ and $\mathcal{F}$ is monotonic.

2. We prove that $\mathcal{F}(\cup_i I_i) = \bigcup_i (\mathcal{F}(I_i))$:

$$\begin{array}{lll} \mathcal{F}(\cup_i I_i) & = F(\cup_i I_i \cap S) \cap J & \text{(by def. of } \mathcal{F}) \\ & = F(\cup_i (I_i \cap S)) \cap J & \text{(by distrib.)} \\ & = \bigcup_i F(I_i \cap S) \cap J & \text{(by monotonicity of } F) \\ & = \bigcup_i \mathcal{F}(I_i) & \text{(by definition of } \mathcal{F}) \end{array}$$

We conclude that $\mathcal{F}$ is continuous.

□

In what follows we will use the following notation: given $I = \langle l, u\rangle$, $x < I$ is equivalent to $x < l$ if I is a (left) closed interval or to $x \leq l$ if I is an (left) open interval; dually for $x > I$ (w.r.t. u). Given two intervals $\langle l_1, u_1\rangle$ and $\langle l_2, u_2\rangle$, $l_1 < l_2$ it means $l_1 \leq l_2$ if $\langle l_2, u_2\rangle$ is a (left) closed interval and $\langle l_2, u_2\rangle$ is an (left) open interval and it means $l_1 < l_2$ otherwise. Similarly for the right extremes of the intervals.

We use the notation $\hat{\mathcal{F}}$ for truncated affine multi-valued operators with $S = J$; i.e. the image and the domain coincide and then $\hat{\mathcal{F}}(I) = F(I \cap H) \cap H$.Let $H = \langle l_H, u_H\rangle$ and $I \cap H = \langle l_{I\cap H}, u_{I\cap H}\rangle$. For this kind of functions we have the following result

Lemma 14 (Fundamental lemma) *Let $\hat{\mathcal{F}}$ be a truncated affine multi-valued operator. Then $\hat{\mathcal{F}}^n(I) = F^n(I \cap H) \cap H$.*

Proof:

Base case ($n = 1$): By definition $\hat{\mathcal{F}}(I) = F(I \cap H) \cap H$.

Inductive step ($n \geq 2$): We have to prove that $\hat{\mathcal{F}}^{n+1} = F^{n+1}(I \cap H) \cap H$.

$$\begin{aligned} \hat{\mathcal{F}}^{n+1}(I) &= \hat{\mathcal{F}}(\hat{\mathcal{F}}^n(I)) & \\ &= \hat{\mathcal{F}}(F^n(I \cap H) \cap H) & \text{(By inductive hypothesis)} \\ &= F(F^n(I \cap H) \cap H) \cap H & \text{(By definition of } \hat{\mathcal{F}}\text{)} \end{aligned} \tag{3}$$

We consider two cases:

1. $\mathbf{F^n(I \cap H) \subseteq H}$: In this case (3) is equal to $F(F^n(I \cap H)) \cap H$ that is the same as $F^{n+1}(I \cap H) \cap H$.
2. $\mathbf{F^n(I \cap H) \not\subseteq H}$: We split the proof into two sub-cases
   (a) $\mathbf{F^n(I \cap H) \cap H = \emptyset}$: By inductive hypothesis, $\hat{\mathcal{F}}^n(I) = \emptyset$ and then $\hat{\mathcal{F}}^{n+1}(I) = \emptyset$. On the other hand, that $F^{n+1}(I \cap H) \cap H = \emptyset$ follows from the supposition by monotonicity.
   (b) $\mathbf{F^n(I \cap H) \cap H \neq \emptyset}$: In this case the distributivity holds (by Lemma 7) and expression (3) is equal to $F^{n+1}(I \cap H) \cap F(H) \cap H$, that can be written as

$$[\max\{F^{n+1}(l_{I \cap H}), F(l_H), l_H\},\ \min\{F^{n+1}(u_{I \cap H}), F(u_H), u_H\}]$$

   We analyze the following cases depending on the relationship between $F(H)$ and H as shown in Figure 12
      i. $F(l_H) < l_H \wedge F(u_H) < u_H$: The first condition ($F(l_H) < l_H$) implies $\max\{F^{n+1}(l_{I \cap H}), F(l_H), l_H\} = \max\{F^{n+1}(l_{I \cap H}), l_H\}$. On the other hand, $F(u_H) < u_H$ implies $F^{n+1}(u_{I \cap H}) < F(u_H)$ (from $u_{I \cap H} < u_H$ and Lemma 6) and then $\min\{F^{n+1}(u_{I \cap H}), F(u_H), u_H\} = \min\{F^{n+1}(u_{I \cap H}), u_H\}$.
      ii. $F(l_H) < l_H \wedge u_H < F(u_H)$: The analysis of the first condition is as before, whereas the second one implies that $\min\{F^{n+1}(u_{I \cap H}), F(u_H), u_H\} = \min\{F^{n+1}(u_{I \cap H}), u_H\}$.
      iii. $l_H < F(l_H) \wedge u_H < F(u_H)$: By Lemma 5 and from $l_H < l_{I \cap H}$ we have that $l_H < F(l_H)$ implies $F(l_H) < F^{n+1}(l_{I \cap H})$. The second condition is analysed as the above case.
      iv. $l_H < F(l_H) \wedge F(u_H) < u_H$: Both cases were already treated before.

   From all the cases above, we have that $F^{n+1}(I \cap H) \cap F(H) \cap H = [\max\{F^{n+1}(l_{I \cap H}), l_H\}, \min\{F^{n+1}(u_{I \cap H}), u_H\}] = F^{n+1}(I \cap H) \cap H$.

Thus, from cases 1 and 2 we finally obtain that $\hat{\mathcal{F}}^{n+1}(I) = F^{n+1}(I \cap H) \cap H$. $\square$

Let $\mathcal{F}$ be a TAMVF and $\hat{\mathcal{F}}$ be equal to $F(I \cap H) \cap H$ with $H = S \cap J$, then the following result holds.

Lemma 15 $\hat{\mathcal{F}}^n(I) = \mathcal{F}^n(I \cap J) \cap S$.

Proof: The proof is done by induction over n.

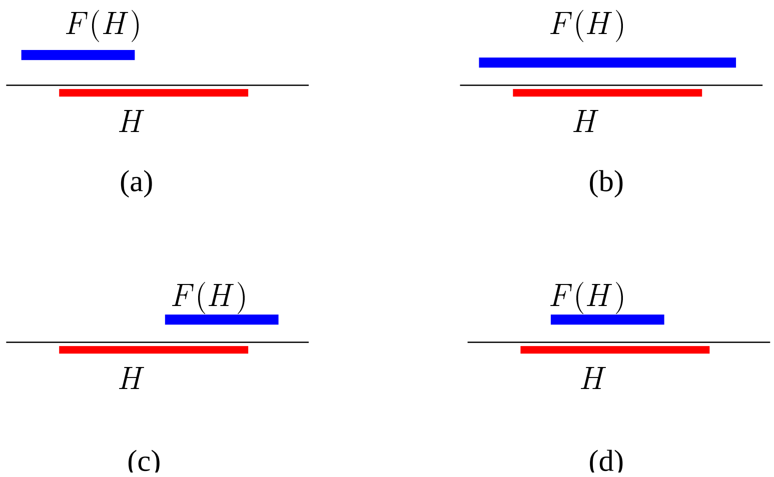


Figure 12: Relationship between $F(H)$ and H (case 2b of Lemma 14).

Base case $(n = 1)$**:**

$$\begin{aligned}
\hat{\mathcal{F}}(I) &= F(I \cap J \cap S) \cap J \cap S && \text{(by def. of } \hat{\mathcal{F}}) \\
&= (F((I \cap J) \cap S) \cap J) \cap S && \text{(by associativity)} \\
&= \mathcal{F}(I \cap J) \cap S && \text{(by def. of } \mathcal{F})
\end{aligned}$$

Inductive Step $(n > 1)$**:**

$$\begin{aligned}
\hat{\mathcal{F}}^{n+1}(I) &= \hat{\mathcal{F}}^{n}(\hat{\mathcal{F}}(I)) \\
&= F^{n}(\hat{\mathcal{F}}(I) \cap J \cap S) \cap J \cap S && \text{(by Lemma 14)} \\
&= F^{n}((F(I \cap J \cap S) \cap J \cap S) \cap J \cap S) \cap J \cap S && \text{(by def. of } \hat{\mathcal{F}}) \\
&= F^{n}(F(I \cap J \cap S) \cap J \cap S) \cap J \cap S \\
&= \hat{\mathcal{F}}^{n}(F(I \cap J \cap S)) && \text{(by Lemma 14)} \\
&= \mathcal{F}^{n}(F(I \cap J \cap S) \cap J) \cap S && \text{(by H.I.)} \\
&= \mathcal{F}^{n}(\mathcal{F}(I \cap J)) \cap S && \text{(by def. of } \mathcal{F}) \\
&= \mathcal{F}^{n+1}(I \cap J) \cap S
\end{aligned}$$

□

As a consequence we have the following

Corollary 16 $\mathcal{F}^{n+2} = \mathcal{F} \circ \hat{\mathcal{F}}^{n} \circ \mathcal{F}(I)$.

Proof:

$$\begin{aligned}
\mathcal{F}^{n+2}(I) &= \mathcal{F}(\mathcal{F}^{n}(\mathcal{F}(I))) \\
&= \mathcal{F}(\mathcal{F}^{n}(\mathcal{F}(I) \cap J)) \\
&= \mathcal{F}(\mathcal{F}^{n}(\mathcal{F}(I) \cap J) \cap S) \\
&= \mathcal{F}(\hat{\mathcal{F}}^{n}(\mathcal{F}(I))) && \text{(by Lemma 15)} \\
&= \mathcal{F} \circ \hat{\mathcal{F}}^{n} \circ \mathcal{F}(I).
\end{aligned}$$

□

5.3 Application to successors

Now we need an effective representation of (rational) points and intervals on edges. Let us introduce a one-dimensional coordinate system on each edge. For each edge e we chose a point on

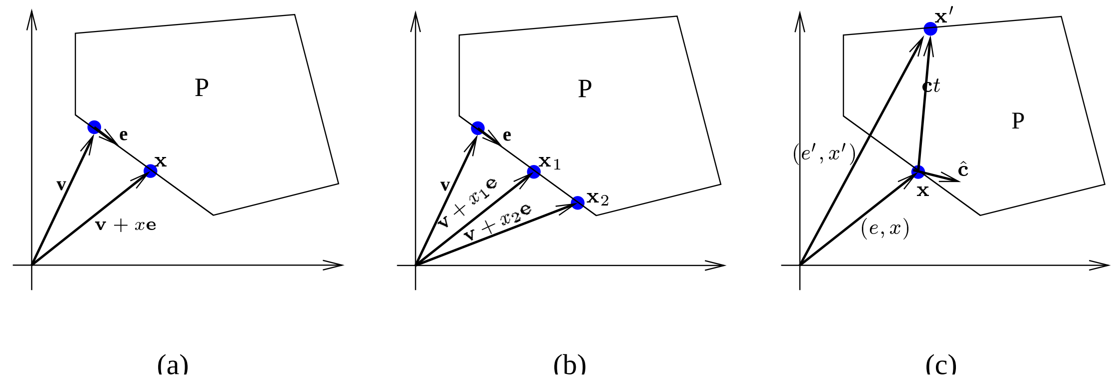


Figure 13: (a) Representation of edges; (b) Representation of an interval; (c) One-step successor.

it (the origin) with radius-vector $\mathbf{v}$, and a director vector $\mathbf{e}$ going in the positive direction in the sense of the order $\prec$.

To characterize e we need the coordinates of its extreme points: two more numbers $e^l, e^u \in \mathbb{Q} \cup \{-\infty, \infty\}$ such that $e = \{\mathbf{v} + x\mathbf{e} \mid e^l < x < e^u\}$. Clearly, having fixed $\mathbf{v}$ and $\mathbf{e}$ for every edge we can represent every point $\mathbf{x} \in e$ by a pair (e, x) identifying the edge e and the coordinate x (see Fig.13(a)). Every interval $\langle \mathbf{x}_1, \mathbf{x}_2 \rangle$ contained in e is represented as $(e, \langle x_1, x_2 \rangle)$, where $\mathbf{x}_1 = (e, x_1)$ and $\mathbf{x}_2 = (e, x_2)$ (see Fig.13(b)). Notice that if e is a vertex, then $e = \{\mathbf{v}\}$, where $\mathbf{v}$ is the only vector that characterizes e. Moreover, all the vertices have local coordinates $x \in [0, 0]$, i.e. a vertex v is represented by a pair $(v, 0)$; hence, whenever e is a vertex, $e = \langle e^l, e^u \rangle$ must be understood as $e = [e^l, e^u]$ whereas if e is a “true” edge then $e = (e^l, e^u)$.

Let $e = \langle e^l, e^u \rangle \in \mathit{in}(P)$ and $e' = \langle e'^l, e'^u \rangle \in \mathit{out}(P)^3$. For $\mathbf{x} = (e, x)$ and $\mathbf{c} \in \phi(P)$, we denote by $\mathsf{Succ}^{\mathbf{c}}_{e,e'}(x)$ the unique $\mathbf{x}' = (e', x')$ such that $\mathbf{x}' = \mathbf{x} + \mathbf{c}t$ for some $t > 0$. The point (e', x') is the successor of (e, x) in the direction $\mathbf{c}$ (see Fig.13(c)). Expanding, $\mathbf{v}' + x'\mathbf{e}' = \mathbf{v} + x\mathbf{e} + t\mathbf{c}$. Multiplying both expressions by $\hat{\mathbf{c}}$ (the right rotation of $\mathbf{c}$) we obtain that $(\mathbf{v}' + x'\mathbf{e}')\hat{\mathbf{c}} = \mathbf{v}\hat{\mathbf{c}} + x\mathbf{e}\hat{\mathbf{c}}$, i.e. $x'(\mathbf{e}'\hat{\mathbf{c}}) = x(\mathbf{e}\hat{\mathbf{c}}) + (\mathbf{v} - \mathbf{v}')\hat{\mathbf{c}}$. Thus $x' = \frac{\mathbf{e}\hat{\mathbf{c}}}{\mathbf{e}'\hat{\mathbf{c}}}x + \frac{\mathbf{v}-\mathbf{v}'}{\mathbf{e}'\hat{\mathbf{c}}}\hat{\mathbf{c}}$ and $x' \in \langle e'^l, e'^u \rangle$. Thus putting $\alpha(\mathbf{c}) = \frac{\mathbf{e}\hat{\mathbf{c}}}{\mathbf{e}'\hat{\mathbf{c}}}$, $\beta(\mathbf{c}) = \frac{\mathbf{v}-\mathbf{v}'}{\mathbf{e}'\hat{\mathbf{c}}}\hat{\mathbf{c}}$ we have that $\mathsf{Succ}^{\mathbf{c}}_{e,e'}(x) = \alpha(\mathbf{c})x + \beta(\mathbf{c})$ with $\alpha > 0$ (which follows from the region's *goodness* property). Notice that we have $x' = \mathsf{Succ}^{\mathbf{c}}_{e,e'}(x)$ iff $x \in e$, $x' \in e'$ and $x' = F(x)$. Thus, $x' = F(\{x\} \cap S) \cap J$ (with $F(x) = \alpha(\mathbf{c})x + \beta(\mathbf{c})$, $S = (\langle e^l, e^u \rangle)$ and $J = \langle e'^l, e'^u \rangle)$), i.e. $x' = \mathcal{F}_{F, \langle e^l, e^u \rangle, \langle e'^l, e'^u \rangle}$. Hence,

Lemma 17 *The function* $\mathsf{Succ}^{\mathbf{c}}_{e,e'}$ *is truncated affine.* $\square$

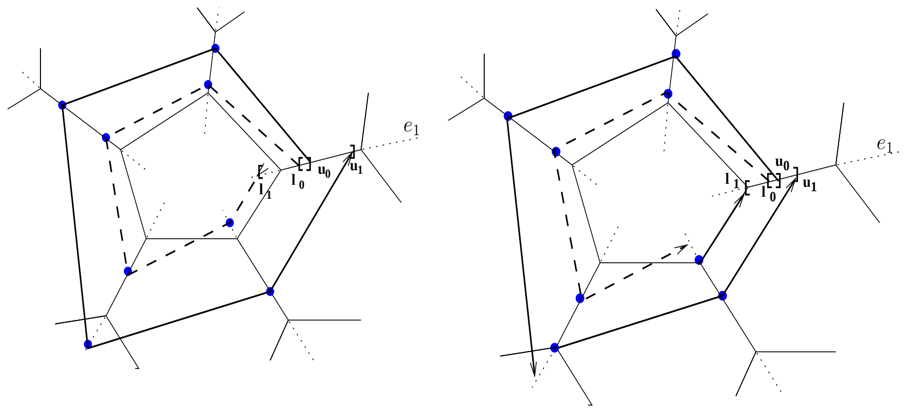
$\widetilde{\mathsf{Succ}}^{\mathbf{c}}_{e,e'}$ will denote the *non-truncated* function $\alpha(\mathbf{c})x + \beta(\mathbf{c})$. The notion of *successor* can be extended on all possible directions $\mathbf{c} \in \phi(P)$ and it can be applied to any subset $A \subseteq \langle e^l, e^u \rangle$ and in particular to intervals $\langle l, u \rangle$:

Lemma 18 *Let* $\phi(P) = \angle^{\mathbf{b}}_{\mathbf{a}}$, $\mathbf{x} = (e, x)$ *and* $\langle l, u \rangle \subseteq \langle e^l, e^u \rangle$. *Then:*

1. $\mathsf{Succ}_{e,e'}(x) = \bigcup_{\mathbf{c} \in \phi(P)} \mathsf{Succ}^{\mathbf{c}}_{e,e'}(x) = \langle \widetilde{\mathsf{Succ}}^{\mathbf{b}}_{e,e'}(x), \widetilde{\mathsf{Succ}}^{\mathbf{a}}_{e,e'}(x) \rangle \cap \langle e'^l, e'^u \rangle$;

2. $\mathsf{Succ}_{e,e'}(\langle l, u \rangle) = \langle \widetilde{\mathsf{Succ}}^{\mathbf{b}}_{e,e'}(l), \widetilde{\mathsf{Succ}}^{\mathbf{a}}_{e,e'}(u) \rangle \cap \langle e'^l, e'^u \rangle$. $\square$

Therefore, $\mathsf{Succ}_{e,e'}$ is truncated affine multivalued ($\mathsf{Succ}_{e,e'}(\langle l, u \rangle) = \langle \widetilde{\mathsf{Succ}}_{e,e'}(\langle l, u \rangle \cap \langle e^l, e^u \rangle) \cap \langle e'^l, e'^u \rangle$). This lemma shows that in order to find a successor of an interval in an edge e, we should apply the smallest dynamics ($\mathbf{a}$) to its right end and the greatest ($\mathbf{b}$) to its left end, and intersect



(a) (b)

Figure 14: (a) Non-truncated operator ($\mathsf{Succ}_{e_1}(l_0, u_0) = \langle l_1, u_1\rangle \in \langle e_1^l, e_1^u\rangle$); (b) Truncated successor ($\mathsf{Succ}_{e_1}(l_0, u_0) = \langle l_1, u_1\rangle$, with $l_1 < e_1^l < u_1 \leq e_1^u$).

the result with the target edge. Fig. 14 shows the difference between non-truncated and truncated successors.

The successor operator will be used as a building block in the reachability algorithm. It can be naturally extended on edge signatures: for $w = e_1, e_2, \ldots, e_n$ let

$$\mathsf{Succ}_w(I) = \mathsf{Succ}_{e_{n-1},e_n} \circ \ldots \circ \mathsf{Succ}_{e_2,e_3} \circ \mathsf{Succ}_{e_1,e_2}(I)$$

that by Lemma 8 is truncated affine. One important remark here is that if for one-step successor the intervals S and J of last section coincide with the source edge and the target edge respectivelly, in general is not longer the case when composing successors. The "new" S and J are computed as explained in Lemma 8.

Notice that since we use edge signatures the semi-group property takes the following form.

Lemma 19 *For any edge signatures w and v and an edge e*

$$\mathsf{Succ}_{ew} \circ \mathsf{Succ}_{ve} = \mathsf{Succ}_{vew}$$

It is convenient to define a (trivial) successor Succ_e where e is a single edge. The only way to do it preserving the semi-group property is to put $\mathsf{Succ}_e(x) = x$.

In order to manipulate successor operators we should investigate their algebraic properties. Since one-step successors $\mathsf{Succ}_{e_1e_2}$ are truncated affine, Lemma 19 guarantees that all the multi-step Succ_u are truncated affine as well. In the sequel we will apply the iteration analysis to their non-truncated versions $\widetilde{\mathsf{Succ}}_u$.

Lemma 20 *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 $f_i(x) = \widetilde{\mathsf{Succ}}^{\mathbf{c}}_{e,e_i}(x)$. If $e_2 \prec e_1$ and $l_1 < f_1(x)$ then $u_2 < f_2(x)$.*

Proof: (See Fig. 15). By definition, from $e_2 \prec e_1$, we have that $\mathbf{e}_2 u_2 + \mathbf{v}_2 \preceq \mathbf{e}_1 l_1 + \mathbf{v}_1$ and by definition of $\preceq$ between points $\hat{\mathbf{c}}(\mathbf{e}_2 u_2 + \mathbf{v}_2) \leq \hat{\mathbf{c}}(\mathbf{e}_1 l_1 + \mathbf{v}_1)$. Distributing we obtain that $\hat{\mathbf{c}}\mathbf{e}_2 u_2 + \hat{\mathbf{c}}\mathbf{v}_2 \leq \hat{\mathbf{c}}\mathbf{e}_1 l_1 + \hat{\mathbf{c}}\mathbf{v}_1$. On the other hand, from the hypothesis we have that $l_1 < \frac{\hat{\mathbf{c}}\mathbf{e}}{\hat{\mathbf{c}}\mathbf{e}_1}x + \frac{\hat{\mathbf{c}}(\mathbf{v}-\mathbf{v}_1)}{\hat{\mathbf{c}}\mathbf{e}_1}$ and doing some algebraic manipulation we obtain $\hat{\mathbf{c}}\mathbf{e}_1 l_1 + \hat{\mathbf{c}}\mathbf{v}_1 < \hat{\mathbf{c}}\mathbf{v} + \hat{\mathbf{c}}\mathbf{e}x$. From both results we obtain (by transitivity) that $\hat{\mathbf{c}}\mathbf{e}_2 u_2 + \hat{\mathbf{c}}\mathbf{v}_2 < \hat{\mathbf{c}}\mathbf{v} + \hat{\mathbf{c}}\mathbf{e}x$. By rearranging we have that $u_2 < \frac{\hat{\mathbf{c}}\mathbf{e}}{\hat{\mathbf{c}}\mathbf{e}_2}x + \frac{\hat{\mathbf{c}}(\mathbf{v}-\mathbf{v}_2)}{\hat{\mathbf{c}}\mathbf{e}_2}$, i.e. $u_2 < f_2(x)$. $\square$

As a consequence we have the following

[3] We will use both e and $\langle e^l, e^u\rangle$ to denote an edge.

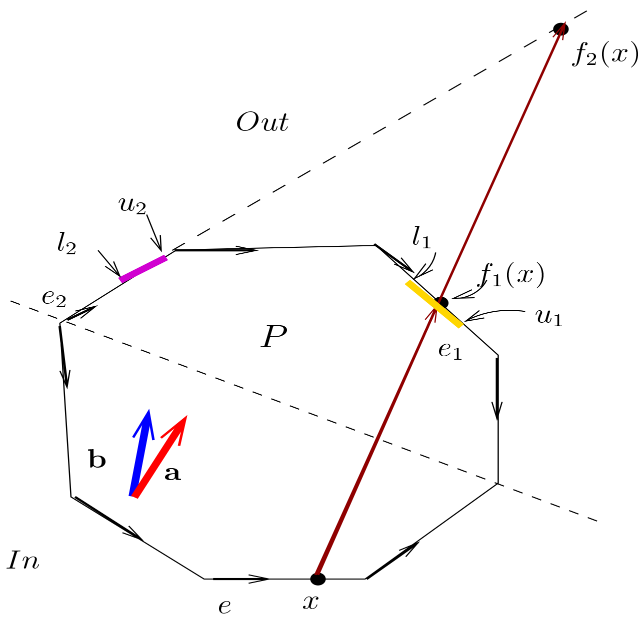


Figure 15: Statement of Lemma 20.

Corollary 21 *Let P be a region, $e \in in(P)$, $e_1, e_2 \in out(P)$, $f_i(x) = \widetilde{\mathsf{Succ}}^{\mathbf{c}}_{e,e_i}(x)$ 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 $J_i = \langle L_i, U_i\rangle$). Given that $e_2 \prec e_1$ we have that*

1. *If $L_1 < f_1^l(x)$ then $\mathcal{F}_2(\langle x, y\rangle) = \emptyset$;*
2. *If $f_2^u(y) < U_2$ then $\mathcal{F}_1(\langle x, y\rangle) = \emptyset$.*

Proof: Direct from lemma 20 and definition of $\mathcal{F}_i(\langle x, y\rangle)$. $\square$

Example. Let us come back to the example of the swimmer trying to escape from a whirlpool in a river (see Fig. 3). Suppose that the swimmer is following a trajectory with edge signature $(e_1 \ldots e_8)^*$. It is not difficult to find a representation of the edges such that for each edge e_i, $(e_i^l, e_i^u) = (0, 1)$. Besides, the (non-truncated) affine successor functions are:

$$
\begin{aligned}
&\widetilde{\mathsf{Succ}}_{e_1 e_2}(x) = \left[\tfrac{x}{2}, \tfrac{x}{2}\right]\\
&\widetilde{\mathsf{Succ}}_{e_2 e_3}(x) = \left[x - \tfrac{1}{4}, x + \tfrac{11}{60}\right]\\
&\widetilde{\mathsf{Succ}}_{e_i e_{i+1}}(x) = [x, x], \text{ for all } i \in [3, 7]\\
&\widetilde{\mathsf{Succ}}_{e_8 e_1}(x) = \left[x + \tfrac{1}{5}, x + \tfrac{1}{5}\right]
\end{aligned}
$$

The truncated affine version of the functions above (normalized) are

$$
\begin{aligned}
\mathsf{Succ}_{e_1 e_2}(x) &= \begin{cases} \left[\frac{x}{2}, \frac{x}{2}\right] \cap (0,1) & \text{if } x \in (0,1) \\ \emptyset & \text{otherwise} \end{cases}\\
\mathsf{Succ}_{e_2 e_3}(x) &= \begin{cases} \left[x - \frac{1}{4}, x + \frac{11}{60}\right] \cap (0,1) & \text{if } x \in (0,1) \\ \emptyset & \text{otherwise} \end{cases}\\
\mathsf{Succ}_{e_i e_{i+1}}(x) &= \begin{cases} [x, x] \cap (0,1) & \text{if } x \in (0,1) \\ \emptyset & \text{otherwise} \end{cases}\\
\mathsf{Succ}_{e_8 e_1}(x) &= \begin{cases} \left[x + \frac{1}{5}, x + \frac{1}{5}\right] \cap (0,1) & \text{if } x \in (0, \frac{4}{5}) \\ \emptyset & \text{otherwise} \end{cases}
\end{aligned}
$$

The successor function for the loop $s = e_1 \dots e_8$ is obtained by composition of the above functions as follows. Let us first compute

$$\mathsf{Succ}_{e_1e_2e_3}(l,u) = F(\langle l,u\rangle \cap S) \cap J$$

where

$$\begin{array}{ll} F & = \widetilde{\mathsf{Succ}}_{e_2e_3} \circ \widetilde{\mathsf{Succ}}_{e_1e_2} \\ S & = S_1 \cap \widetilde{\mathsf{Succ}}^{-1}_{e_1e_2}(J_1 \cap S_2) \\ J & = J_2 \cap \widetilde{\mathsf{Succ}}_{e_2e_3}(J_1 \cap S_2) \end{array}$$

with

$$\begin{array}{ll} J_0 & = e_1 = (0,1) \\ J_1 & = e_2 = (0,1) \\ J_2 & = e_3 = (0,1) \\ S_1 & = \widetilde{\mathsf{Succ}}^{-1}_{e_1e_2}(J_1) \cap J_0 \\ S_2 & = \widetilde{\mathsf{Succ}}^{-1}_{e_2e_3}(J_2) \cap J_1 \end{array}$$

and

$$\begin{array}{ll} \widetilde{\mathsf{Succ}}^{-1}_{e_1e_2}(l,u) & = [2l, 2u] \\ \widetilde{\mathsf{Succ}}^{-1}_{e_2e_3}(l,u) & = [l - \frac{11}{60}, u + \frac{1}{4}] \end{array}$$

We compute now all the parameters above in order to obtain F, S and J

$$\begin{array}{ll} S_1 & = \widetilde{\mathsf{Succ}}^{-1}_{e_1e_2}((0,1)) \cap (0,1) = (0,2) \cap (0,1) = (0,1) \\ S_2 & = \widetilde{\mathsf{Succ}}^{-1}_{e_2e_3}((0,1)) \cap (0,1) = (-\frac{11}{60}, \frac{5}{4}) \cap (0,1) = (0,1) \\ & \\ F(l,u) & = [\frac{l}{2} - \frac{1}{4}, \frac{u}{2} + \frac{11}{60}] \\ S & = (0,1) \cap \widetilde{\mathsf{Succ}}^{-1}_{e_1e_2}((0,1) \cap (0,1)) = (0,1) \cap (0,2) = (0,1) \\ J & = (0,1) \cap \widetilde{\mathsf{Succ}}_{e_2e_3}((0,1) \cap (0,1)) = (0,1) \cap (-\frac{1}{4}, \frac{71}{60}) = (0,1) \end{array}$$

We have then that

$$\mathsf{Succ}_{e_1e_2e_3}(l,u) = \begin{cases} [\frac{l}{2} - \frac{1}{4}, \frac{u}{2} + \frac{11}{60}] \cap (0,1) & \text{if } \langle l,u\rangle \subseteq (0,1) \\ \emptyset & \text{otherwise} \end{cases}$$

Since $\widetilde{\mathsf{Succ}}_{e_ie_{i+1}}$ for $i \in [3,7]$ are the identity functions, we have that

$$\mathsf{Succ}_{e_3\dots e_8}(l,u) = \begin{cases} \langle l,u\rangle \cap (0,1) & \text{if } \langle l,u\rangle \subseteq (0,1) \\ \emptyset & \text{otherwise} \end{cases}$$

and composing the functions above we obtain that $\mathsf{Succ}_{e_1\dots e_8} = \mathsf{Succ}_{e_1e_2e_3}$. We compute now

$$\mathsf{Succ}_{e_1\dots e_8e_1}(l,u) = F'(\langle l,u\rangle \cap S') \cap J'$$

where

$$\begin{array}{ll} F' & = \widetilde{\mathsf{Succ}}_{e_8e_1} \circ \widetilde{\mathsf{Succ}}_{e_1\dots e_8} \\ S' & = S_1 \cap \widetilde{\mathsf{Succ}}^{-1}_{e_1\dots e_8}(J_1 \cap S_2) \\ J' & = J_2 \cap \widetilde{\mathsf{Succ}}_{e_8e_1}(J_1 \cap S_2) \end{array}$$

with

$$\begin{array}{ll} J_0 & = e_1 = (0,1) \\ J_1 & = J = (0,1) \\ J_2 & = e_1 = (0,1) \\ S_1 & = \widetilde{\mathsf{Succ}}^{-1}_{e_1\ldots e_8}(J_1) \cap J_0 \\ S_2 & = \widetilde{\mathsf{Succ}}^{-1}_{e_8 e_1}(J_2) \cap J_1 \end{array}$$

and

$$\begin{array}{ll} \widetilde{\mathsf{Succ}}^{-1}_{e_1\ldots e_8}(l,u) & = [2l - \frac{11}{30}, 2u + \frac{1}{2}] \\ \widetilde{\mathsf{Succ}}^{-1}_{e_8 e_1}(l,u) & = [l - \frac{1}{5}, u - \frac{1}{5}] \end{array}$$

We compute the parameters above to obtain F', S' and J':

$$\begin{array}{ll} S_1 & = \widetilde{\mathsf{Succ}}^{-1}_{e_1\ldots e_8}((0,1)) \cap (0,1) = (-\frac{11}{30}, \frac{5}{2}) \cap (0,1) = (0,1) \\ S_2 & = \widetilde{\mathsf{Succ}}^{-1}_{e_8 e_1}((0,1)) \cap (0,1) = (-\frac{1}{5}, \frac{4}{5}) \cap (0,1) = (0, \frac{4}{5}) \end{array}$$

$$\begin{array}{ll} F'(l,u) & = [\frac{l}{2} - \frac{1}{20}, \frac{u}{2} + \frac{23}{60}] \\ S' & = (0,1) \cap \widetilde{\mathsf{Succ}}^{-1}_{e_1\ldots e_8}((0,1) \cap (0, \frac{4}{5})) = (0,1) \cap (-\frac{11}{30}, \frac{21}{10}) = (0,1) \\ J' & = (0,1) \cap \widetilde{\mathsf{Succ}}_{e_8 e_1}((0,1) \cap (0, \frac{4}{5})) = (0,1) \cap (\frac{1}{5}, 1) = (\frac{1}{5}, 1) \end{array}$$

Hence,

$$\mathsf{Succ}_{e_1\ldots e_8 e_1}(l,u) = \begin{cases} [\frac{l}{2} - \frac{1}{20}, \frac{u}{2} + \frac{23}{60}] \cap (\frac{1}{5}, 1) & \text{if } \langle l,u \rangle \subseteq (0,1) \\ \emptyset & \text{otherwise} \end{cases}$$

Finally, by Lemma 10 we obtain the limits: $l^* = \frac{-\frac{1}{20}}{1-\frac{1}{2}} = -\frac{1}{10}$, and $u^* = \frac{\frac{23}{60}}{1-\frac{1}{2}} = \frac{23}{30}$. $\square$

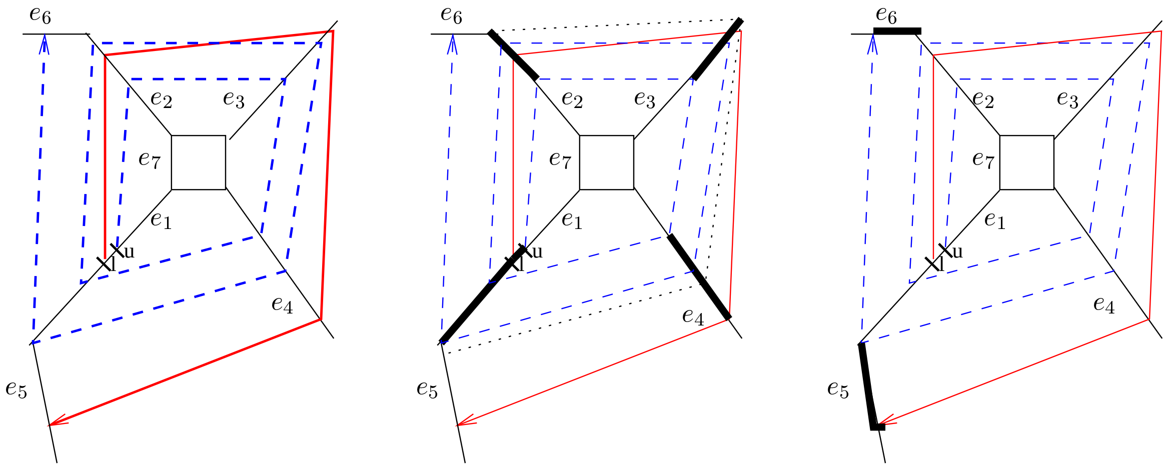
6 Reachability analysis

6.1 Main algorithm

We are interested in the reachability analysis between two points $\mathbf{x}_0 = (e_0, x_0)$ and $\mathbf{x}_f = (e_f, x_f)$ and from the section above we know that there exists a finite number of type signatures of the form $r_1, s_1, \ldots, r_n, s_n, r_{n+1}$. Moreover, the type signatures are restricted to those with $e_0 = first(r_1)$ and $e_f \in r_{n+1}$. Given such a set of type signatures $type(e_0, e_f)$, the following algorithm is guaranteed to terminate, answering YES if $\mathbf{x}_f$ is reachable from $\mathbf{x}_0$ or NO otherwise:

function $Reach(\mathbf{x}_0, \mathbf{x}_f)$
    $R = \mathsf{false}$
    for each $w \in type(e_0, e_f)$
        $R = R \vee Reach_{type}(x_0, x_f, w)$
    $\longleftarrow R$

Reachability from $\mathbf{x}_0$ to $\mathbf{x}_f$ with fixed signature w is tested by the function $Reach_{type}(x_0, x_f, w)$. Let the type w have the form $w = r_1, s_1, \ldots, r_n, s_n, r_{n+1}$. Put $f_i = first(s_i)$ and $ex_i = first(r_{i+1})$ if r_{i+1} is non-empty and f_{i+1} otherwise (i.e. ex_i is the edge to which the trajectory exits from the loop s_i). Let us say that a type signature w has a $\mathsf{loop_{end}}$ property if $first(r_{n+1}) = first(s_n)$, i.e. signatures of type w terminate by several repetitions of the last loop.



(a) (b) (c)

Figure 16: [DIE] (a) Both trajectories leave the cycle $(e_1, e_2, e_3, e_4)^*$ through the left; (b) Reachable points on the cycle (in bold); (c) Possible continuation after leaving the cycle (in bold).

> **function** $Reach_{type}(x_0, x_f, w)$:
> $Z = \mathsf{Succ}_{r_1 f_1}(x_0)$
>   **for** $i = 1$ *to* $n-1$
> $Z = \mathsf{Succ}_{r_{i+1} f_{i+1}}(Exit(Z, s_i, ex_i))$
>   **if** $\mathsf{loop}_{\mathsf{end}}(w)$
>   **then** $\longleftarrow Test(Z, s_n, x_f)$
>   **else** $\longleftarrow x_f \in \mathsf{Succ}_{r_{n+1}}(Exit(Z, s_n, ex_n))$?

This algorithm uses two functions: $Test(Z, s, x)$ that answers whether x is reachable from a set Z (represented as a finite union of intervals) in the loop s (formally, whether $x \in \mathsf{Succ}_{s^+ first(s)}(I)$); and the function $Exit(Z, s, e)$ that for an initial set S, a loop s, and an edge e (not in this loop) finds all the points on e reachable by making s several times and then exiting to e (formally, it computes $\mathsf{Succ}_{s^+ e}(I)$, which is always a finite union of intervals).

Since we know how to calculate the successor of a given interval in one and in several steps ($\mathsf{Succ}_{ee'}(\cdot)$ and $\mathsf{Succ}_r(\cdot)$), in order to implement $Test(\cdot)$ and $Exit(\cdot)$ it remains to show how to analyze the (simple) cycles s_i and eventually its continuation.

Both algorithms $Test(\cdot)$ and $Exit(\cdot)$ start by qualitative analysis of the cycle. This analysis proceeds as follows.

Let s be a simple cycle, $f = first(s)$ its first edge, and $I = \langle l, u \rangle \subset f$ an initial interval and $\mathsf{Succ}_{s,f}(x) = \widetilde{\mathsf{Succ}}_{s,f}(\{x\} \cap S) \cap J$. Notice that the above formula is correct (by def. of truncated successor) but if we want to iterate the successor (apply it again) we must guarantee that we can really do it. Thus, the result should be in S and hence we obtain that to be able to iterate, $\mathsf{Succ}_{s,f}(x)$ must be included on $S \cap J$, that is then the precondition for cycling. In what follows $\langle L, U \rangle$ will denote $S \cap J$.

The first thing to do is to determine the qualitative behavior of the leftmost and rightmost trajectories of the interval endpoints in the cycle. This can be done without iterating Succ_{sf}. Indeed, by Lemma 10, we can compute the limits $(l_1^*, u_1^*) = \lim_{n\to\infty} \widetilde{\mathsf{Succ}}^{\,n}_{s,f}(\langle l, u \rangle)$ (notice that those are limits only for the *non-truncated operator* $\widetilde{\mathsf{Succ}}$), not taking into account that the edges are possible bounded (we use Lemma 14) and compare these limit points corresponding to unrestricted dynamics with L and U. There are five possibilities:

1. STAY The cycle is not abandoned by any of the two trajectories: $L \leq l^* \leq u^* \leq U$;

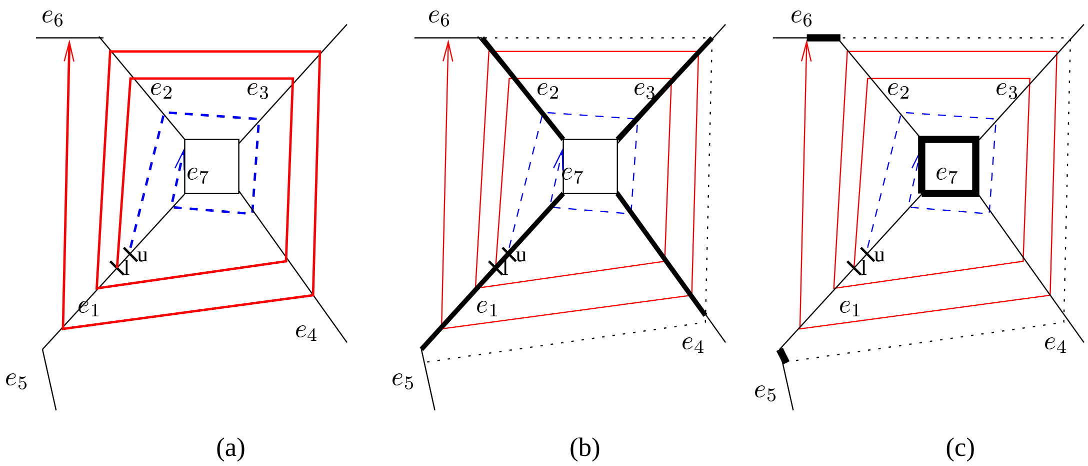


Figure 17: [EXIT-BOTH]] (a) Both trajectories leave the cycle $(e_1, e_2, e_3, e_4)^*$; (b) Reachable points on the cycle (in bold); (c) Possible continuation after leaving the cycle (in bold).

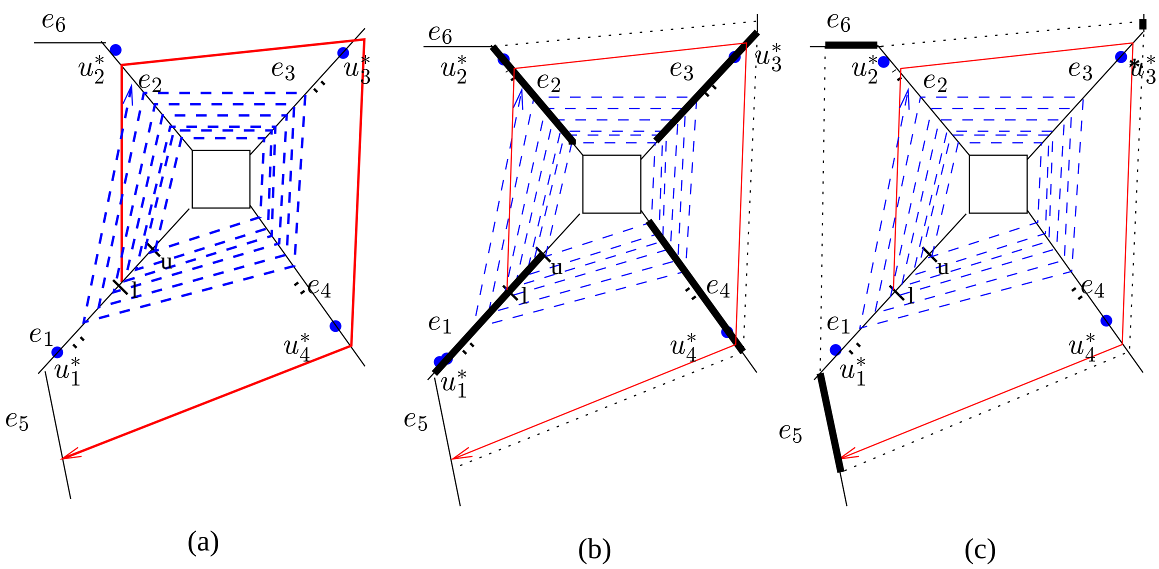


Figure 18: [EXIT-LEFT] (a) The left trajectory leave the cycle $(e_1, e_2, e_3, e_4)^*$ through the left, whereas the right one tends to the limit u^*; (b) Reachable points on the cycle (in bold); (c) Possible continuation after leaving the cycle (in bold).

2. DIE The right trajectory exits the cycle through the left (consequently the left one also exits) or the left trajectory exits the cycle through the right (consequently the right one also exits). In symbols, $u^* < L \vee l^* > U$, see Fig. 16;

3. EXIT-BOTH Both trajectories exit the cycle (the left one through the left and the right one through the right): $l^* < L \wedge u^* > U$, see Fig.17;

4. EXIT-LEFT The leftmost trajectory exits the cycle but not the other: $l^* < L \leq u^* \leq U$, see Fig.18.

5. EXIT-RIGHT The rightmost trajectory exits the cycle but not the other: $L \leq l^* \leq U < u^*$.

This qualitative analysis is implemented in the function $Analyze(I, s)$ which returns the kind of qualitative behavior of the interval $I = \langle l, u \rangle$ under the loop s.

function $Analyze(I, s)$
  cases
    $L \leq l^* \leq u^* \leq U$: $\longleftarrow$ STAY
    $u^* < L \vee l^* > U$: $\longleftarrow$ DIE
    $l^* < L \wedge u^* > U$: $\longleftarrow$ EXIT-BOTH
    $L \leq l^* \leq U < u^*$: $\longleftarrow$ EXIT-RIGHT
    $l^* < L \leq u^* \leq U$: $\longleftarrow$ EXIT-LEFT
  endcases

Notice that one (of both) of the successor functions can be the identity. In this case we have an infinite number of fixpoints but the analysis above continue to apply.

6.2 Exit

In this section we describe the EXIT algorithm and show its soundness and termination. The exit set, for a given initial interval I, is $Ex = \cup_{m>0}\mathsf{Succ}_{s,ex} \circ \mathsf{Succ}^m_{s,f}(I)$. The function $Exit(Z, s, ex)$ should return $\mathsf{Succ}_{s+ex}(Z)$. Both the argument Z and the result are finite collections of intervals. The exploration is made for each initial interval separately.

function $Exit(Z, s, ex)$
  $E = \emptyset$
  for each $I \in Z$
    if $\mathsf{Succ}_{s,f}(I) \cap S \neq \emptyset$
    then $E = E \cup Exit_{Analyze}(\mathsf{Succ}_{s,f}(I) \cap S, s, ex)$
    else $E = E \cup \mathsf{Succ}_{s,ex}(\mathsf{Succ}_{s,f}(I))$
  $\longleftarrow$ E

Notice that the call $\mathsf{Succ}_{s,f}(I)$ ensures that $I \subseteq \langle L, U \rangle$. All the work for each initial interval I is done by the function $Exit_{Analyze}(I, s, ex)$ which launches the $Analyze(\cdot)$ procedure described above and last, according to the result of this analysis launches one of five specialized procedures $Exit_{STAY}$, $Exit_{LEFT}$, $Exit_{RIGHT}$, $Exit_{BOTH}$, $Exit_{DIE}$ which calculates the exit set.

We consider separately all these five algorithms and state their termination and soundness. This will imply termination and soundness of the $Exit$ function itself.

Notation. We recall the notations introduced before and we introduce others to simplify the proofs. As before, let s be a simple cycle, $f = first(s)$ its first edge and $I = \langle l, u \rangle \subset f$ be the initial interval. Notice that the functions $Exit_*$ are always called with $I \subseteq \langle L, U \rangle$ (in fact this is the precondition for iterating, see Lemma 14). Let $I_i = \langle l_i, u_i \rangle = \mathsf{Succ}^i_{s,f}(I)$ and $\tilde{I}_i = \langle \tilde{l}_i, \tilde{u}_i \rangle = \widetilde{\mathsf{Succ}}^i_{s,f}(I)$. The Fundamental Lemma (Lemma 14) guarantees that $I_i = \tilde{I}_i \cap \langle L, U \rangle$. Remember that $\mathcal{F}(I) = \mathsf{Succ}_{s,f}(I) = \widetilde{\mathsf{Succ}}_{s,f}(I \cap S) \cap J$ and $\hat{\mathcal{F}}(I) = \widetilde{\mathsf{Succ}}_{s,f}(I \cap S \cap J) \cap S \cap J$.

6.2.1 Exit-STAY

$$\begin{array}{|l|}\hline \textbf{function} \;\; Exit_{STAY}(I,s,ex) \\ \qquad\qquad\quad \longleftarrow \emptyset \\ \hline \end{array}$$

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 21 we have that $\mathsf{Succ}^i_{s,ex}(I) = \emptyset$.

Termination Trivial. $\Box$

6.2.2 Exit-DIE

$$\begin{array}{|l|}\hline \textbf{function} \;\; Exit_{DIE}(I,s,ex) \\ \qquad\qquad Z = \emptyset \\ \qquad\qquad \textbf{repeat} \\ \qquad\qquad\qquad I = \mathsf{Succ}_{s,f}(I) \\ \qquad\qquad\qquad Z = Z \bigcup \mathsf{Succ}_{s,ex}(I) \\ \qquad\qquad \textbf{until } I = \emptyset \\ \qquad\qquad \longleftarrow \mathrm{Z} \\ \hline \end{array}$$

Soundness Trivial.

Termination From the hypothesis we know that there exists an n s.t. $\tilde{I}_n \cap \langle L, U \rangle = \emptyset$ (either because $\tilde{u}_n < L$ if $u^* < L$ or because $U < \tilde{l}_n$ if $U < l^*$). Both cases imply that $\mathsf{Succ}^n_{s,f}(I) = \emptyset$. $\Box$

6.2.3 Exit-BOTH

$$\begin{array}{|l|}\hline \textbf{function} \;\; Exit_{BOTH}(I,s,ex) \\ \qquad\qquad \longleftarrow \mathsf{Succ}_{s,ex}(\mathsf{Succ}_{s,f}(\langle L, U \rangle)) \\ \hline \end{array}$$

Soundness Notice that we call $Exit_{BOTH}$ with $\mathsf{Succ}_{s,f}(I) \cap S = \mathcal{F}(I)$. On the other hand, because the limits are out of $\langle L, U \rangle$, we know that there exists an n such that $\langle L, U \rangle \subset \tilde{I}_n$ and by the Fundamental Lemma (Lemma 14), $\hat{\mathcal{F}}^n(I) = I_n = \langle L, U \rangle$ (i.e. $\hat{\mathcal{F}}^n \circ \mathcal{F}(I) = \langle L, U \rangle$). By Corollary 16 we have that $\mathcal{F}^n(I) = \mathcal{F} \circ \hat{\mathcal{F}}^{n-2} \circ \mathcal{F}(I) = \mathcal{F}(\langle L, U \rangle) = \mathsf{Succ}_{s,f}(\langle L, U \rangle)$.

1. We prove first that the algorithm produces just 'exits' (i.e., $\mathsf{Succ}_{s,ex}(\mathsf{Succ}_{s,f}(\langle L, U \rangle)) \subseteq Ex$). This follows directly from the fact that $\mathsf{Succ}_{s,f}(\langle L, U \rangle) = \mathsf{Succ}^n_{s,f}(I) \subseteq \cup_{m>0} \mathsf{Succ}^m_{s,f}(I)$;
2. We prove now that all the 'exits' are computed ($Ex \subseteq \mathsf{Succ}_{s,ex}(\mathsf{Succ}_{s,f}(\langle L, U \rangle))$). By definition, $Ex = \cup_{m>0} \mathsf{Succ}_{s,ex} \circ \mathcal{F}^m(I)$, that can be written as $Ex = \mathsf{Succ}_{s,ex} \circ \mathcal{F}(I) \cup \mathsf{Succ}_{s,ex} \circ \mathcal{F} \circ \mathcal{F}(\cup_{m \geq 2} \mathcal{F}^{m-2}(I))$. Let A be the set $\cup_{m \geq 2} \mathcal{F}^{m-2}(I)$, thus $\mathcal{F} \circ \mathcal{F}(A) = \mathcal{F}(S \cap \mathcal{F}(A)) \subseteq \mathcal{F}(S \cap J) = \mathcal{F}(\langle L, U \rangle)$. On the other hand, $\mathsf{Succ}_{s,ex} \circ \mathcal{F}(I) \subseteq \mathsf{Succ}_{s,ex} \circ \mathcal{F}(\langle L, U \rangle)$, since $I \subseteq \langle L, U \rangle$ and by monotonicity of both functions. Hence, $Ex \subseteq \mathsf{Succ}_{s,ex} \circ \mathcal{F}(\langle L, U \rangle)$.

Termination Trivial. $\Box$

6.2.4 Exit-LEFT

$$\begin{array}{|l|}\hline \textbf{function} \;\; Exit_{LEFT}(I,s,ex) \\ \qquad\qquad \longleftarrow \mathsf{Succ}_{s,ex}(\mathsf{Succ}_{s,f}(\langle L, \max\{u, u^*\} \rangle)) \\ \hline \end{array}$$

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 $f \prec ex$ then $Ex = \emptyset$ (by definition of Exit-LEFT) and $\mathsf{Succ}_{s,ex}(I_i) = \emptyset$ for any i (by Corollary 21-2), so $\mathsf{Succ}_{s,ex}(\mathsf{Succ}_{s,f}(\langle L, \max\{u, u^*\}\rangle)) = \emptyset$;
2. If $ex \prec f$, we consider two cases:
   (a) If $u < u^*$ then for all i, $u_i = \tilde{u}_i \leq u^*$ and then $\cup_{m>0}\mathsf{Succ}^m_{s,f}(I) = \mathsf{Succ}_{s,f}(L, u^*)$, 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. □

In fact, it can be shown that $Exit_{LEFT} = \mathsf{Succ}_{s,ex}(\mathsf{Succ}_{s,f}(\langle L, U\rangle))$.

6.2.5 Exit-RIGHT

Similar to the previous case.

6.3 Test

In this section we describe the Test function and show its soundness and termination. In what follows, $l\uparrow$ means that the sequence $l, l_1, l_2, \ldots$ of successive successors of l is increasing whereas $l\downarrow$ means that the sequence is decreasing. Similarly for $u\uparrow$ and $u\downarrow$. Notice that detecting whether the sequences l_n and u_n are increasing or decreasing can be easily done at the stage of the preliminary analysis of the loop.

The upper-level structure is the same as for EXIT: each initial interval is treated separately by $Test_{Analyze}$, which makes one turn of the loop, calls *Analyze* and delegates all the remaining to one of the five specialized functions $Test_{STAY}$, $Test_{LEFT}$, $Test_{RIGHT}$, $Test_{BOTH}$, $Test_{DIE}$.

```
function  Test(Z, s, x)
          R = false
          for each  I ∈ Z such that Succ_{s,f}(I) ∩ S ≠ ∅
                    R = R ∨ Test_{Analyze}(Succ_{s,f}(I), s, x)
          ⟵ R
```

The five specialized *Test* functions use the following two procedures: The function $Found(I, x)$ determines, if the current interval I contains x (YES), does not contain x and moves in the opposite direction (NO), or none of both these cases (NOTYET). The function $Search(I, x)$ iterates the loop s until the previous function *Found* gives a definite answer YES or NO. Special measures will be taken to guarantee termination.

```
function  Found(I, x)
          cases
                x ∈ I :          ⟵ YES
                I = ∅ :          ⟵ NO
                x < I ∧ l↑ :     ⟵ NO
                x > I ∧ u↓ :     ⟵ NO
                else :           ⟵ NOTYET
          endcases
```

```
function  Search(I, x)
          while Found(I, x) = NOTYET
                I = Succ_{s,f}(I)
          ⟵ Found(I, x)
```

6.3.1 Test-STAY

$$\begin{array}{|l|}\hline \textbf{function } \ Test_{STAY}(I,s,x) \\ \qquad \textbf{cases} \\ \qquad\qquad l^* < x < u^* : \longleftarrow \text{YES} \\ \qquad\qquad x \leq l^* \wedge l \downarrow : \longleftarrow \text{NO} \\ \qquad\qquad x \geq u^* \wedge u \uparrow : \longleftarrow \text{NO} \\ \qquad\qquad \textbf{else} : \qquad \longleftarrow Search(I,x) \\ \qquad \textbf{endcases} \\ \hline \end{array}$$

Soundness We prove the soundness considering each case separately:

1. We have to prove that if $l^* < x < u^*$ then $x \in Reach(I)$. By hypothesis $l^* < x_f < u^*$, then there exists a positive real number ϵ such that $l^* + \epsilon < x_f < u^* - \epsilon$. It's not difficult to see that exists two real numbers N_1 and N_2 such that for all n greater (or equal) than N_1, $u_n > u^* - \epsilon$ and for all n greater (or equal) than N_2, $l_n < l^* + \epsilon$. Let N be equal to the maximum between N_1 and N_2, then it follows that $l_N < l^* + \epsilon$ and $u^* - \epsilon < u_N$. Thus, $l_N < x_f < u_N$ and x_f is reachable.
2. We have to prove that if $x \leq l^* \wedge l \downarrow$ then $x \notin Reach(I)$. Trivial, by definition of limit and monotonicity of the sequence.
3. We have to prove that if $u^* \leq x \wedge u \uparrow$ then $x \notin Reach(I)$. Trivial, by definition of limit and monotonicity of the sequence.
4. We have to prove that if $(x < l^* \wedge l \uparrow) \ \vee \ (u^* < x \wedge u \downarrow)$ then $Search(I,x) \equiv (x \in Reach(I)?)$. Computing $Search(I,x)$ gives a sequence of intervals $I, I_1, \ldots, I_n$ s.t. $Reach(I) = \bigcup_i I_i$. If $Search(I,x)$ terminates then $\exists i \cdot (Found(I_i,x) = \text{YES} \vee Found(I_i,x) = \text{NO})$ and $\forall j < i \cdot Found(I_i,x) = \text{NOTYET}$. We analyze then each of the cases of $Found(I,x)$:
   (a) If $x \in I$ then $Found(I_i,x) = \text{YES}$ and $x \in I_i$, i.e. $x \in Reach(I)$.
   (b) If $I = \emptyset$ then $Found(I_i,x) = \text{NO}$ and $\forall k \geq i \cdot I_k = \emptyset$ and $x \notin I_j$. Thus $x \notin Reach(I)$.
   (c) If $x < I \wedge l \uparrow$ then $Found(I_i,x) = \text{NO}$ and $\forall k \geq i \cdot x < l_i < l_k$ and because $x \notin I_j$ then $x \notin I_k$ and hence $x \notin Reach(I)$.
   (d) If $I < x \wedge u \downarrow$ then $Found(I_i,x) = \text{NO}$ and $\forall k \geq i \cdot u_k < u_i < x$ and because $x \notin I_j$ then $x \notin I_k$ and hence $x \notin Reach(I)$.

Termination We have to show termination just when $(x < l^* \wedge l \uparrow) \ \vee \ (u^* < x \wedge u \downarrow)$. If $x < l^* \wedge l \uparrow$ then $\exists i \cdot (x < l_i < l^* \ \wedge \ Found(I_i,x) = NO)$. Thus, it terminates. Similarly for the other case. $\square$

6.3.2 Test-DIE

$$\begin{array}{|l|}\hline \textbf{function } \ Test_{DIE}(I,s,x) \\ \qquad\qquad \longleftarrow Search(I,x) \\ \\ \hline \end{array}$$

Soundness Trivial.

Termination Eventually I becomes empty. Hence, at this stage $Found(I,x) = \text{NO}$ and $Search$ terminates. $\square$

6.3.3 Test-BOTH

$$\begin{array}{|l|}\hline \textbf{function } \ Test_{BOTH}(I,s,x) \\ \qquad\qquad \longleftarrow x \in \mathsf{Succ}_{s,f}(\langle L,U \rangle)? \\ \\ \hline \end{array}$$

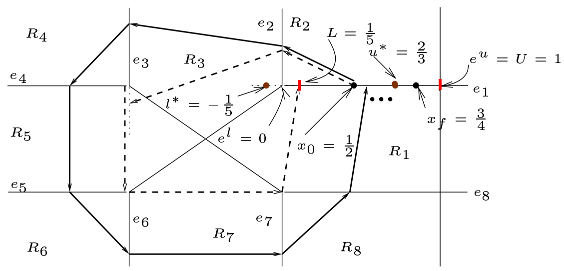


Figure 19: Example: $\mathbf{x}_f = (e_1, \frac{3}{4})$ is reachable from $\mathbf{x}_0 = (e_1, \frac{1}{2})$ ($\frac{3}{4} \in \mathsf{Succ}_{e_1 e_8 \cdots e_1}(L, u^*)$).

Soundness Immediate from the proof of soundness of the Exit algorithm for EXIT-BOTH.

Termination Trivial. $\Box$

6.3.4 Test-LEFT

function $Test_{LEFT}(I, s, x)$
  cases
    $x \in \mathsf{Succ}_{s,f}(\langle L, u^* \rangle)$: $\longleftarrow$ YES
    $x < \mathsf{Succ}_{s,f}(\langle L, u^* \rangle)$: $\longleftarrow$ NO
    $\mathsf{Succ}_{s,f}(\langle L, u^* \rangle) < x \wedge u \uparrow$: $\longleftarrow$ NO
    else : $\longleftarrow$ $Search(I, x)$
  endcases

Soundness The proof is similar to the STAY case.

Termination We have to consider just the case when $u \downarrow$ and $\mathsf{Succ}_{s,f}(\langle L, u^* \rangle) < x$. In this case we know that $\exists i \cdot u^* < u_i < x \ \wedge \ Found(I_i, x) = \text{NO}$. Thus the algorithm terminates. $\Box$

6.3.5 Test-RIGHT

The algorithm and its correctness proof are similar to the previous case. $\Box$

6.4 Main results

From all the results above we have the following theorem.

Theorem 22 (Point–to–Point Reachability) *The problem* **Reach**$(\mathcal{H}, \mathbf{x}_0, \mathbf{x}_f)$ *is decidable.* $\Box$

Example. Consider again the swimmer. Let $\mathbf{x}_0 = (e_1, \frac{1}{2})$ be her initial position. We want to decide whether she is able to escape from the whirlpool and reach the final position $\mathbf{x}_f = (e_1, \frac{3}{4})$. From the computations before done we have that $(L, U) = S \cap J = (\frac{1}{5}, 1)$. Recall that $l^* = \frac{-\frac{1}{20}}{1-\frac{1}{2}} = -\frac{1}{10}$ and $u^* = \frac{\frac{23}{60}}{1-\frac{1}{2}} = \frac{23}{30}$. Thus, by the *Analyze* function we know that the cycle behaves as an Exit-LEFT and applying the function $Test_{LEFT}$ we obtain that $\mathbf{x}_f = (e_1, \frac{3}{4})$ is reachable from $\mathbf{x}_0 = (e_1, \frac{1}{2})$ because we have that $\mathsf{Succ}_{e_1 e_8 \cdots e_1}((L, u^*)) = \mathsf{Succ}_{e_1 e_8 \cdots e_1}((\frac{1}{5}, \frac{23}{30})) = (\frac{1}{20}, \frac{23}{30})$ and $\frac{3}{4} \in (\frac{1}{20}, \frac{23}{30})$ (Fig. 19). $\Box$

Example. Let's change the above example in order to show another behavior. For simplicity we consider the same partition of the swimmer but with the following differential inclusion dynamics:

- $R_1 : \mathbf{a} = (1, \frac{10}{3}), \mathbf{b} = (1, 5)$ ($\mathbf{a} \simeq 73°18'$ and $\mathbf{b} \simeq 78°41'$);
- $R_2 : \mathbf{a} = \mathbf{b} = (-1, 1)$ ($\mathbf{a} = \mathbf{b} = 135°$);
- $R_3 : \mathbf{a} = \mathbf{b} = (-1, 0)$ ($\mathbf{a} = \mathbf{b} = 180°$);
- $R_4 : \mathbf{a} = \mathbf{b} = (-1, -1)$ ($\mathbf{a} = \mathbf{b} = 225°$);
- $R_5 : \mathbf{a} = \mathbf{b} = (0, -1)$ ($\mathbf{a} = \mathbf{b} = 270°$);
- $R_6 : \mathbf{a} = \mathbf{b} = (1, -1)$ ($\mathbf{a} = \mathbf{b} = 315°$);
- $R_7 : \mathbf{a} = \mathbf{b} = (1, 0)$ ($\mathbf{a} = \mathbf{b} = 0°$);
- $R_8 : \mathbf{a} = \mathbf{b} = (1, 1)$ ($\mathbf{a} = \mathbf{b} = 45°$).

We are interested in the edge signature $e_0(e_1 \dots e_8)^* e_9$, and what matters for computing the reachable points of e_9 starting from $x_0 \in e_0$ are the following edge-to-edge successor functions:

$$\begin{aligned}
\mathsf{Succ}_{e_0 e_1}(x) &= \begin{cases} \left[\frac{1}{5}x, \frac{3}{10}x\right] \cap (0,1) & \text{if } x \in (0,1) \\ \emptyset & \text{otherwise} \end{cases} \\
\mathsf{Succ}_{e_i e_{i+1}}(x) &= \begin{cases} [x, x] \cap (0,1) & \text{if } x \in (0,1) \\ \emptyset & \text{otherwise} \end{cases} \\
\mathsf{Succ}_{e_8 e_1}(x) &= \begin{cases} \left[x + \frac{1}{5}, x + \frac{3}{10}\right] \cap (0,1) & \text{if } x \in (0, \frac{4}{5}) \\ \emptyset & \text{otherwise} \end{cases} \\
\mathsf{Succ}_{e_8 e_9}(x) &= \begin{cases} \left[5x - 4, \frac{10}{3}x - \frac{7}{3}\right] \cap (0,1) & \text{if } x \in (\frac{7}{10}, 1) \\ \emptyset & \text{otherwise} \end{cases}
\end{aligned}$$

Let x_0 be equal to $\frac{1}{2}$ on edge e_0 and x_f be $\frac{3}{10}$ on e_9; deciding whether exists a trajectory from $(e_0, \frac{1}{2})$ to $(e_9, \frac{3}{10})$ can be done following the steps:

1. Compute the "enter interval" to the loop: $\mathsf{Succ}_{e_0 e_1}(\frac{1}{2}) = [\frac{1}{10}, \frac{3}{20}]$;

2. Compute the successor function of the loop $(e_1 \dots e_8)^{*}$[4]:

$$\mathsf{Succ}_{e_1 \dots e_8 e_1}[l, u] = \begin{cases} \left[l + \frac{1}{5}, u + \frac{3}{10}\right] \cap (\frac{1}{5}, 1) & \text{if } [l, u] \subseteq (0, \frac{4}{5}) \\ \emptyset & \text{otherwise} \end{cases}$$

3. Compute the limits of the loop signature: By Lemma 10 we have that $u^* = l^* = \infty$ for both affine functions. We can then conclude that the trajectories will be counterclockwise expanding spirals and the *Analyze* function gives that the loop will behave as a DIE (see Section 6.1);

4. Execute the function $Exit_{DIE}([\frac{1}{10}, \frac{3}{20}], e_1 \dots e_8, e_9) = \{[0, 1]\}$.

The execution trace is given in Fig. 20, where in the Z column we can see the set of (truncated) exit intervals over the edge e_9: in the third iteration the exit interval is the whole edge e_9. From the above we conclude that $(e_9, \frac{3}{10})$ is reachable from $(e_0, \frac{1}{2})$.

As an example of a non reachable point, consider the edge signature $(e_1 \dots e_8)^*$ with $[\frac{9}{10}, \frac{19}{20}] \in e_1$ as initial interval and $x_f = \frac{3}{10}$ in e_9 as before. After computing the corresponding functions we obtain that in the first iteration the loop is left and the exit interval on edge e_9 is $[\frac{1}{2}, \frac{5}{6}]$, from where we can conclude that $(e_9, \frac{3}{10})$ is not reachable from $(e_1, [\frac{9}{10}, \frac{19}{20}])$. □

[4] Notice that in fact this function is the same as $\mathsf{Succ}_{e_8 e_1}$ since the other functions are the identity.

Iteration	I	Z
0	$[\frac{1}{10}, \frac{3}{20}]$	$\emptyset$
1	$[\frac{3}{10}, \frac{9}{20}]$	$\emptyset$
2	$[\frac{1}{2}, \frac{3}{4}]$	$\{[0, \frac{1}{6}]\}$
3	$[\frac{7}{10}, 1]$	$\{[0, 1]\}$
4	$[\frac{9}{10}, 1]$	$\{[0, 1]\}$

Figure 20: Execution trace of the cycle $e_1 \ldots e_8$ starting from $[0.1, 0.15] \in e_1$. I represents the current interval and Z is the set of exit intervals (in e_9).

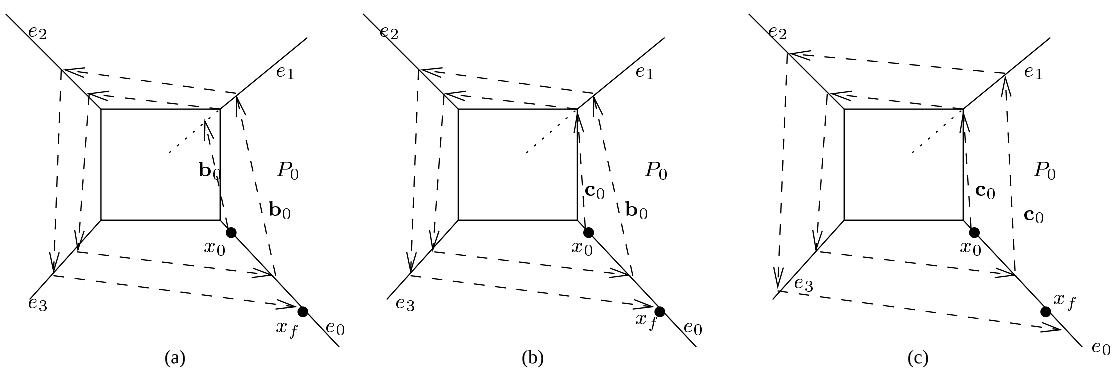


Figure 21: (a) x_f is reachable from x_0 following the leftmost (truncated) trajectory; (b) The dynamics in region P_0 is different the second time the trajectory enters it; (c) Taking $\mathbf{c}_0$ makes x_f unreachable.

7 Concluding remarks

We have presented an algorithm for solving the reachability problem for simple planar differential inclusions. The novelty of the approach for the domain of Hybrid System is the combination of two techniques, namely, the representation of the two-dimensional continuous dynamics as a one-dimensional discrete system (due to Poincaré), and the characterization of the set of qualitative behaviors of the latter as a finite set of types of signatures.

One possible direction of future work is to try to apply the same method for solving the *parameter synthesis problem* for SPDI's, that is, for any two points, $\mathbf{x}_0$ and $\mathbf{x}_f$, assign a constant slope $\mathbf{c}_P \in \phi(P)$ to every region P such that $\mathbf{x}_f$ is reachable from $\mathbf{x}_0$, or conclude that such an assignment does not exist. Clearly, the decidability of the reachability problem does not imply the decidability of the parameter synthesis one. As a simple counter-example just think of an edge signature $(e_0, \ldots, e_3)^3$ with $\mathbf{x}_0, \mathbf{x}_f \in e_0$ (i.e. $\mathbf{x}_0 = (e_0, x_0)$ and $\mathbf{x}_f = (e_0, x_f)$) with differential inclusions $\dot{\mathbf{x}} \in \angle_{\mathbf{a}_0}^{\mathbf{b}_0}$ such that the leftmost truncated successor is different from the non-truncated one for the first iteration ($\mathsf{Succ}^l_{e_0,e_0}(x_0) \neq \widetilde{\mathsf{Succ}}^l_{e_0,e_0}(x_0)$) and they are equal thereinafter. Suppose also that x_f is reachable from x_0 following always the leftmost (truncated) trajectory cycling three times, that is $x_f = \mathsf{Succ}^3_{e_0,e_0}(x_0)$ (see Fig. 21(a)). Clearly, x_f is reachable from x_0 for the SPDI system, but there does not exist a unique value $\mathbf{c}_0 \in \angle_{\mathbf{a}_0}^{\mathbf{b}_0}$ that gives a positive answer to the reachability problem. In Fig. 21(b) we can see that the dynamics in region P_0 is different the second time the trajectory enters it (suppose $\mathbf{c}_0 = \mathbf{a}_0 < \mathbf{b}_0$) whereas Fig. 21(c) shows that the truncated successor restricts the following choice of possible values of the dynamics for P_0 to be at most equal to $\mathbf{c}_0 \in \angle_{\mathbf{a}_0}^{\mathbf{b}_0}$, therefore fixing the slope to be at least $\mathbf{c}_0$ in P_0, makes x_f unreachable (independently of the slopes chosen in the other regions).

Another question that naturally arises is decidability of the reachability problem for hybrid automata whose locations are equipped with SPDI's. We can certainly find (stringent) conditions, such as planarity of the automaton, "memory-less" resets, etc., under which decidability follows almost straightforwardly from the decidability of SPDI's. On the other hand, it is not difficult to see that this problem, without such conditions, is equivalent to deciding whether given a piece-wise linear map f on the unit interval and a point x in this interval, the sequence of iterates x, $f(x)$, $f(f(x))$, and so on, reaches some point y. This last question is still open [12]. And last but not the least, another interesting issue is the complexity analysis of the algorithm. It should be based on counting all "feasible" types of signatures. Our finiteness argument of proposition 3 gives a doubly exponential estimation, which can certainly be improved.

References

[1] R. Alur, C. Courcoubetis, N. Halbwachs, T. Henzinger, P. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *TCS*, 138:3–34, 1995.

[2] R. Alur and D.L. Dill. A theory of timed automata. *TCS*, 126:183–235, 1994.

[3] E. Asarin, O. Bournez, T. Dang, and O. Maler. Reachability analysis of piecewise-linear dynamical systems. In *HSCC'00*, pages 20–31. LNCS 1790, Springer Verlag, 2000.

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

[5] O. Botchkarev and S. Tripakis. Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations. In *HSCC'00*. LNCS 1790, Springer Verlag, 2000.

[6] M. Broucke. A geometric approach to bisimulation and verification of hybrid systems. In *HSCC'99*. LNCS 1569, Springer Verlag, 1999.

[7] K. Čerāns and J. Vīksna. Deciding reachability for planar multi-polynomial systems. In *Hybrid Systems III*. LNCS 1066, Springer Verlag, 1996.

[8] T. Dang and O. Maler. Reachability analysis via face lifting. In *HSCC'98*, pages 96–109. LNCS 1386, Springer Verlag, 1998.

[9] J. Della Dora and S. Yovine. Looking for a methodology for analyzing hybrid systems. Submitted to ECC 2001, 2000.

[10] M. R. Greenstreet and I. Mitchell. Reachability analysis using polygonal projections. In *HSCC'99*. LNCS 1569, pages 103–116. Springer Verlag, 1999.

[11] T.A. Henzinger, P.W. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata? In *27th Annual Symposium on Theory of Computing*, pages 373–382. ACM Press, 1995.

[12] P. Koiran. My favourite problems. http://www.ens-lyon.fr/˜koiran/problems.html.

[13] A.B. Kurzhanski and P. Varaiya. Ellipsoidal techniques for reachability analysis. In *HSCC'00*. LNCS 1790, Springer Verlag, 2000.

[14] G. Lafferriere, G. J. Pappas, and S. Yovine. A new class of decidable hybrid systems. In *HSCC'99*. LNCS 1569, pages 137–151. Springer Verlag, 1999.

[15] O. Maler and A. Pnueli. Reachability analysis of planar multi-linear systems. In *CAV'93*. LNCS 697, pages 194–209. Springer Verlag, 1993.