Asynchronous Correspondences Between Hybrid Trajectory Semantics

Patrick Cousot[0000−0003−0101−9953]

CS, CIMS, New York University, USA
pcousot@cims.nyu.edu https://cs.nyu.edu/~pcousot/

Dedicated to Thomas Henzinger
for his 60^{th} birthday

Abstract. We formalize the semantics of hybrid systems as sets of hybrid trajectories, including those generated by an hybrid transition system. We study the abstraction of hybrid trajectory semantics for verification, static analysis, and refinement. We mainly consider abstractions of hybrid semantics which establish a correspondence between trajectories derived from a correspondence between states such as homomorphisms, simulations, bisimulations, and preservations with progress. We also consider abstractions that cannot be defined stepwise like discretization. All these abstractions are Galois connections between concrete and abstract hybrid trajectory or discrete trace semantics. In contrast to semantic based abstractions, we investigate the problematic trace-based composition of abstractions.

Keywords: Hybrid systems, semantics, abstraction, homomorphism, simulations, bisimulations, preservations, progress, discretization, verification, refinement, abstract interpretation, Galois connection, Galois relation, Logical relation.

1 Introduction

State and transition-based abstractions such as homomorphisms, simulations, bisimulations [26], and preservations with progress (as used in type theory [38]) formalize a correspondence between concrete and abstract discrete semantics. They have been successfully applied to the verification, analysis, and refinement of programs. In program refinement, such state and transition-based abstractions are used to transform specifications into implementations. In program verification and analysis they are used to simplify the reasoning on properties of program executions.

All these abstractions have two fundamental properties. The first is that a reasoning on computation steps (via a transition system) is sufficient to establish a correspondence between program semantics (which is the set of all their possible maximal executions). The second is that they compose. For example the composition of simulations is a simulation. This allows, for example, for stepwise

refinement in program construction or composing successive sound abstractions in program verification.

Our objective is to extend and study these state and transition-based abstractions for dynamical systems that exhibits both continuous and discrete dynamic behavior as found in cyber-physical systems. We consider concrete and abstract hybrid semantics (that is sets of sequences of configurations specifying continuous behaviors between discrete changes of modes) that allow for arbitrary timings, arbitrary continuous dynamic mode changes, and arbitrary evolutions of the states over time. We also consider hybrid semantics generated by hybrid transition systems hoping that, as in the discrete case, the abstraction of transition systems will induce the abstraction of the hybrid semantics. But contrary to the discrete case, this is problematic.

Such hybrid trajectory semantics can be understood as specifications, implementations, or abstractions of hybrid dynamical systems. They are more general than particular abstract models of hybrid systems such as synchronous systems [7], timed automata [2], switched systems (for which the sequence of modes and mutation times are known in advance) [22], hybrid automata [1], including restrictions for decidability subclasses [4,18], Simulink [24], and so on. Hybrid trajectory semantics can also be used to specify the semantics of these abstract models, that is, the set of possible behaviors that they describe.

We study homomorphisms, simulations, (bisimulations, preservations with progress in the ArXiv version) between concrete and abstract hybrid semantics as well as discretization of hybrid semantics to establish a correspondence between an hybrid system and a discrete system (such as a computer). Considered as semantic transformers they all form Galois connections and so do compose. However, when considering individual concrete and abstract trajectories, the problem is that in full generality, these abstraction may not compose well. For examples the discretization of two trajectories of (bi)simular hybrid systems may not be (bi)simular discrete traces. We investigate sufficient conditions to solve this compositionally problem when reasoning on individual trajectories.

The paper organized as follows. In section 2 we recall the definitions of Galois connections, Galois relations (ordered logical relations), and tensor products. In section 3, we introduce hybrid trajectory semantics to define the arbitrary evolution of hybrid systems over time. In section 4, we introduce hybrid transition systems that can be used to generate hybrid trajectory semantics (the same way that discrete transition systems generate a discrete trace-based operational semantics for discrete systems). In section 5, we consider the abstraction of hybrid trajectory semantics by reasoning on trajectories, that is executions of the hybrid system as defined by its semantics. It is often considered that reasoning on states, or consecutive states, is simpler that reasoning on full trajectories (although less general). This is the objective of section 6, where an abstraction of states is shown to induce an abstraction of hybrid trajectories, hence of hybrid semantics (which are sets of hybrid trajectories). In case the hybrid semantics is defined by a transition system, we consider in section 7 the abstraction of transition systems by homomorphisms, simulations, (bisimulation and preservation

with progress in the ArXiv version) and study which abstraction of trajectories and hybrid semantics they induce. The main difficulty is that concrete and abstract trajectories may have different, not necessarily comparable timelines, that is timings for mode changes. A difficulty, in particular for discretization, is that the abstraction of transition systems may not be an abstraction of their hybrid semantics (which is never the case for discrete systems). We solve the problem under sufficient conditions. We conclude in section 8.

2 Galois connections and relations

2.1 Galois connections

A Galois connection $\langle C,\, \sqsubseteq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A,\, \preccurlyeq\rangle$[1] between posets $\langle C,\, \sqsubseteq\rangle$ and $\langle A,\, \preccurlyeq\rangle$ is a pair of an abstraction function α and a concretization function γ such that

$$\langle C,\, \sqsubseteq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle A,\, \preccurlyeq\rangle \triangleq \begin{cases} \alpha \in C \nearrow A & \text{is increasing} & (1.\text{a}) \\ \gamma \in A \nearrow C & \text{is increasing} & (1.\text{b}) \\ \gamma \circ \alpha & \text{is an upper closure} & (1.\text{c}) \\ \alpha \circ \gamma & \text{is a lower closure} & (1.\text{d}) \end{cases} \tag{1}$$

where an upper closure is increasing, idempotent, and extensive ($\forall c \in C \,.\, x \sqsubseteq \gamma \circ \alpha(x)$) while a lower closure is increasing, idempotent, and reductive ($\forall y \in A \,.\, \alpha \circ \gamma(y) \preccurlyeq y$). An equivalent definition of a Galois connection is a pair of increasing functions satisfying

$$\forall x \in C \,.\, \forall y \in A \,.\, \alpha(x) \preccurlyeq y \Longrightarrow x \sqsubseteq \gamma(y) \quad \wedge \tag{2}$$

$$\alpha(x) \preccurlyeq y \Longleftarrow x \sqsubseteq \gamma(y) \tag{3}$$

Example 1 (Classic examples of Galois connections). Set transformers form Galois connections

$$\langle \wp(\mathsf{S}),\, \subseteq\rangle \underset{\mathsf{pre}[r]}{\overset{\widetilde{\mathsf{post}}[r]}{\leftrightarrows}} \langle \wp(\overline{\mathsf{S}}),\, \subseteq\rangle \text{ and } \langle \wp(\mathsf{S}),\, \subseteq\rangle \underset{\mathsf{post}[r]}{\overset{\widetilde{\mathsf{pre}}[r]}{\leftrightarrows}} \langle \wp(\overline{\mathsf{S}}),\, \subseteq\rangle \tag{4}$$

where $r \in \wp(\mathsf{S} \times \overline{\mathsf{S}})$, $\mathsf{post}[r]P \triangleq \{y \mid \exists x \in P \,.\, \langle x,\, y\rangle \in r\}$, $\mathsf{pre}[r] = \mathsf{post}[r^{-1}]$, $r^{-1} \triangleq \{\langle y,\, x\rangle \mid \langle x,\, y\rangle \in r\}$, $\tilde{f} \triangleq \neg \circ f \circ \neg$, and $(f \circ g)(x) = f(g(x))$ is function composition.

Another classic example is an homomorphic abstraction, where given $h \in \mathsf{S} \to \overline{\mathsf{S}}$, $\alpha_h(X) \triangleq \{h(x) \mid x \in X\}$, and $\gamma_h(Y) \triangleq \{x \in \mathsf{S} \mid h(x) \in Y\}$, we have

$$\langle \wp(\mathsf{S}),\, \subseteq\rangle \underset{\alpha_h}{\overset{\gamma_h}{\leftrightarrows}} \langle \wp(\overline{\mathsf{S}}),\, \subseteq\rangle \tag{5}$$ □

Interpreting C in (1) as a concrete semantics (e.g. a set of execution discrete traces or hybrid trajectories) and A as an abstract semantics, the concretization

[1] see an introduction in [11, Ch. 11]

$\gamma(y)$ is the concrete semantics corresponding to the abstract semantics $y \in A$, that is its concrete meaning. Conversely, $\alpha(x)$ is the abstraction of the concrete semantics $x \in C$.

The conditions (1.a) and (1.b) of order preservation express that the notions of over approximation in the concrete and the abstract are the same.

Condition (1.c) implies $x \sqsubseteq \gamma(\alpha(x))$. This expresses that $\alpha(x)$ is an abstract sound over approximation of x.

Condition (1.c) with $y = \alpha(x)$ implies (3) which expresses that $\alpha(x)$ is the best abstraction of x (since given any other abstraction of x which is sound, that is $x \sqsubseteq \gamma(y)$, $\alpha(x)$ is more precise since $\alpha(x) \preccurlyeq y$).

2.2 Galois relations

Any Galois connection $\langle \mathcal{C}, \sqsubseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \mathcal{A}, \preccurlyeq \rangle$ can be encoded by a *Galois relation* $R_\alpha \in \wp(\mathcal{C} \times \mathcal{A})$ (also called ordered logical relations) defined as

$$R_\alpha \triangleq \{\langle x, y \rangle \in C \times A \mid \alpha(x) \preccurlyeq y\} = \{\langle x, y \rangle \in C \times A \mid x \sqsubseteq \gamma(y)\} \tag{6}$$

If $\langle \mathcal{C}, \sqsubseteq, \bigsqcup \rangle$ and $\langle \mathcal{A}, \preccurlyeq, \curlywedge \rangle$ are complete lattices such relations R_α satisfy the following characteristic properties of Galois relations R.

$$\begin{aligned}
&(x \sqsubseteq x' \wedge \langle x', y' \rangle \in R \wedge y' \preccurlyeq y) \Longrightarrow (\langle x, y \rangle \in R) && \text{(a)}\\
&(\forall i \in \Delta \,.\, \langle x_i, y \rangle \in R) \Longrightarrow \langle \bigsqcup_{i \in \Delta} x_i, y \rangle \in R && \text{(b)}\\
&(\forall i \in \Delta \,.\, \langle x, y_i \rangle \in R) \Longrightarrow \langle x, \curlywedge_{i \in \Delta} y_i \rangle \in R && \text{(c)}
\end{aligned} \tag{7}$$

The tensor product $\langle \mathcal{C}, \sqsubseteq \rangle \otimes \langle \mathcal{A}, \preccurlyeq \rangle$ of two complete lattices $\langle \mathcal{C}, \sqsubseteq \rangle$ and $\langle \mathcal{A}, \preccurlyeq \rangle$ is [34]

$$\langle \mathcal{C}, \sqsubseteq \rangle \otimes \langle \mathcal{A}, \preccurlyeq \rangle \triangleq \{R \in \wp(\mathcal{C} \times \mathcal{A}) \mid R \text{ is a relation satisfying (7)}\} \tag{8}$$

Galois connections and relations are mathematically equivalent. If $\langle \mathcal{C}, \sqsubseteq \rangle$ and $\langle \mathcal{A}, \preccurlyeq \rangle$ be complete lattices then $\langle \mathcal{C}, \sqsubseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \mathcal{A}, \preccurlyeq \rangle$ if and only if $R_\alpha \in \langle \mathcal{C}, \sqsubseteq \rangle \otimes \langle \mathcal{A}, \preccurlyeq \rangle$ where R_α is defined in (6) and, conversely, $\alpha(x) \triangleq \curlywedge \{y \mid \langle x, y \rangle \in R_\alpha\}$ and $\gamma(y) \triangleq \bigsqcup \{x \mid \langle x, y \rangle \in R_\alpha\}$.

Dual definitions of Galois connections and relations can be used to cope with under approximation.

3 Hybrid trajectory semantics

Time. We let the time t run over the set $\mathbb{R}_{\geqslant 0}$ of all positive reals.

States and flows. We let S be a set of states. In our pictures, we use Cartesian coordinates where the horizontal axis is time and the vertical axis is the set of states (which we take to be $\mathsf{S} = \mathbb{R}$).

Flows. A flow $f \in \mathsf{F} \triangleq \mathbb{R}_{\geqslant 0} \nrightarrow \mathsf{S}$ is a partial map from time to states representing the evolution of the state over time. Flows can be specified e.g. by ODEs over a period of time (with appropriate hypothesis, see e.g. [21, Ch. XIX], [19, ch. 8 & 9], [20], and [30]).

Time intervals. If $t_1 \in \mathbb{R}_{\geqslant 0}$, $t_2 \in \mathbb{R}_{\geqslant 0} \cup \{\infty\}$, and $t_1 < t_2$ then $[t_1, t_2[\triangleq \{t \in \mathbb{R}_{\geqslant 0} \mid t_1 \leqslant t < t_2\}$ is the interval of time between t_1 and t_2, the lower bound $\mathsf{b}([t_1, t_2[) \triangleq t_1$ being included while the upper bound $\mathsf{e}([t_1, t_2[) \triangleq t_2$ is excluded. The set of all such time intervals is

$$i \in \mathsf{I} \triangleq \{[t_1, t_2[\mid t_1 \in \mathbb{R}_{\geqslant 0} \wedge t_2 \in \mathbb{R}_{\geqslant 0} \cup \{\infty\} \wedge t_1 + \zeta \leqslant t_2\} \tag{9}$$

where $\zeta > 0$ is any arbitrarily chosen infinitesimal defining the minimal duration $\mathsf{d}(i) \triangleq \mathsf{e}(i) - \mathsf{b}(i)$ of a time interval i. This implies that the duration of successive configurations cannot tend to 0 so we exclude *zeno* systems (with infinitely many successive configurations in a finite interval of time [39]).

The closure of an interval $\mathsf{cl}([t_1, t_2[) \triangleq [t_1, t_2]$ if $t_2 \neq \infty$ and $\mathsf{cl}([t_1, \infty[) = [t_1, \infty[$ includes the upper bound unless it is infinite. By convention, $[t_1, \infty] = [t_1, \infty[= \{t \in \mathbb{R}_{\geqslant 0} \mid t_1 \leqslant t\}$. We let $\mathsf{cl}(\mathsf{I}) \triangleq \{\mathsf{cl}(i) \mid i \in \mathsf{I}\}$.

Configurations. A configuration is a pair of a flow and a time interval

$$c \in \mathsf{C} \triangleq \{\langle f,\, i\rangle \in \mathsf{F} \times \mathsf{I} \mid \forall t \in i \,.\, f(t) \in \mathsf{S}\} \tag{10}$$

while final configurations include the upper bound

$$c \in \mathsf{cl}(\mathsf{C}) \triangleq \{\langle f,\, i\rangle \in \mathsf{F} \times \mathsf{cl}(\mathsf{I}) \mid \forall t \in i \,.\, f(t) \in \mathsf{S}\} \tag{11}$$

such that the flow is well-defined in the set of states S on the time interval, i.e. $i \subseteq \mathsf{dom}(f)$. A configuration $c = \langle f, i\rangle$ starts a time $\mathsf{b}(c) = \mathsf{b}(i)$ and ends at time $\mathsf{e}(c) = \mathsf{e}(i)$, excluded in (10) and included in (11). We call $\mathsf{dom}(c) = i$ the time interval of configuration c. Notice that by the choice of the infinitesimal $\zeta > 0$ and the definition (9) of I, the intervals $i \in \mathsf{I}$ in (10) and (11) cannot be empty.

Configurations c record the evolution of the state as specified by the flow during the period of time $\mathsf{dom}(\mathrm{c})$. During that time interval the definition of the flow f, which is the law of continuous evolution of the system as a function of the time, is fixed. It may be different in the next configuration of the system. In that case, it is common to say that the mode of the hybrid system has changed. The duration $\mathsf{d}(c) = \mathsf{e}(c) - \mathsf{b}(c) \geqslant \zeta$ of the configuration is lower-bounded by $\zeta > 0$ so that infinite sequences of configurations are always nonzeno. Additional hypotheses might be necessary on the flow f of configurations $\langle f, i\rangle$ such as continuity, uniform continuity, Lipschitz continuity, etc. However, discontinuities are always allowed (but not mandatory) when changing mode between consecutive configurations.

By convention the state of a configuration c at time $t \in \mathbb{R}_{\geqslant 0}$ is

$$\begin{aligned} c(t) &\triangleq f(t) && \text{if } c = \langle f,\, i\rangle \text{ and } t \in i \\ &\triangleq \text{undefined} && \text{otherwise} \end{aligned} \tag{12}$$

Let us define the concatenation of two consecutive configurations $\langle f,\, i\rangle \in \mathsf{C}$ and $\langle f',\, i'\rangle \in \mathsf{C} \cup \mathsf{cl}(\mathsf{C})$ where $\mathsf{e}(i) = \mathsf{b}(i')$ (i.e. the concatenation is undefined for non-consecutive intervals).

$$\langle f,\, i\rangle \fatsemi \langle f',\, i'\rangle \triangleq \langle f'',\, i \cup i'\rangle \text{ where } \begin{cases} f''(t) = f(t) \text{ when } t \in i \\ f''(t) = f'(t) \text{ when } t \in i' \end{cases} \tag{13}$$

Since the state at the beginning of a configuration may be different from the state at the end of the previous configuration at the same time, definitions (13), (15), and (23) favor states at the beginning of configurations (because intervals are left closed and open right).

To simplify notations, the empty configuration is, by convention, $\varepsilon \triangleq \langle \emptyset,\, \emptyset\rangle$ where $\emptyset$ is the empty set, that is, the everywhere undefined function. By convention, $\mathsf{b}(\varepsilon) \triangleq +\infty$ and $\mathsf{e}(\varepsilon) = -\infty$ so that $\min(t, \mathsf{b}(\varepsilon)) = \max(t, \mathsf{e}(\varepsilon)) = t$ when $t \in \mathbb{R}_{\geqslant 0}$. Observe that although $\varepsilon \notin \mathsf{C} \cup \mathsf{cl}(\mathsf{C})$ since the time interval is empty, we nevertheless have $\langle f,\, i\rangle \fatsemi \varepsilon \triangleq \varepsilon \fatsemi \langle f,\, i\rangle \triangleq \langle f,\, i\rangle$, for ease of writing.

The selection of a time slice during the configuration time interval.

$$\begin{array}{lll} \langle f,\, i\rangle(\!|t_1, t_2|\!) \triangleq \langle f,\, i \cap [t_1, t_2]\rangle & \text{where} & \mathsf{b}(i \cap [t_1, t_2]) + \zeta \leqslant \mathsf{e}(i \cap [t_1, t_2]) \\ \langle f,\, i\rangle(\!|t_1, t_2(\!| \triangleq \langle f,\, i \cap [t_1, t_2[\rangle & & \mathsf{b}(i \cap [t_1, t_2[) + \zeta \leqslant \mathsf{e}(i \cap [t_1, t_2[) \end{array} \tag{14}$$

In particular, we define $\varepsilon(\!|t_1, t_2|\!) \triangleq \varepsilon(\!|t_1, t_2(\!| \triangleq \varepsilon$.

Trajectories The trajectories over configurations C are nonempty finite or infinite sequences of contiguous configurations.

$$\begin{array}{ll} \mathsf{T}^n_{\mathsf{C}} \triangleq \{\sigma \in [0, n] \to \mathsf{cl}(\mathsf{C}) \mid \mathsf{b}(\sigma_0) = 0 \wedge \forall i \in [0, n[\,.\, \sigma_i \in \mathsf{C} \wedge \\ \qquad\qquad \mathsf{e}(\sigma_i) = \mathsf{b}(\sigma_{i+1}) \wedge \sigma_n \in \mathsf{cl}(\mathsf{C})\} \\ \qquad\qquad \text{finite trajectories } \sigma \in \mathsf{T}^n_{\mathsf{C}} \text{ of length } |\sigma| = n+1,\ n \in \mathbb{N} \\ \mathsf{T}^+_{\mathsf{C}} \triangleq \bigcup_{n \in \mathbb{N}} \mathsf{T}^n_{\mathsf{C}} \qquad \text{finite nonempty trajectories} \\ \mathsf{T}^\infty_{\mathsf{C}} \triangleq \{\sigma \in \mathbb{N} \to \mathsf{C} \mid \mathsf{b}(\sigma_0) = 0 \wedge \forall i \in \mathbb{N} \,.\, \mathsf{e}(\sigma_i) = \mathsf{b}(\sigma_{i+1})\} \\ \qquad\qquad \text{infinite trajectories } \sigma \in \mathsf{T}^\infty_{\mathsf{C}} \text{ of length } |\sigma| = \infty \\ \mathsf{T}^{+\infty}_{\mathsf{C}} \triangleq \mathsf{T}^+_{\mathsf{C}} \cup \mathsf{T}^\infty_{\mathsf{C}} \qquad \text{nonempty trajectories} \end{array} \tag{15}$$

A finite or infinite trajectory $\sigma \in [0, |\sigma|[\to \mathsf{C}$ is a sequence of configurations that will be denoted $\sigma = \langle \sigma_i,\, i \in [0, |\sigma|[\rangle$. Such a trajectory σ records the evolution of the state along discrete changes of the flows encoded by configurations. The state at the end of a configuration is that of the next state, if any. Therefore, the configuration intervals are open right and consecutive except for the last one in finite trajectories which is closed. No configuration in a trajectory can be empty.

We let $\sigma[i, j]$ denote the subsequence of configurations in σ of ranks i to j, $i, j \in [0, |\sigma|[$. $\sigma[i, j[$ excludes j (usually ∞).

Traces. We let traces $\varsigma \in \mathsf{T}_{\mathsf{S}}^{+\infty}$ be discrete finite or infinite untimed sequences of states in S and use the same notations for continuous trajectories and discrete traces. The homomorphic timeline abstraction ((_?_ : _) is the conditional)

$$\begin{aligned}\alpha_{tl}(\sigma) &\triangleq \boldsymbol{\lambda}\, i \in [0, |\sigma|] \bullet (i = 0 \;?\; 0 \;:\; (i = \infty \;?\; \infty \;:\; \mathsf{e}(\sigma_{i-1})))\\ \alpha_{tl}(T) &\triangleq \{\alpha_{tl}(\sigma) \mid \sigma \in T\}\end{aligned}$$

such that, by (5), $\langle \mathsf{T}_{\mathsf{C}}^{+\infty}, \subseteq\rangle \underset{\alpha_{tl}}{\overset{\gamma_{tl}}{\leftrightarrows}} \langle \mathsf{T}_{\mathbb{R}_{\geqslant 0}\cup\{\infty\}}^{+\infty}, \subseteq\rangle$ is an example of abstraction of trajectories into traces (by projection of the mode change timings).

Hybrid trajectory semantics and properties. Given a set S of states and the corresponding configurations C in (10), a hybrid trajectory semantics $\mathcal{S}_{\mathsf{C}} \in \wp(\mathsf{T}_{\mathsf{C}}^{+\infty})$ is a subset of all possible trajectories (15). Properties of hybrid trajectory semantics belong to $\wp(\wp(\mathsf{T}_{\mathsf{C}}^{+\infty}))$ (sometimes called hyper properties) while there abstraction $\alpha_{\cup}(P) = \bigcup P$ into trajectory properties belong to $\wp(\mathsf{T}_{\mathsf{C}}^{+\infty})$.

Similarly a trace semantics $\mathcal{S}_{\mathsf{S}} \in \wp(\mathsf{T}_{\mathsf{S}}^{+\infty})$ is a subset of all possible traces.

Trajectory states. The duration $[\![\sigma]\!]$ of a trajectory σ is

$$\begin{aligned}[\![\sigma]\!] &\triangleq \sum_{k=0}^{n} \mathsf{e}(\sigma_i) - \mathsf{b}(\sigma_i) = \mathsf{e}(\sigma_n) \qquad \text{when} \quad \sigma \in \mathsf{T}_{\mathsf{C}}^{n} \qquad (16)\\ &\triangleq \sum_{k=0}^{\infty} \mathsf{e}(\sigma_i) - \mathsf{b}(\sigma_i) = \infty \qquad \text{when} \quad \sigma \in \mathsf{T}_{\mathsf{C}}^{\infty} \qquad \text{(nonzeno hypothesis)}\end{aligned}$$

as indicated by the time at which the last configuration in the trajectory ends or ∞ for infinite trajectories.

Time-evolution law abstraction. A trajectory σ can be abstracted into a function $\alpha_{tr}(\sigma) \in \mathbb{R}_{\geqslant 0} \to \mathsf{S}$ mapping time to a state such that

$$\begin{aligned}\mathsf{dom}(\alpha_{tr}(\sigma)) &\triangleq [0, [\![\sigma]\!]] \qquad \text{(by convention, excluding } \infty \text{ if } [\![\sigma]\!] = \infty)\\ \alpha_{tr}(\sigma)(t) &\triangleq f(t) \text{ such that } \exists k \in [0, |\sigma|[\,.\, \sigma_k = \langle f,\, i\rangle \wedge t \in i \qquad (17)\\ \sigma_t &\triangleq \alpha_{tr}(\sigma)(t) \qquad \text{(abbreviated notation)}\end{aligned}$$

So we have two different representations of trajectories, σ in (15) and $\alpha_{tr}(\sigma)$ in (17), this second representation being closer to the time-evolution law of the theory of dynamical systems [20]. Notice that $\alpha_{tr}(\sigma)$ is a function defined by parts on the timeline abstraction $\alpha_{tl}(\sigma)$ of the trajectory σ so the time-evolution law $\alpha_{tr}(\sigma)$ is not simpler that the trajectory σ to reason upon, in particular because the timeline information is abstracted away.

We leave this α_{tr} abstraction implicit and use the same notation for both cases. Therefore a trajectory σ is either a discrete sequence of configurations

$\sigma = \langle \sigma_i,\ i \in [0, |\sigma|[\rangle$ or a state function of the time $\sigma = \langle \sigma_t,\ t \in [0, \lbrack\!\lbrack \sigma \rbrack\!\rbrack[\rangle$ where $\sigma_t \triangleq \alpha_{tr}(\sigma)(t)$. By homomorphic abstraction (5), this extends to hybrid trajectory semantics T with $\alpha_{tr}(T) \triangleq \{\alpha_{tr}(\sigma) \mid \sigma \in T\}$

$$\langle \wp(\mathsf{T}_{\mathsf{C}}^{+\infty}),\ \subseteq \rangle \underset{\alpha_{tr}}{\overset{\gamma_t}{\leftrightarrows}} \langle \wp(\mathbb{R}_{\geqslant 0} \to \mathsf{S}),\ \subseteq \rangle \tag{18}$$

Maximal trajectory semantics. A trajectory semantics $T \in \wp(\mathsf{T}_{\mathsf{C}}^{+\infty})$ on configurations C is a set of finite or infinite trajectories. Let us define the maximal trajectories of T as those without strict prefixes

$$\max(T) \triangleq \{\langle \sigma_i,\ i \in [0, |\sigma|[\rangle \in T \mid \forall n < |\sigma| \,.\, \langle \sigma_i,\ i \in [0, n[\rangle \notin T\}$$

A maximal trajectory semantics has no strict prefixes, that is $\max(T) = T$.

Example 2 (Specification of a water tank [17]). A water tank (or water dam) runs for ever with a continuous inflow and a valve (or spillway floodgate) than can be opened or shut to control the outflow. The objective is to design a controller to maintain the water level y between 0 and 3 (for some length unit). When the valve is opened, the water level y decreases while, when the valve is shut down, the water level y increases. The tank should never remain empty more than ζ units of time.

Define states

$$s \in \mathsf{S} \triangleq \mathbb{R} \times \{\mathit{open}, \mathit{shut}\} \tag{19}$$

such that $s.y \in \mathbb{R}$ and $s.v \in \{\mathit{open}, \mathit{shut}\}$. Let C be the corresponding set (10) of configurations. The above informal specification can be formalized by the following abstract hybrid semantics of the water tank.

$$\begin{aligned} P(\sigma) \triangleq\ & \forall t \in \mathbb{R}_{\geqslant 0} \,.\, 0 \leqslant \sigma(t).y \leqslant 3 \wedge \forall t_2 > t_1 \geqslant 0 \,. && \text{(a)} \\ & \forall t \in [t_1, t_2] \,.\, \sigma(t).v = \mathit{open} \Longrightarrow \sigma(t_1).y > \sigma(t_2).y \wedge && \text{(b)} \\ & \forall t \in [t_1, t_2] \,.\, \sigma(t).v = \mathit{shut} \Longrightarrow \sigma(t_1).y < \sigma(t_2).y \wedge && \text{(c)} \\ & \forall t \in \mathbb{R}_{\geqslant 0} \,.\, \sigma(t).y = 0 \Longrightarrow \sigma(t + \zeta).y > 0 && \text{(d)} \end{aligned} \tag{20}$$

The hybrid semantics specification the water tank is then

$$\mathcal{S}^2 \triangleq \{\sigma \in \{0\} \to \mathsf{C} \mid \mathsf{b}(\sigma_0) = 0 \wedge \mathsf{e}(\sigma_0) = \infty \wedge P(\sigma_0)\} \tag{21}$$

with only one configuration, or using the homomorphic abstraction (17),

$$\mathcal{S}^2 \triangleq \{\sigma \in \mathbb{R}_{\geqslant 0} \to \mathsf{S} \mid P(\sigma)\}$$ □

4 Transition-based hybrid trajectory semantics

As in the discrete case, a simple way to define a hybrid trajectory semantics, is to first define a hybrid transition system and then to consider the hybrid semantic defined as the set of all possible trajectories generated by the hybrid transition system. As is the case for discrete trace semantics, not all hybrid semantics can be generated by a hybrid transition system on the same set of configurations (which cannot e.g. express fairness without adding a scheduler to the transition system or adding conditions on the generated traces).

Hybrid transition system. A hybrid transition system is defined by a triple $\langle \mathsf{C},\ \mathsf{C}^0,\ \tau \rangle$ of a set of configurations C, initial configurations C^0 and a transition relation $\tau \in \wp(\mathsf{C} \times (\mathsf{C} \cup \mathsf{cl}(\mathsf{C})))$ such that

$$\begin{array}{ll} \text{initial configurations} & \mathsf{C}^0 \subseteq \{c \in \mathsf{C} \mid \mathsf{b}(c) = 0\} \\ \text{consecutiveness} & \forall \langle c,\ c' \rangle \in \tau \ .\ c \in \mathsf{C} \wedge \mathsf{e}(c) = \mathsf{b}(c') \\ \text{closeness of final configurations} & \forall c \ .\ (\forall c' \ .\ \langle c,\ c' \rangle \notin \tau) \iff c \in \mathsf{cl}(\mathsf{C}) \end{array} \tag{22}$$

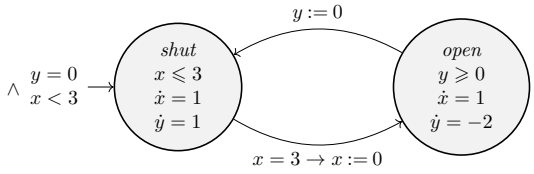
Maximal trajectory semantics of a transition system. A transition semantics $\langle \mathsf{C},\ \mathsf{C}^0,\ \tau \rangle$ is usually used to define a hybrid trajectory semantics $[\![\langle \mathsf{C},\ \mathsf{C}^0,\ \tau \rangle]\!]$ abbreviated $[\![\tau]\!]$, for example the maximal one.

$$\begin{array}{rcl} [\![\tau]\!]^n & \triangleq & \{\sigma \in \mathsf{T}_{\mathsf{C}}^n \mid \sigma_0 \in \mathsf{C}^0 \wedge \forall i \in [0, n[\ .\ \langle \sigma_i,\ \sigma_{i+1} \rangle \in \tau \wedge \forall c \ .\ \langle \sigma_n,\ c \rangle \notin \tau\} \\ [\![\tau]\!]^+ & \triangleq & \displaystyle\bigcup_{n \in \mathbb{N}} [\![\tau]\!]^n \\ [\![\tau]\!]^\infty & \triangleq & \{\sigma \in \mathsf{T}_{\mathsf{C}}^\infty \mid \sigma_0 \in \mathsf{C}^0 \wedge \forall i \in \mathbb{N} \ .\ \langle \sigma_i,\ \sigma_{i+1} \rangle \in \tau\} \\ [\![\tau]\!] & \triangleq & [\![\tau]\!]^+ \cup [\![\tau]\!]^\infty \end{array} \tag{23}$$

The trajectories of $[\![\tau]\!]$ are maximal, that is,

$$\max([\![\tau]\!]) = [\![\tau]\!] \tag{24}$$

Example 3 (Water tank automaton [17]). Continuing example 2, the water tank specification can be implemented as described by the following hybrid automaton.



As soon as the tank is empty, the valve is shut down. The valve is reopened after 3 units of time.

The states, configurations, initial configurations, and transitions are (we write $\dot{x}$ for the derivative $\frac{dx}{dt}$ of the everywhere differentiable (hence continuous) real-valued function $x(t)$ of the time t).

$$\begin{array}{rcl} \mathsf{S} & \triangleq & \{open, shut\} \times \mathbb{R} \times \mathbb{R} \\ \mathsf{C}^{shut} & \triangleq & \{\langle f,\ [t_1, t_2[\rangle \mid \exists x, y \ .\ \forall t \in [t_1, t_2] \ .\ f(t) = \langle shut,\ x(t),\ y(t) \rangle \wedge \\ & & \quad (t = t_1 \Longrightarrow y(t) = 0) \wedge x(t) \leqslant 3 \wedge (x(t) = 3 \Longrightarrow t = t_2) \\ & & \quad \wedge \dot{x}(t) = 1 \wedge \dot{y}(t) = 1\} \\ \mathsf{C}^{open} & \triangleq & \{\langle f,\ [t_1, t_2[\rangle \mid \exists x, y \ .\ \forall t \in [t_1, t_2] \ .\ f(t) = \langle open,\ x(t),\ y(t) \rangle \wedge \\ & & \quad (t = t_1 \Longrightarrow x(t) = 0) \wedge y(t) \geqslant 0 \wedge (y(t) = 0 \Longrightarrow t = t_2) \\ & & \quad \wedge \dot{x}(t) = 1 \wedge \dot{y}(t) = -2\} \\ \mathsf{C} & \triangleq & \mathsf{C}^{shut} \cup \mathsf{C}^{open} \end{array}$$

$$\mathsf{C}^0 \triangleq \{\langle f,\, [0,t[\rangle \in \mathsf{C}^{shut} \mid t > 0 \wedge \exists x < 3 \,.\, f(0) = \langle shut,\, x,\, 0\rangle\} \in \mathsf{C}^{shut}$$
$$\tau^3 \triangleq (\mathsf{C}^{shut} \times \mathsf{C}^{open}) \cup (\mathsf{C}^{open} \times \mathsf{C}^{shut}) \text{ as restricted by (22)} \tag{25}$$

Notice that the final time t_2 is not part of the time interval of configurations in C^{shut} but, by (22), the starting time of the next configuration in C^{open}. Therefore, at that time the value of x is 0, not 3. So in this example, f is continuous on $[t_1, t_2[$ that is continuous on $]t_1, t_2[$ and right continuous at t_1. Same for y in C^{open}. An example of execution is given in figure (5.b). The hybrid semantics $[\![\tau^3]\!]$ of the water tank automaton is given by (23). □

Lemma 1. [2]

If $\tau \subseteq \tau'$ *and the* blocking condition *holds, i.e.*

$$\forall c \,.\, (\forall c' \,.\, \langle c,\, c'\rangle \notin \tau) \Longrightarrow (\forall c' \,.\, \langle c,\, c'\rangle \notin \tau') \tag{26}$$

then $[\![\tau]\!] \subseteq [\![\tau']\!]$.

(so if $[\![\tau']\!]$ has trajectory property $P \in \wp(\mathsf{T}^{+\infty}_{\mathsf{C}})$ (i.e. $[\![\tau']\!] \subseteq P$) then lemma 1 implies that $[\![\tau]\!]$ has the same property P.)

Observe that the transition of one configuration to the next in (22) requires the specification of the time at which the next configuration will terminate. As shown by the water tank automaton example 3 of [17], this is not a problem when the duration of the configuration is specified by a condition on the flow.

5 Trajectory-based hybrid trajectory semantics abstraction

In many program verification and refinement methods, the hybrid semantics is abstracted or concretized to simplify soundness and completeness proofs. One way of simplifying the proofs is to reason on an abstraction of trajectories, by applying an homomorphic abstraction (5) to these trajectories.

A classic example is sampling in signal processing, to reduce a continuous-time signal to a discrete-time signal. For an hybrid semantics, this is defined as follows.

Let $\delta > 0$ be a sampling interval (see [29, Ch. 9] for an adequate choice of the sampling rate). Define

$$\begin{aligned} h_\delta(\sigma) &\triangleq \langle \sigma_{n\delta},\, n \in \mathbb{N} \wedge n\delta \leqslant [\![\sigma]\!]\rangle \\ \alpha_\delta(T) &\triangleq \{h_\delta(\sigma) \mid \sigma \in T\} \end{aligned} \tag{27}$$

which, by (5), is an homomorphic Galois connection

$$\langle \wp(\mathsf{T}^{+\infty}_{\mathsf{C}}),\, \subseteq\rangle \underset{\alpha_\delta}{\overset{\gamma_\delta}{\leftrightarrows}} \langle \wp(\mathsf{T}^{+\infty}_{\mathsf{S}}),\, \subseteq\rangle \tag{28}$$

[2] Underlined equation or theorem numbers link to proofs given in the ArXiv version.

where $\gamma_\delta(\Theta) \triangleq \{\sigma \in \mathsf{T}_{\mathsf{C}}^{+\infty} \mid h_\delta(\sigma) \in \Theta\}$.

(In general a trajectory σ cannot be regained from its discretization $h_\delta(\sigma)$. This might be possible under specific hypotheses. For example, the Nyquist–Shannon sampling theorem [28,33] establishes a sufficient condition for a sample rate that permits a discrete sequence of samples to capture all the information from a continuous-time signal of finite bandwidth.)

Trajectory based abstractions are useful to prove trajectory properties of hybrid systems by considering one possible trajectory at a time (but inadequate to prove (hyper) properties relating two of more trajectories). But reasoning on a complete trajectory is often complicated, in which case local reasonings relating states or transitions locally are preferred.

6 State-based hybrid trajectory semantics abstraction

Since reasoning on discrete execution traces (hence on hybrid trajectories) is difficult, a number of proof techniques have been developed to reduce the reasoning on trajectories to reasonings on states (or pairs of states, that is transitions). Examples are discrete simulations that we extend to hybrid trajectories (and bisimulation [26] as well as preservation with progress [38] considered in the ArXiv version). Sampling in (28) is a counter example since, in general, sampling must be defined by reasoning on trajectories, not states and transitions.

Our objective is to show that a relation between states can be extended to configurations, then to trajectories, and then to hybrid semantics (independently of whether trajectories are generated by transition systems or not).

6.1 Relation between states

For timed trajectories, the relation r between concrete states S to abstract states $\overline{\mathsf{S}}$ is a function of the time.

$$r \in \mathbb{R}_{\geqslant 0} \to \wp(\mathsf{S} \times \overline{\mathsf{S}}) \tag{29}$$

For simplicity, we assume r to be a total function of the time. If necessary, a partial function could be encoded using an undefined element (like $\bot$ in denotational semantics).

6.2 Relation between configurations

Let us define a partial relation between configurations with related states

$$\gamma(r) \triangleq \{\langle\langle f,\, i\rangle,\, \langle\overline{f},\, \overline{i}\rangle\rangle \mid i \cap \overline{i} \neq \emptyset \wedge \forall t \in i \cap \overline{i}\,.\, \langle f(t),\, \overline{f}(t)\rangle \in r(t)\} \tag{30}$$

(which is said to be total when $i = \overline{i}$ e.g. for homomorphic abstractions or well-nested when $i \subseteq \overline{i}$). Define

$$\alpha(R) \triangleq \boldsymbol{\lambda}\, t \bullet \{\langle f(t),\, \overline{f}(t)\rangle \mid \exists i, \overline{i}\,.\, t \in i \cap \overline{i} \wedge \langle\langle f,\, i\rangle,\, \langle\overline{f},\, \overline{i}\rangle\rangle \in R\} \tag{31}$$

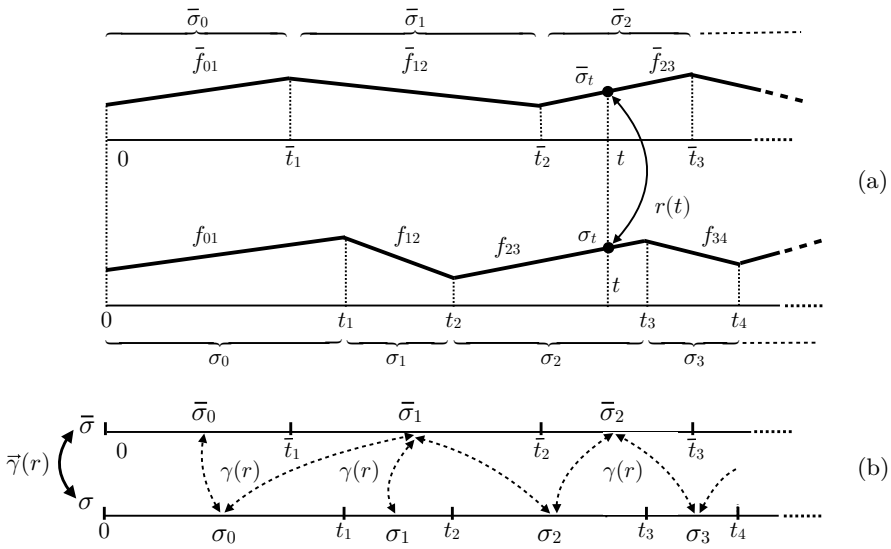


Fig. 1. Relations r between states in (a), and relations $\gamma(r)$ between configurations and $\vec{\gamma}(r)$ between trajectories in (b)

Define the set of all relations between overlapping configurations as

$$\mathsf{R_C} \triangleq \{R \in \wp(\mathsf{C} \times (\mathsf{C} \cup \mathsf{cl}(\mathsf{C}))) \mid \forall \langle\langle f,\, i\rangle,\, \langle \overline{f},\, \overline{i}\rangle\rangle \in R \,.\, i \cap \overline{i} \neq \emptyset\} \tag{32}$$

We have a Galois isomorphism

$$\langle \mathsf{R_C},\, \subseteq\rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \mathbb{R}_{\geqslant 0} \to \wp(\mathsf{S} \times \mathsf{S}),\, \dot{\subseteq}\rangle \tag{33}$$

where $\dot{\subseteq}$ is the pointwise extension of set inclusion $\subseteq$. So when abstracting trajectories by abstraction of their configurations, we can equivalently start from a relation r between states and use the relation $\gamma(r) \in \mathsf{R_C}$ or start from a relation between configurations $R \in \mathsf{R_C}$ which induces a relation $\alpha(R)$ between states. In discrete systems, the two notions of state and configuration coincide.

6.3 Relation between trajectories

Let us also define a relation between trajectories so as to relate states of trajectories

$$\vec{\gamma}(r) \triangleq \{\langle \sigma,\, \overline{\sigma}\rangle \mid \forall t \in [0, \min(\lfloor\!\lfloor \sigma \rfloor\!\rfloor, \lfloor\!\lfloor \overline{\sigma} \rfloor\!\rfloor)[\,.\, \langle \sigma_t,\, \overline{\sigma}_t\rangle \in r(t)\} \tag{34}$$

as illustrated in figure (1.a) Notice that in the definition (34) of related trajectories, we do not use the relation $\gamma(r)$ in (30) between configurations since the states of the configurations with the same ranks in the concrete and abstract in trajectories may be unrelated while the timings are (i.e. the concrete and abstract configurations of same rank k may not even overlap in time). However, we have

the following equivalent definition using ranks of configurations in trajectories, as illustrated in figures (1.b) and (2).

$$\vec{\gamma}(r) \triangleq \vec{\gamma}_c(r) \cap \vec{\gamma}_a(r) \tag{35}$$

$$\vec{\gamma}_c(r) \triangleq \{\langle \sigma,\, \overline{\sigma}\rangle \mid \forall j < |\sigma| \,.\, (\mathsf{e}(\sigma_j) \leqslant [\![\overline{\sigma}]\!]) \Longrightarrow (\exists k < |\overline{\sigma}| \,.\, \langle \sigma_j,\, \overline{\sigma}_k\rangle \in \gamma(r))\} \tag{a}$$

$$\vec{\gamma}_a(r) \triangleq \{\langle \sigma,\, \overline{\sigma}\rangle \mid \forall k < |\overline{\sigma}| \,.\, (\mathsf{e}(\overline{\sigma}_k) \leqslant [\![\sigma]\!]) \Longrightarrow (\exists j < |\sigma| \,.\, \langle \sigma_j,\, \overline{\sigma}_k\rangle \in \gamma(r))\} \tag{b}$$

(By the isomorphism (33), there is a definition equivalent to (35) using $R \in \mathsf{R_C}$ instead of $\gamma(r)$.)

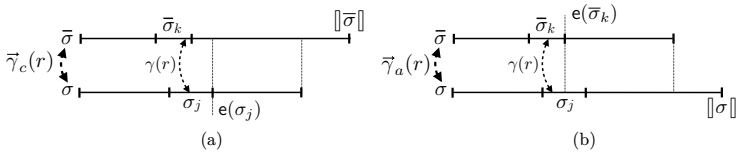


Fig. 2. Relations $\vec{\gamma}_c(r)$ and $\vec{\gamma}_a(r)$ between traces

Defining $\vec{\alpha}(\overline{R}) \triangleq \boldsymbol{\lambda}\, t \bullet \{\langle \sigma_t,\, \overline{\sigma}_t\rangle \mid \langle \sigma,\, \overline{\sigma}\rangle \in \overline{R} \wedge t \in [0, \min([\![\sigma]\!], [\![\overline{\sigma}]\!])[\}$, this is a Galois connection

$$\langle \wp(\mathsf{T}_{\mathsf{C}}^{+\infty} \times \mathsf{T}_{\overline{\mathsf{C}}}^{+\infty}),\, \subseteq\rangle \underset{\vec{\alpha}}{\overset{\vec{\gamma}}{\leftrightarrows}} \langle \mathbb{R}_{\geqslant 0} \to \wp(\mathsf{S} \times \overline{\mathsf{S}})),\, \dot{\subseteq}\rangle \tag{36}$$

6.4 Relation between hybrid trajectory semantics

The abstraction (36) is then extended to hybrid trajectory semantics through a preorder, a common one being the overapproximation in verification ($\overline{T}$ is an abstraction of T since $\overline{T}$ has more possible behaviors than T) and underapproximation in refinement (T is a refinement of $\overline{T}$ since T has less behaviors that the specification $\overline{T}$), that is

$$\begin{aligned} \vec{\gamma}(R) &\triangleq \{\langle T,\, \overline{T}\rangle \mid T \subseteq \mathsf{pre}[R]\overline{T}\} \\ &= \{\langle T,\, \overline{T}\rangle \mid \forall \sigma \in T \,.\, \exists \overline{\sigma} \in \overline{T} \,.\, \langle \sigma,\, \overline{\sigma}\rangle \in R\} \end{aligned} \tag{37}$$

Defining $\vec{\alpha}(P) \triangleq \{\langle \sigma,\, \overline{\sigma}\rangle \mid \exists \overline{T} \,.\, \langle \{\sigma\},\, \overline{T}\rangle \in P \wedge \overline{\sigma} \in \overline{T}\}$, we have the Galois connection

$$\langle \{\langle T,\, \overline{T}\rangle \in \wp(\mathsf{T}_{\mathsf{C}}^{+\infty}) \otimes \wp(\mathsf{T}_{\overline{\mathsf{C}}}^{+\infty}) \mid \overline{T} = \emptyset \Longrightarrow T = \emptyset\},\, \supseteq\rangle \underset{\vec{\alpha}}{\overset{\vec{\gamma}}{\leftrightarrows}} \langle \wp(\mathsf{T}_{\mathsf{C}}^{+\infty} \times \mathsf{T}_{\overline{\mathsf{C}}}^{+\infty}),\, \supseteq\rangle \tag{38}$$

where by (37), (4), and (8), the concrete domain is a tensor product.

Example 4 (The water tank automaton is a state-based refinement of the specification). Le us define the state-based relation

$$r^{(39)}(t) \triangleq \{\langle\langle v,\, x,\, y\rangle,\, \langle v,\, y\rangle\rangle \mid v \in \{\mathit{shut}, \mathit{open}\} \wedge x, y \in \mathbb{R}\} \tag{39}$$

between states (19) and (25) of the water tank specification and automaton.

This induces a relation (30) between configurations, as follows.

$$\gamma(r^{(39)}) = \{\langle\langle \boldsymbol{\lambda}\, t \bullet \langle v(t),\, x(t),\, y(t)\rangle,\, i\rangle,\, \langle \boldsymbol{\lambda}\, t \bullet \langle \overline{v}(t),\, \overline{y}(t)\rangle,\, \overline{i}\rangle\rangle \mid i \cap \overline{i} \neq \emptyset \wedge \quad (40)$$
$$\forall t \in i \cap \overline{i}\ .\ v(t) = \overline{v}(t) \in \{\mathit{shut}, \mathit{open}\} \wedge y(t) = \overline{y}(t)\}$$

i.e. at any time in overlapping configurations, the water height and the state of the valve coincide. This induces a relation (35) between trajectories, as follows.

$$\vec{\gamma}(r^{(39)}) = \mathsf{let}\ \rho(c, \overline{c}) \triangleq \exists v, x, y, i, \overline{v}, \overline{y}, \overline{i}\ .\ c = \langle \boldsymbol{\lambda}\, t \bullet \langle v(t),\, x(t),\, y(t)\rangle,\, i\rangle \wedge \quad (41)$$
$$\overline{c} = \langle \boldsymbol{\lambda}\, t \bullet \langle \overline{v}(t),\, \overline{y}(t)\rangle,\, i\rangle \wedge i \cap \overline{i} \neq \emptyset \wedge \forall t \in i\ \cap i\ .\ v(t) = \overline{v}(t) \wedge$$
$$y(t) = \overline{y}(t)\ \mathsf{in}$$
$$\{\langle \sigma,\, \overline{\sigma}\rangle \mid \forall j < |\sigma|\ .\ (\mathsf{e}(\sigma_j) \leqslant [\![\overline{\sigma}]\!]) \Longrightarrow (\exists k < |\overline{\sigma}|\ .\ \rho(\sigma_j, \overline{\sigma}_k) \wedge$$
$$\forall k < |\overline{\sigma}|\ .\ (\mathsf{e}(\overline{\sigma}_k) \leqslant [\![\sigma]\!]) \Longrightarrow (\exists j < |\sigma|\ .\ \rho(\sigma_j, \overline{\sigma}_k)\}$$

Let us prove that the hybrid semantics $[\![\tau^3]\!]$ (25) of the water tank automaton of example 3 is a state based refinement of the water tank specification $\mathcal{S}^2$ of example 2 for $r^{(39)}$ in (39) (denoted $r^{(39)}$ to avoid confusions), meaning that

$$\langle [\![\tau^3]\!],\, \mathcal{S}^2\rangle \in \vec{\gamma}(\vec{\gamma}(r^{(39)}))$$

or equivalently

$$\forall \sigma \in [\![\tau^3]\!]\ .\ \exists \overline{\sigma}\ .\ P(\overline{\sigma}) \wedge \forall t \geqslant 0\ .\ \sigma(t).y = \overline{\sigma}(t).y \wedge \sigma(t).v = \overline{\sigma}(t).v \quad (42)$$

By definition (20) of P, we have to show that $\forall t \in \mathbb{R}_{\geqslant 0}\ .\ 0 \leqslant \sigma(t).y \leqslant 3$. In a shut configuration of $\mathsf{C}^{\mathit{shut}}$, $y(t) = 0$ at the beginning, y evolves as the same rate as x, and $x(t)$ is bounded by 3 so that that $y(t)$ is also bounded by 3. By definition of initial configurations C^0, any trajectory of $[\![\tau^3]\!]$ starts with a *shut* configuration, and so, by definition (25) of the transition relation τ^3, any open configuration of $\mathsf{C}^{\mathit{open}}$ follows a *shut* configuration. At the end t of this *shut* configuration, and so at the beginning t of the following *open* configuration, we have shown that $\sigma(t).y \leqslant 3$. In the *open* configuration, y decreases by $\dot{y} = -2$ and remains positive, so the invariant holds.

Moreover, we must show that if the valve remains opened, then y decreases. If $\forall t \in [t_1, t_2]\ .\ \sigma(t).v = \mathit{open}$ then t is within an open configuration, so $\dot{y} = -2$ implies that y decreases between t_1 and t_2. Similarly, if $\forall t \in [t_1, t_2]\ .\ \sigma(t).v = \mathit{shut}$ then t is within a shut configuration, so $\dot{y} = 1$ implies that y increases.

Finally, if at some point t of time, $y(t) = 0$ then if we are in an *open* configuration, the system instantaneously moves to a *shut* configuration which last at least ζ by the nonzeno hypothesis, and so, by $\dot{y} = 1$, we have $\sigma(t+\zeta).y > 0$. □

7 Transition-based hybrid trajectory semantics abstraction

Reasonings on trajectories is often considered difficult and reasonings involving only one computation step at a time are preferred. An example is Turing/Naur/Floyd/Hoare invariance proof method where verification conditions involve only one computation step at a time.

So we assume that the concrete and abstract semantics are generated by transitions systems $\langle \mathsf{C},\ \mathsf{C}^0,\ \tau\rangle$ and $\langle \overline{\mathsf{C}},\ \overline{\mathsf{C}}^0,\ \overline{\tau}\rangle$, that is $T = [\![\tau]\!]$ and $\overline{T} = [\![\overline{\tau}]\!]$, and, given a relation (29) between states, we study relations between transition relations which enable us to define relations (34) between trajectories hence relations (38) between trajectory semantics. In the literature of abstraction of discrete transition systems, basic state and transition-based abstractions are homomorphisms, simulations, bisimulations, and preservations with progress, which we extend to hybrid transition systems, adding discretization.

7.1 Homomorphisms

Homomorphisms are the case when relation r in (29) is given by a function $h(t) \in \mathsf{S} \to \overline{\mathsf{S}}$ at time t. Following (30), the homomorphism is extended to configurations as

$$\alpha_h(\langle f,\ i\rangle) \triangleq \langle h \circ f,\ i\rangle) \tag{43}$$

The function h is composed with the flow and the timings remain the same. The extension to trajectories is

$$\alpha_h(\langle \sigma_i,\ i \in [0, |\sigma|[\rangle) \triangleq \langle \alpha_h(\sigma_i),\ i \in [0, |\sigma|[\rangle \tag{44}$$

and to trajectory semantics

$$\alpha_h(T) \triangleq \{\alpha_h(\sigma) \mid \sigma \in T\} \tag{45}$$

which, by (5), is a Galois connection

$$\langle \wp(\mathsf{T}_{\mathsf{C}}^{+\infty}),\ \subseteq\rangle \underset{\alpha_h}{\overset{\gamma_h}{\leftrightarrows}} \langle \wp(\mathsf{T}_{\overline{\mathsf{C}}}^{+\infty}),\ \subseteq\rangle \tag{46}$$

The homomorphic abstraction of a transition system is

$$\alpha_h(\langle \mathsf{C},\ \mathsf{C}^0,\ \tau\rangle) \triangleq \langle \{h(c) \mid c \in \mathsf{C}\},\ \{h(c) \mid c \in \mathsf{C}^0\},\ \{\langle h(c),\ h(c')\rangle \mid \langle c,\ c'\rangle \in \tau\}\rangle \tag{47}$$

For brevity, we write $\alpha_h(\tau)$ for $\alpha_h(\langle \mathsf{C},\ \mathsf{C}^0,\ \tau\rangle)$. The homomorphic abstraction of the trajectory semantics generated by the concrete transition system is the abstract trajectory semantics generated by the homomorphic abstraction of the concrete transition system

Theorem 1.

$$\alpha_h([\![\tau]\!]) = [\![\alpha_h(\tau)]\!] \tag{48}$$

The *verification* of a property of an hybrid system $[\![\tau]\!]$ defined by a transition relation τ can be done in the abstract, as follows.

Theorem 2. *For any abstract hybrid trajectory property* $\overline{P} \in \wp(\mathsf{T}_{\overline{\mathsf{C}}}^{+\infty})$,

$$\frac{\alpha_h(\tau) \subseteq \overline{\tau},\quad (26),\quad [\![\overline{\tau}]\!] \subseteq \overline{P}}{[\![\tau]\!] \subseteq \gamma_h(\overline{P})} \tag{49}$$

i.e. a sound abstract small-step semantics $\overline{\tau}$ overapproximating the concrete semantics τ is designed so that the concretization $\gamma_h(\overline{P})$ of its trace properties $\overline{P}$ holds for the concrete semantics $[\![\tau]\!]$.

Given a specification in the form of an abstract transition system $\overline{\tau}$, *refinement* consists in designing a concrete transition system τ such that $[\![\tau]\!] \subseteq \gamma_h(\{[\![\overline{\tau}]\!]\})$. By induction principle (49) where $\overline{P} = \{[\![\overline{\tau}]\!]\}$, it is sufficient to ensure that $\alpha_h(\tau) \subseteq \overline{\tau}$ and the blocking condition (26).

Finally, the homomorphic abstraction is preserved by discretization (27).

Theorem 3.

$$\alpha_\delta(\alpha_h(T)) = \alpha_h(\alpha_\delta(T)) \tag{50}$$

In conclusion of this section 7.1, homomorphic abstractions are very simple since they compose (because $h(t) \in \mathsf{S} \to \overline{\mathsf{S}}$ and $\overline{h}(t) \in \overline{\mathsf{S}} \to \overline{\overline{\mathsf{S}}}$ implies $\overline{h}(t) \circ h(t) \in \mathsf{S} \to \overline{\overline{\mathsf{S}}}$), there is a unique best abstract homomorphic abstract hybrid semantics (by (46)), they extend from hybrid transition systems to hybrid semantics (by theorem 1), allow proofs of trajectory properties by abstraction (by theorem 2), and are preserved by discretization (by theorem 3). Homomorphic abstractions seem to be almost the only ones considered in model-checking [5, pp. 499–504].

However, homomorphic abstractions are very restrictive in that the relation between flows is deterministic and the concrete and abstract timelines must be exactly the same[3].

7.2 Simulations

Simulations were introduced by Robin Milner [26] to relate discrete transitions systems hence, implicitly, their trace semantics or abstractions of these trace semantics. They have been used for program verification and refinement. Notice that Robin Milner originally used (bi)simulation relations to abstract reachability/invariance properties for which reasoning on transitions and their reflexive closure is sound and complete. So there was no need to consider (bi)similar traces.

Various extensions to continuous and hybrid systems have been proposed such as [23,3,31,13,14,25,15,12,15,16,35,6,36,9] among others. In contrast with this previous work, our definition of (bi)simulation takes into account the fact that concrete and abstract trajectories may have different durations and not necessarily comparable timelines for mode changes.

[3] One could argue that the time-evolution low abstraction of (17) applied to the concrete and abstract trajectories would solve the problem of having the same timeline by merging the trajectories into a single configuration, but then the original timelines are hidden in the flow functions, which does not make time-dependent reasonings simpler.

Definition of asynchronous hybrid simulations. A relation $R \in \wp(\mathsf{C} \times \overline{\mathsf{C}})$ between concrete and abstract configurations (which can be the extension (30) $R = \gamma(r)$ of the timed relation r between states in (29)) is a *hybrid simulation* between the transition relations τ and $\overline{\tau}$ if and only if

$$\begin{aligned}\forall c, \overline{c}, c' \,.\, \exists \overline{c}' \,.\, (\langle c,\, \overline{c}\rangle \in R \wedge (\langle c,\, c'\rangle \in \tau \vee c' = \varepsilon)) \Longrightarrow \\ ((\langle \overline{c},\, \overline{c}'\rangle \in \overline{\tau} \vee \overline{c}' = \varepsilon) \wedge \langle c \fatsemi c'(\!|\min(\mathsf{b}(c'), \mathsf{b}(\overline{c}')), \min(\mathsf{e}(c'), \mathsf{e}(\overline{c}'))|\!), \\ \overline{c} \fatsemi \overline{c}'(\!|\min(\mathsf{b}(c'), \mathsf{b}(\overline{c}')), \min(\mathsf{e}(c'), \mathsf{e}(\overline{c}'))|\!)\rangle \in R)\end{aligned} \tag{51}$$

To simplify notations, we write $c' = \varepsilon$ for $\mathsf{e}(\overline{c}') \leqslant \mathsf{e}(c) \wedge c' = \varepsilon$ and similarly $\overline{c}' = \varepsilon$ stands for $\mathsf{e}(c') \leqslant \mathsf{e}(\overline{c}) \wedge \overline{c}' = \varepsilon$, see figure 3.

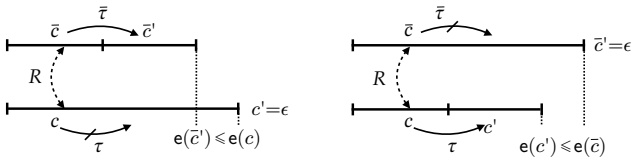


Fig. 3. Empty successor configurations

As shown in figure 4, this definition of an *asynchronous simulation* takes into account the fact that the concrete and abstract configurations may correspond to different timelines. Concrete and abstract configurations $\langle c, \overline{c}\rangle \in R$ are related

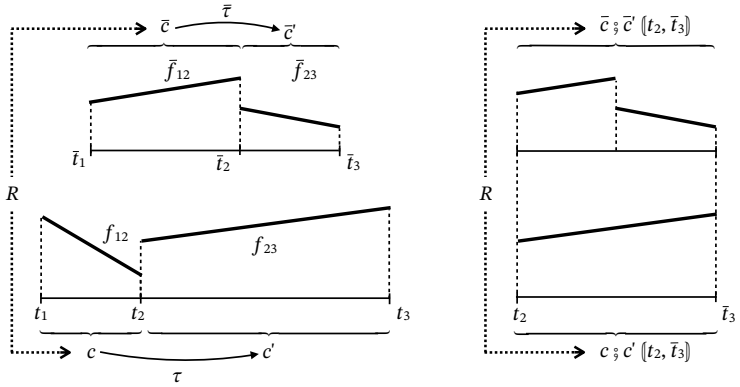


Fig. 4. Asynchronous hybrid simulation

and there is a concrete transition $\langle c,\, c'\rangle \in \tau$ from c to c' so there must exists an abstract transition $\langle \overline{c},\, \overline{c}'\rangle \in \overline{\tau}$ such that c' and $\overline{c}'$ are related. But since c' and $\overline{c}'$ may have different timings, one of them is extended in the past by the previous configuration ($\overline{c}'$ extended to $t_2 = \min(t_2, \overline{t}_2) = \min(\mathsf{b}(c'), \mathsf{b}(\overline{c}'))$ using the previous $\overline{c}$ in figure 4) while one of them, maybe the same, is truncated in the future to the first terminating configuration (c' truncated to $\overline{t}_3 = \min(t_3, \overline{t}_3) = \min(\mathsf{e}(c'), \mathsf{e}(\overline{c}'))$ in figure 4).

The simulation $R = \{\langle c, \alpha_h(c)\rangle \mid c \in \mathsf{C}\}$ may be the homomorphic abstraction (43) which would enforce the concrete and abstract timings to be the same. More generally, the simulation R may be many-to-many. For example the concrete states can be equipped with a distance and R would ensure that, at each time instant, the concrete state is in a ball around the abstract state [9,16], the size of the ball evolving over time (this would, for example, account for cumulated rounding errors when the abstract states are reals and the refined concrete states are floats).

Another particular case is that of a *synchronous simulation* of well-nested configurations when concrete timelines are subdivisions of the abstract timelines (that is, if $\langle f,\ i\rangle \in \mathsf{C}$, $\langle \overline{f},\ \overline{i}\rangle \in \overline{\mathsf{C}}$, and $i \cap \overline{i} \neq \emptyset$ then $\mathsf{b}(\overline{i}) \leqslant \mathsf{b}(i) < \mathsf{e}(i) \leqslant \mathsf{e}(\overline{i})$). If moreover all configurations have at least one successor, there are no blocking configurations so that (51) becomes

$$\begin{aligned}\forall c, \overline{c}, c' \,.\, \exists \overline{c}' \,.\, (\langle c,\ \overline{c}\rangle \in R \wedge (\langle c,\ c'\rangle \in \tau)) \Longrightarrow \\ ((\langle \overline{c},\ \overline{c}'\rangle \in \overline{\tau} \vee \overline{c}' = \varepsilon) \wedge \langle c',\ \overline{c}'(\!|\mathsf{b}(c'), \mathsf{e}(c')|\!)\rangle \in R)\end{aligned} \tag{52}$$

Example 5 (Change of variables). Let $\langle c_i,\ i \in [0, |c|[\rangle$ be a concrete semantics with concrete configurations $c_i = \langle \boldsymbol{\lambda}\, t \bullet f_i(t - t_i^\ell),\ [t_i^\ell, t_i^h[\rangle$ where $f_i(t)$ is given by the Cauchy-Euler implicit ordinary differential (ODE) equation $t^2 f_i''(t) + a_i t f_i'(t) + b_i f_i(t) = 0$. Under appropriate continuity hypotheses, a classic resolution method [30, ch.19, p. 170] consists in applying the change of variable $t = \ln(\overline{t})$, that is $\overline{t} = e^t$ to get $\varphi_i(\overline{t})$ solution of $\varphi_i''(\overline{t}) + (a_i - 1)\varphi_i'(\overline{t}) + b_i\varphi_i(\overline{t}) = 0$ which is a linear ODE solved via its characteristic polynomial. Let the abstract hybrid semantics be $\langle \overline{c}_i,\ i \in [0, |c|[\rangle$ with abstract configurations $\overline{c}_i = \langle \boldsymbol{\lambda}\, \overline{t} \bullet \varphi_i(\overline{t} - e^{t_i^\ell}),\ [e^{t_i^\ell}, e^{t_i^h}[\rangle$. This is a hybrid simulation (indeed a bisimulation) $\gamma(r)$ for $r(t) = \{\langle f_i(t - t_i^\ell),\ \varphi_i(e^t - e^{t_i^\ell})\rangle \mid i \in [0, |c|[\wedge\, t \in [t_i^\ell, t_i^h[\}$. □

Example 6. Continuing the water tank automaton example 3, we refine the tank specification by taking some time ϵ to close (in *off* configuration) and open (in *on* configuration) the valve while in *shut* mode. We assume $\epsilon > \zeta$ to ensure that the duration of the valve opening and closing is not infinitesimal. The water inflow $\dot{y}$ is increased to compensate for this delay. We assume that ϵ is large enough for the valve opening and closing to be mechanically feasible in this period of time. We assume that ϵ is small enough so that the duration of the *shut* configuration in (25) is much larger than 2ϵ. In particular it must be chosen so that the increase of the inflow is physically possible.

The *open* mode is unchanged, as shown in figure (5.a). A formal definition of example 6 is given in the ArXiv version.

The relation $R^{(53)}$ between configurations of concrete trajectories in (5.a) and the configurations of abstract trajectories in (5.b) is the following.

$$\begin{aligned}R^{(53)} \triangleq \big\{\langle\langle \boldsymbol{\lambda}\, t \bullet \langle m_t,\ x_t,\ y_t\rangle,\ [t_1, t_2[\rangle,\ \langle \boldsymbol{\lambda}\, t \bullet \langle \overline{m}_t,\ \overline{x}_t,\ \overline{y}_t\rangle,\ [\overline{t}_1, \overline{t}_2[\rangle\rangle \mid \\ P^{(53)}(m, x, y, t_1, t_2, \overline{m}, \overline{x}, \overline{y}, \overline{t}_1, \overline{t}_2)\big\}\end{aligned} \tag{53}$$

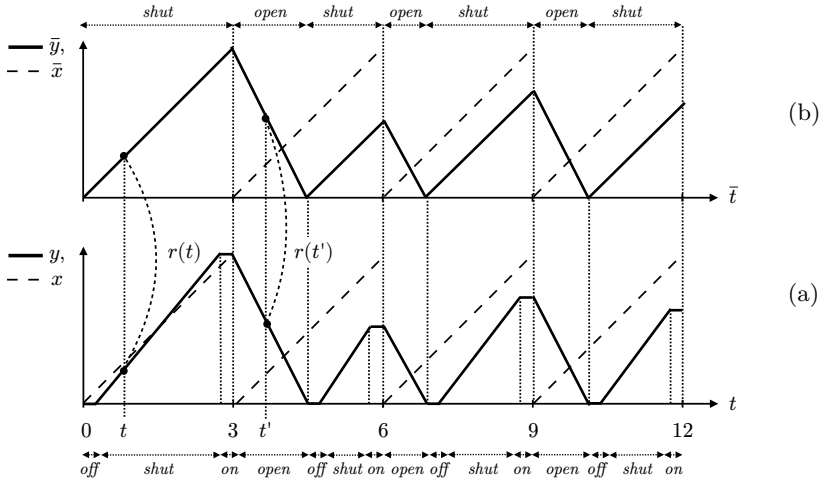


Fig. 5. Concrete (a) and abstract (b) tank trajectories

$$
\begin{aligned}
&P^{(53)}(m,x,y,t_1,t_2,\overline{m},\overline{x},\overline{y},\overline{t}_1,\overline{t}_2) \triangleq \forall t \in [\overline{t}_1,\overline{t}_2[\ .\ x_t = \overline{x}_t \wedge \qquad (54)\\
&\big((\overline{m}_t = \mathit{shut} \wedge\\
&\quad ((m_t = \mathit{off} \wedge y = 0 \wedge \overline{y}_t = t - t_1 \wedge t_1 = \overline{t}_1 \wedge t_2 = \overline{t}_1 + \epsilon)\\
&\qquad \vee (m_t = \mathit{shut} \wedge t_1 = \overline{t}_1 + \epsilon \wedge t_2 = \overline{t}_2 - \epsilon \wedge \overline{y}_t = y_t + \epsilon\Big(1 - \frac{2(t_2 - t)}{t_2 - t_1}\Big))\\
&\qquad \vee (m_t = \mathit{on} \wedge t_1 = \overline{t}_2 - \epsilon \wedge t_2 = \overline{t}_2 \wedge y_t = \overline{y}_t + \epsilon\Big(\frac{\overline{t}_2 - t}{\overline{t}_2 - t_2}\Big))))\\
&\vee (\overline{m}_t = m_t = \mathit{open} \wedge y_t = \overline{y_t} \wedge t_1 = \overline{t}_1 \wedge t_2 = \overline{t}_2)\big)
\end{aligned}
$$

By the Galois isomorphism (33), the relation $R^{(53)}$ between configurations defines the relation $r^{(53)}$ between states as a function of the time.

$$
\begin{aligned}
r^{(53)} \triangleq \big\{\langle\langle m_t,\ x_t,\ y_t\rangle,\ \langle\overline{m}_t,\ \overline{x}_t,\ \overline{y}_t\rangle\rangle \ \big|\ &\exists [t_1,t_2[\subseteq [\overline{t}_1,\overline{t}_2[\ .\ t \in [t_1,t_2[\qquad (55)\\
&\wedge P^{(53)}(m,x,y,t_1,t_2,\overline{m},\overline{x},\overline{y},\overline{t}_1,\overline{t}_2)\big\}
\end{aligned}
$$

$R^{(53)}$, that is $\gamma(r^{(53)})$, is a synchronous simulation (52), where in *shut* mode the concrete level is within ϵ of the abstract water level while in *open* mode they are the same. □

Trace abstraction by asynchronous hybrid simulations. Our objective is now to generalize the results of section 7.1 on homomorphisms to (weaker ones for) simulations. Similar to (48) for homomorphic abstractions, simulations induce related hybrid trajectory semantics.

Theorem 4. *If the timed relation r between states in* (29) *is such that its extension $\gamma(r)$ to configurations in* (30) *is a simulation* (51) *between $\langle \mathsf{C},\ \mathsf{C}^0,\ \tau\rangle$ and $\langle \overline{\mathsf{C}},\ \overline{\mathsf{C}}^0,\ \overline{\tau}\rangle$ satisfying the* initialization hypothesis

$$\forall c \in \mathsf{C}^0 \,.\, \exists \overline{c} \in \overline{\mathsf{C}}^0 \,.\, \langle c,\, \overline{c}\rangle \in \gamma(r) \tag{56}$$

then $\langle [\![\tau]\!],\, [\![\overline{\tau}]\!]\rangle \in \vec{\gamma}(\vec{\gamma}_c(r))$. *If moreover, the* blocking hypothesis

$$\forall c, \overline{c} \,.\, (\langle c,\, \overline{c}\rangle \in \gamma(r) \wedge \forall c' \,.\, \langle c,\, c'\rangle \notin \tau) \Longrightarrow (\forall \overline{c}' \,.\, \langle \overline{c},\, \overline{c}'\rangle \notin \overline{\tau}) \tag{57}$$

holds then

$$\langle [\![\tau]\!],\, [\![\overline{\tau}]\!]\rangle \in \vec{\gamma}(\vec{\gamma}(r)) \tag{58}$$

(*that is, by* (37), $\forall \sigma \in [\![\tau]\!] \,.\, \exists \overline{\sigma} \in [\![\overline{\tau}]\!] \,.\, \langle \sigma,\, \overline{\sigma}\rangle \in \vec{\gamma}(r)$ *and so, by* (34), $\forall t \in [0, \min([\![\sigma]\!], [\![\overline{\sigma}]\!])[\cap \mathsf{dom}(r) \,.\, \langle \sigma_t,\, \overline{\sigma}_t\rangle \in r(t)$).

Example 7. Continuing the water tank automaton in examples 3 and 6, $R^{(53)}$ is a synchronous simulation (52). So the hybrid semantics of example 6 is a simulation of (58) of the hybrid semantics $[\![\tau^3]\!]$ of example 3. □

Simulations are abstractions. Observe that theorem 4 implies that hybrid simulations are Galois connection-based abstractions (38).

Compositionality of simulations. The composition of simulations may not, in general, correspond to the composition of their timed relations between states (defined as $(r_1 \circ r_2)(t) = r_1(t) \circ r_2(t)$). This is because the intermediate trajectories may be shorter that those in the composition, as shown in figure 6.

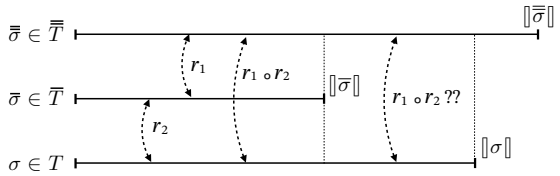


Fig. 6. Non-composition due to short intermediate trajectory duration

Another problem is that of interval mismatches where the intervals along trajectories thus leaving some states time-unrelate in the composition, see figure 7.

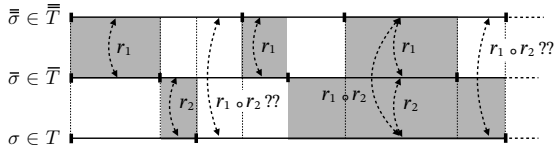


Fig. 7. Non-nested intervals

A sufficient condition for compositionally is that the involved trajectories be all infinite with well nested interval, meaning

$$\begin{aligned}\forall \langle \langle f_j,\, i_j\rangle,\, j \in \mathbb{N}\rangle \in T \,.\, \forall \langle \langle \overline{f}_k,\, \overline{i}_k\rangle,\, k \in \mathbb{N}\rangle \in \overline{T} \,.\quad & \\ \forall j, k \in \mathbb{N} \,.\, (i_j \cap \overline{i}_k \neq \emptyset) \Longrightarrow (i_j \subseteq \overline{i}_k) & \end{aligned} \tag{59}$$

Theorem 5. *If* $T \in \mathsf{T}_{\mathsf{C}}^{\infty}$, $\overline{T} \in \mathsf{T}_{\overline{\mathsf{C}}}^{\infty}$, $\overline{\overline{T}} \in \mathsf{T}_{\overline{\overline{\mathsf{C}}}}^{\infty}$ *are well-nested,* $\langle T, \overline{T}\rangle \in \vec{\overline{\gamma}}(\vec{\gamma}_c(r_1))$ *and* $\langle \overline{T}, \overline{\overline{T}}\rangle \in \vec{\overline{\gamma}}(\vec{\gamma}_c(r_2))$ *then* $\langle T, \overline{\overline{T}}\rangle \in \vec{\overline{\gamma}}(\vec{\gamma}_c(r_1 \circ r_2))$.

Example 8 (Composition of the water tank simulations). Continuing the water tank specification in example 2, automaton in example 3, and implementation in example 6, the hybrid trajectory semantics are well-nested according to (59). The hybrid semantics $[\![\tau^6]\!]$ of example 6 is a simulation of the hybrid semantics $[\![\tau^3]\!]$ of example 3 by $r^{(53)}$, which itself is a simulation of the specification $\mathcal{S}^2$ by $r^{(39)}$. So, by theorem 5, their composition $r^{(53)} \circ r^{(39)}$ holds at any time between the implementation $[\![\tau^6]\!]$ and the specification $\mathcal{S}^2$.

This may look paradoxical because if $\epsilon > \zeta$ then water in the implementation will remain at the zero level longer than prescribed by the specification (20.d).

However, this is not an anomaly since the composition is

$$r^{(53)} \circ r^{(39)} \triangleq \{\langle\langle m_t,\, x_t,\, y_t\rangle,\, \langle \overline{m}_t,\, \overline{y}_t\rangle\rangle \mid \exists [t_1, t_2[\subseteq [\bar{t}_1, \bar{t}_2[\,.\, t \in [t_1, t_2[\wedge \quad (60)$$
$$P^{(53)}(m_t, x_t, y_t, t_1, t_2, \overline{m}_t, x_t, \overline{y}_t, \bar{t}_1, \bar{t}_2)\}$$

By definition (53), this expresses that the height $\overline{y}_t$ of the water in the specification when the valve is *off* for ϵ units of time is equal to $t - t_1 = t - \bar{t}_1$, not to the level of water $y_t = 0$ in the implementation. So, although each simulation $r^{(39)}$ and $r^{(53)}$ is a satisfactory specification, their composition is an incomplete refinement of the expected water tank behavior. □

Greatest simulation. (51) can be rewritten as

$$R \subseteq F^s_{\tau,\overline{\tau}}(R) \qquad \text{with} \qquad (61)$$
$$F^s_{\tau,\overline{\tau}}(R) \triangleq \{\langle c,\, \overline{c}\rangle \mid \forall c' \,.\, (\langle c,\, c'\rangle \in \tau) \Longrightarrow$$
$$(\exists \overline{c}' \,.\, \langle \overline{c},\, \overline{c}'\rangle \in \overline{\tau} \wedge\ \langle c \fatsemi c'(\!|\min(\mathsf{b}(c'), \mathsf{b}(\overline{c}')), \min(\mathsf{e}(c'), \mathsf{e}(\overline{c}'))|\!),$$
$$\overline{c} \fatsemi \overline{c}'(\!|\min(\mathsf{b}(c'), \mathsf{b}(\overline{c}')), \min(\mathsf{e}(c'), \mathsf{e}(\overline{c}'))|\!)\rangle \in R)\}$$

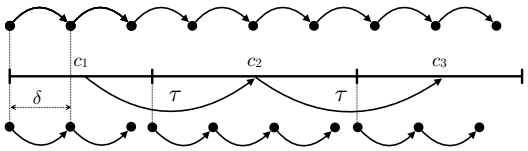
where $F^s_{\tau,\overline{\tau}}$ is increasing on the complete lattice $\langle \wp(\mathsf{C} \times \overline{\mathsf{C}}), \subseteq\rangle$ so that by Tarski's fixpoint theorem [37] there exists a greatest simulation between τ and $\overline{\tau}$, thus extending Robin Milner's classic result [27, Proposition 16, section 4.6] to hybrid simulations (and the least fixpoint is $\emptyset$).

Verification of trace properties by simulation. The homomorphic induction principle (49) can be generalized to hybrid asynchronous simulations $\vec{\overline{\gamma}}(r)$ as follows

$$\frac{\vec{\overline{\gamma}}(r) \subseteq F^s_{\tau,\overline{\tau}}(\vec{\overline{\gamma}}(r)), \quad (56), \quad (57), \quad [\![\overline{\tau}]\!] \subseteq \overline{P}}{\langle [\![\tau]\!], \overline{P}\rangle \in \vec{\overline{\gamma}}(\vec{\overline{\gamma}}(r))} \qquad (62)$$

Discretization by sampling.

Discretization of a hybrid transition system. We have defined the discretization (27) of a hybrid trajectory semantics. In general the discretization of a hybrid transition system (22) and that of the generated trajectories (23) do not coincide, as shown by the following counterexample (for which the discretization of the trajectory and that of the configurations $c_1, c_2, c_3, \ldots$ in the transition relation τ do not coincide).



trajectory discretization

transition-generated trajectory

transition configuration discretization

To solve this dependency, it is generally assumed that the start time and duration of configurations is a multiple of the discretization step

$$\forall c \in \mathsf{C} \,.\, \exists k, k' \in \mathbb{N} \,.\, \mathsf{b}(c) = k\delta < k'\delta = \mathsf{e}(c). \tag{63}$$

The relation (29) between states is time-dependent. Simply ignoring the discrete time $\langle n\delta,\ n \in \mathbb{N}\rangle$ might create circularities (see example 10 thereafter).

One solution is to incorporate the time (or at least the rank $n \in \mathbb{N}$) into states to make the relation time-independent as in classic simulations. So (27) becomes

$$\alpha_\delta(\sigma) \triangleq \langle\langle \sigma_{n\delta},\ n\rangle,\ n \in \mathbb{N} \wedge n\delta \in [0, \lbag\sigma\rbag]\rangle \tag{64}$$

and (29) becomes

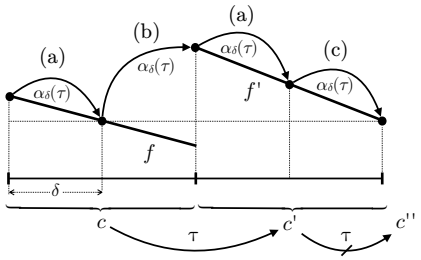
$$\alpha_\delta(r) \triangleq \{\langle\langle s,\ n\rangle,\ \langle \overline{s},\ n\rangle\rangle \mid n \in \mathbb{N} \wedge n\delta \in \mathsf{dom}(r) \wedge \langle s,\ \overline{s}\rangle \in r(n\delta)\}. \tag{65}$$

The timeful discretization of the hybrid transition system is

$$\begin{aligned}
\alpha_\delta(\tau) \triangleq \{\langle\langle s,\ n\rangle,\ \langle s',\ n+1\rangle\rangle \mid & \qquad (66)\\
(\exists c \,.\, (c \in \mathsf{C}^0 \vee \exists c' \,.\, \langle c',\ c\rangle \in \tau) \wedge & \qquad \text{(a)}\\
\mathsf{b}(c) \leqslant n\delta < (n+1)\delta < \mathsf{e}(c) \wedge s = c_{n\delta} \wedge s' = c_{(n+1)\delta}) & \\
\vee (\exists \langle c,\ c'\rangle \in \tau \,.\, (n+1)\delta = \mathsf{e}(c) \wedge s = c_{n\delta} \wedge s' = c'_{(n+1)\delta}) & \qquad \text{(b)}\\
\vee (\exists c \in \mathsf{C} \,.\, \forall c' \,.\, \langle c,\ c'\rangle \notin \tau \wedge (n+1)\delta = \mathsf{e}(c) \wedge & \qquad \text{(c)}\\
s = c_{n\delta} \wedge s' = c_{(n+1)\delta})\} & \\
\alpha_\delta(\mathsf{C}^0) \triangleq \{\langle c_0,\ 0\rangle \mid c \in \mathsf{C}^0\} & \qquad \text{(d)}
\end{aligned}$$

which is well-defined by (22) and since the durations of the configurations are assumed to be multiples of δ. (66.a) covers discrete transitions within a configuration but the last one. The last transition is either to the first state of the next configurations (66.b), or in absence of any successor configuration, to the last state of the current configuration (66.c). This condition (66.c) solves the problem of having open right time intervals in configurations by defining the last state of the last configuration of finite trajectories. This discretization applies to both concrete and abstract transition systems.

Example 9 (hybrid transition discretization). The various cases in (66) are illustrated below.



- By (66.a), the initial configuration $c = \langle f, [0, 2\delta]\rangle$ starting at time 0 of duration 2δ has an internal discrete transition $\langle\langle f(0),\ 0\rangle,\ \langle f(\delta),\ 1\rangle\rangle \in \alpha_\delta(\tau)$ between its states at times 0 and δ;
- Similarly, by (66.a), the successor configuration $c' = \langle f',\ [2\delta, 4\delta]\rangle$ starting at time 2δ of duration 2δ has an internal discrete transition $\langle\langle f'(2\delta),\ 2\rangle,\ \langle f'(3\delta),\ 3\rangle\rangle \in \alpha_\delta(\tau)$;
- By (66.b), the last discrete transition of configuration c is toward the beginning state $\langle f'(2\delta),\ 2\rangle$ of its successor configuration(s) c' (and not toward its final state $\langle f(2\delta),\ 2\rangle$);
- In contrast, by (66.c), the last discrete transition for configuration c' which has no possible successor by τ is toward its final state $\langle f'(4\delta),\ 4\rangle$.

Observe that $f(\delta) = f'(4\delta)$ but they are distinguished by incorporating the rank n of discrete times $n\delta$. □

Example 10 (timeful and timeless abstraction). Consider $\mathsf{S} = \{s\}$, $\mathsf{C} = \mathsf{C}^0 = \{c\}$, $\tau = \emptyset$ where $c = \langle f,\ [0, 2]\rangle$ with $\forall t \in [0, 2]\ .\ f(t) = s$, and $\delta = 1$. We have $[\![\tau]\!] = \{c\}$ and $\alpha_\delta([\![\tau]\!]) = \{\langle s,\ 0\rangle\langle s,\ 1\rangle\}$ as well as $\alpha_\delta(\tau) = \emptyset$ and $[\![\alpha_\delta(\tau)]\!] = \{\langle s,\ 0\rangle\langle s,\ 1\rangle\}$, that is, (67).

Ignoring time, we would have $\widetilde{\alpha}_\delta([\![\tau]\!]) = \{ss\}$ while the transition abstraction $\widetilde{\alpha}_\delta(\tau) = \{\langle s,\ s\rangle\}$ yields a circularity so that $[\![\widetilde{\alpha}_\delta(\tau)]\!] = s^+|s^\infty$ and in general $\widetilde{\alpha}_\delta([\![\tau]\!]) \subseteq [\![\widetilde{\alpha}_\delta(\tau)]\!]$ which would be a rather imprecise overapproximation. □

By definition of α_δ, it follows that

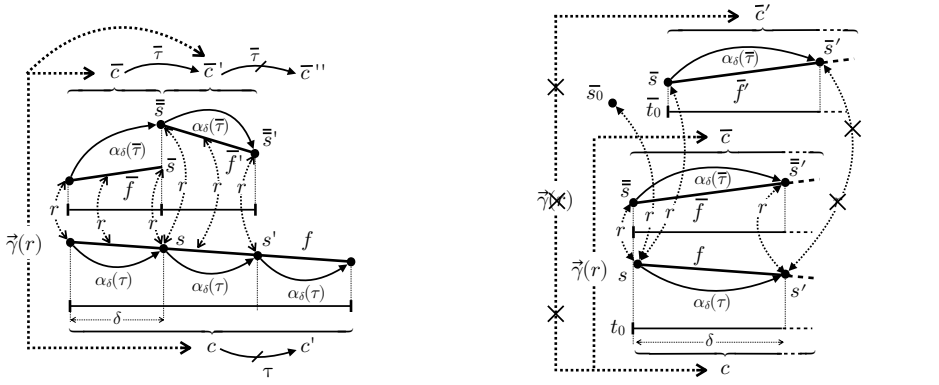
Theorem 6. *The timed discretization of the semantics is the semantics of the timeful discretized transition system, formally*

$$\alpha_\delta([\![\tau]\!]) = [\![\alpha_\delta(\tau)]\!] \tag{67}$$

Is the discretization of a hybrid simulation a discrete simulation? If the simulation of a hybrid transition system is a generalization of Robin Milner's simulation of discrete transition systems [26] there should be an abstraction of time mapping the hybrid simulation of the hybrid transition system into a discrete simulation for the discretized transition system. This is our next objective.

Sampling in (28) is a discretization. But this discretization of a hybrid simulation may not be a discrete simulation, even when configuration durations are a multiples of a base duration δ, as assumed in (63).

For a counter example, on figure (8.a), we have a hybrid simulation since states are related by r on the common interval of time of c and $\overline{c}$. But in the discretization, concrete state s has a successor while the related state $\overline{s}$ has none, so this is not a discrete simulation.



(a) blocking states $\overline{s}$ and $\overline{\overline{s}}'$

(b) related states s and isolated state $\overline{s}_0$ or s and $\overline{s}$ of unrelated configurations c and $\overline{c}'$

Fig. 8. Effects of asynchronous discretization

A second counterexample is given in figure (8.b) when $t_0 = \overline{t}_0$. We have $\langle s, \overline{s}_0\rangle \in r(\overline{t}_0)$ but $\overline{s}_0$ does not belong to any configuration and so has no successor by $\alpha_\delta(\overline{\tau})$. So the discretization of the hybrid simulation is not a discrete simulation.

A third counterexample is also given on figure (8.b) Configurations c and $\overline{c}$ are related because relation r between their states during the first period of time (end included in successor) while c and $\overline{c}'$ are not since r holds between s and $\overline{s}$ whereas it does not hold between s' and $\overline{s}'$. After discretization, the configuration $\overline{c}'$ generates a transition $\alpha_\delta(\overline{\tau})$ from $\overline{s}$ to $\overline{s}'$. Now $\langle s, \overline{s}\rangle \in r^{-1} \wedge \langle s, s'\rangle \in \alpha_\delta(\tau)$ but the only successor $\overline{s}'$ of $\overline{s}$ by $\alpha_\delta(\overline{\tau})$ is not related to s'. So the discretization of the hybrid simulation is not a discrete simulation.

Moreover, the relation r in (29) is the partial function of the time, whereas its abstraction $\alpha_\delta(r)^{-1}$ in Robin Milner's simulation $\alpha_\delta(r)^{-1} \circ \alpha_\delta(\tau) \dot{\subseteq} \alpha_\delta(\overline{\tau}) \circ \alpha_\delta(r)^{-1}$ is a well-defined relation between states. So, in case r in (29) is not total, and to be compatible with Robin Milner's definition, we must assume that r is well-defined at the discretization points

$$\forall n \in \mathbb{N} \,.\, (\exists c \in \mathsf{C} \,.\, n\delta \in \mathsf{dom}(c)) \Longrightarrow (n\delta \in \mathsf{dom}(r)). \tag{68}$$

To prevent the case of isolated state $\overline{s}_0$ in (8.b), we assume that related states must come from related configurations (either initial or successor ones).

$$\forall c \in \mathsf{C}, \overline{s} \in \overline{\mathsf{S}}, n \in \mathbb{N} \,.\, (n\delta \in \mathsf{dom}(c) \cap \mathsf{dom}(r) \wedge \langle c_{n\delta},\, \overline{s}\rangle \in r(n\delta)) \Longrightarrow \\ (\exists \overline{c} \in \overline{\mathsf{C}} \,.\, (\overline{c} \in \overline{\mathsf{C}}^0 \vee \exists \overline{c}' \,.\, \langle \overline{c}',\, \overline{c}\rangle \in \overline{\tau}) \wedge n\delta \in \mathsf{dom}(\overline{c}) \wedge \overline{c}_{n\delta} = \overline{s}) \tag{69}$$

Beyond an initialization hypothesis (similar to (56)), a common hypothesis for discrete simulations is the *non-blocking condition*, which, for hybrid simulations, translates into

$$\forall c \in \mathsf{C}, \overline{c} \in \overline{\mathsf{C}} \,.\, (\exists t \in \mathsf{dom}(c) \cap \mathsf{dom}(\overline{c}) \cap \mathsf{dom}(r) \,.\, \langle c_t,\, \overline{c}_t\rangle \in r(t) \wedge \\ \forall \overline{c}' \,.\, \langle \overline{c},\, \overline{c}'\rangle \notin \overline{\tau}) \Longrightarrow (\mathsf{e}(\overline{c}) = \mathsf{e}(c)) \tag{70}$$

The non-blocking condition (70) will avoid the blocking state $\overline{s}$ in figure 8, on the left, since concrete blocking configurations can only be related to abstract configurations with the same ending time.

Moreover, we request the relation r between states to be compatible with the discretization (66) of transition relations. If a concrete configuration c and an abstract one $\overline{c}$ have related states at some time t then their states must be related at any time $n\delta$ in their common time intervals, except maybe at the end of these time intervals (71.a).

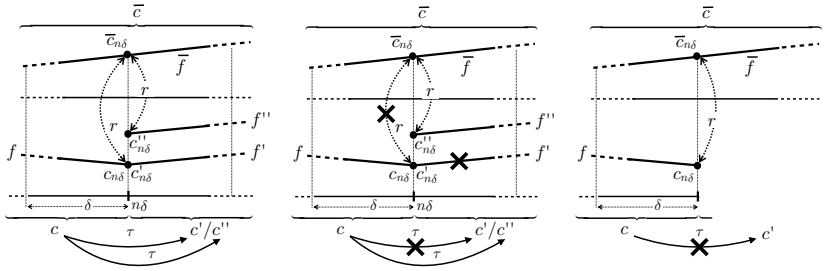
$$\forall c \in \mathsf{C}, \overline{c} \in \overline{\mathsf{C}} \,.\, (\exists t \in \mathsf{dom}(c) \cap \mathsf{dom}(\overline{c}) \cap \mathsf{dom}(r) \,.\, \langle c_t,\, \overline{c}_t\rangle \in r(t)) \Longrightarrow \tag{71}$$
$$(\forall n\delta \in (\mathsf{dom}(c) \cap \mathsf{dom}(\overline{c})) \setminus \{\mathsf{e}(c), \mathsf{e}(\overline{c})\} \,.\, \langle c_{n\delta},\, \overline{c}_{n\delta}\rangle \in r(n\delta)) \wedge \tag{a}$$
$$(\forall n \,.\, n\delta = \mathsf{e}(c) \in \mathsf{dom}(\overline{c})) \Longrightarrow \tag{b}$$
$$(\forall c' \,.\, (\langle c,\, c'\rangle \in \tau) \Longrightarrow (\langle c'_{n\delta},\, \overline{c}_{n\delta}\rangle \in r(n\delta))) \wedge \tag{b.1}$$
$$((\forall c' \,.\, \langle c,\, c'\rangle \in \tau \Longrightarrow c_{n\delta} \neq c'_{n\delta}) \Longrightarrow (\langle c_{n\delta},\, \overline{c}_{n\delta}\rangle \notin r(n\delta)))) \wedge \tag{b.2}$$
$$((\forall c' \,.\, \langle c,\, c'\rangle \notin \tau) \Longrightarrow (\langle c_{n\delta},\, \overline{c}_{n\delta}\rangle \in r(n\delta))) \wedge \tag{b.3}$$
$$(\forall n \,.\, (n\delta = \mathsf{e}(\overline{c}) \in \mathsf{dom}(c)) \Longrightarrow \tag{c}$$
$$(\forall \overline{c}' \,.\, (\langle \overline{c},\, \overline{c}'\rangle \in \overline{\tau}) \Longrightarrow (\langle c_{n\delta},\, \overline{c}'_{n\delta}\rangle \in r(n\delta))) \wedge \tag{c.1}$$
$$((\forall \overline{c}' \,.\, \langle \overline{c},\, \overline{c}'\rangle \in \overline{\tau} \Longrightarrow \overline{c}_{n\delta} \neq \overline{c}'_{n\delta}) \Longrightarrow (\langle c_{n\delta},\, \overline{c}_{n\delta}\rangle \notin r(n\delta))) \wedge \tag{c.2}$$
$$((\forall \overline{c}' \,.\, \langle \overline{c},\, \overline{c}'\rangle \notin \overline{\tau}) \Longrightarrow (\langle c_{n\delta},\, \overline{c}_{n\delta}\rangle \in r(n\delta)))) \tag{c.3}$$

The relations between states at the end of a concrete configuration c are illustrated in figure 9. In case (71.b.1), the state $c'_{n\delta}$ at the beginning of the next concrete configuration c' is related to the abstract state $\overline{c}_{n\delta}$ at the end of this concrete configuration c.

Case (71.b.2) states that if there is no concrete configuration c' which initial state $c'_{n\delta}$ is equal to the last state $c_{n\delta}$ of the previous configuration c then $c_{n\delta}$ should *not* be related to the abstract state $\overline{c}_{n\delta}$ at the end of this concrete configuration c.

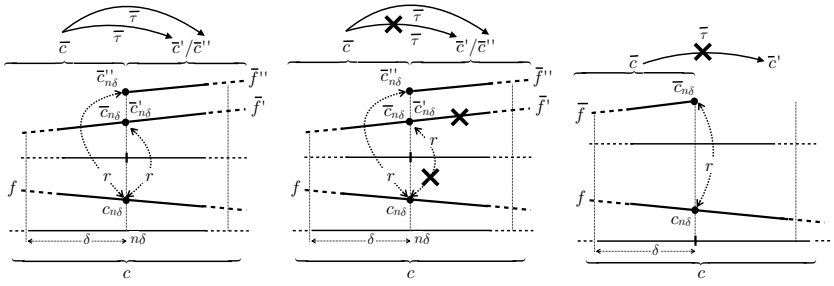
Case (71.b.3) states that if the concrete configuration c ending at time $n\delta$ has no successor then its last state should be related to the abstract state $\overline{c}_{n\delta}$ at the end of this concrete configuration c.

Cases (71.c) in figure 10 are symmetrical.



(71.b.1) (71.b.2) (71.b.3)

Fig. 9. Relation between states after discretization of concrete configuration transitions



(71.c.1) (71.c.2) (71.c.3)

Fig. 10. Relation between states after discretization of abstract configuration transitions

Then, we have the following result (72) that supports the intuition that state-based hybrid simulations $\vec{\gamma}(r)$ satisfying (51) (or equivalently (61)) are a meaningful generalization of Robin Milner discrete simulations (i.e. $R \circ t \dot{\subseteq} \overline{t} \circ R$).

Theorem 7.

$$(\vec{\gamma}(r) \subseteq F^{s}_{\tau,\overline{\tau}}(\vec{\gamma}(r)) \wedge (68) \wedge (69) \wedge (70) \wedge (71)) \Longrightarrow \alpha_\delta(r)^{-1} \circ \alpha_\delta(\tau) \dot{\subseteq} \alpha_\delta(\overline{\tau}) \circ \alpha_\delta(r)^{-1} \tag{72}$$

8 Conclusion

We have studied correspondences between trajectory semantics of hybrid systems with possibly different durations and timelines (that is different timing for mode changes), including the case where the hybrid semantics is generated by an hybrid transition system.

The abstraction relation between semantics can be derived from relations between trajectories, possibly themselves derived from relations between configurations, possibly themselves derived from timed relations between states. Such correspondences include the particular cases of homomorphisms, simulations, and discretization (as well as bisimulations, preservation, progress considered in the ArXiv version). They induce abstractions of the hybrid semantics that are Galois connections. So the abstractions between hybrid semantics defined by the correspondences between trajectories do compose.

However, contrary to the discrete case [26,32] and with the exception of homomorphic trajectory abstraction in section 7.1, the correspondences between trajectories or configurations do not necessarily compose with the correspondence between states. For example, the discretization of similar hybrid trajectories may not be similar discrete traces. The problem does not appear in Milner's original discrete definition [26] because the notion of state and configuration as well as timings do coincide. We have studied sufficient conditions for composability of trajectory correspondences to hold.

Like in the case of discrete systems [10], further abstractions of the hybrid trajectory semantics will lead to a hierarchy of semantics, verification, and static analysis methods. The most common abstraction is the reachability abstraction $\alpha(\{\langle\sigma^j_{ij},\ i^j \in [0, |\sigma^j|[\rangle \mid j \in \Delta\}) \triangleq \{\sigma^j_{ij}(t) \mid j \in \Delta \wedge i^j \in [0, |\sigma^j|[\wedge t \in [0, |\sigma^j|[\}$, see [8] for a documented and comprehensive survey.

Acknowledgements

I thank Dominique Méry for his suggestions and encouragements, "la discrétisation, c'est coton". I thank the two anonymous referees for their constructive criticisms.

References

1. Rajeev Alur, Costas Courcoubetis, Thomas A. Henzinger, and Pei-Hsin Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*, pages 209–229. Springer, 1992.
2. Rajeev Alur and David L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, 1994.
3. Rajeev Alur, Thomas A. Henzinger, Gerardo Lafferriere, and George J. Pappas. Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88(7):971–984, July 2000.
4. Rajeev Alur and P. Madhusudan. Decision problems for timed automata: A survey. In *SFM*, volume 3185 of *Lecture Notes in Computer Science*, pages 1–24. Springer, 2004.
5. Christel Baier and Joost-Pieter Katoen. *Principles of model checking.* MIT Press, 2008.
6. Richard Banach, Michael J. Butler, Shengchao Qin, Nitika Verma, and Huibiao Zhu. Core hybrid Event-B I: single hybrid Event-B machines. *Sci. Comput. Program.*, 105:92–123, 2015.

7. Paul Caspi and Nicolas Halbwachs. An approach to real time systems modeling. In *ICDCS*, pages 710–716. IEEE Computer Society, 1982.
8. Xin Chen and Sriram Sankaranarayanan. Reachability analysis for cyber-physical systems: Are we there yet? In *NFM*, volume 13260 of *Lecture Notes in Computer Science*, pages 109–130. Springer, 2022.
9. Zheng Cheng and Dominique Méry. A refinement strategy for hybrid system design with safety constraints. In *MEDI*, volume 12732 of *Lecture Notes in Computer Science*, pages 3–17. Springer, 2021.
10. Patrick Cousot. Constructive design of a hierarchy of semantics of a transition system by abstract interpretation. *Theor. Comput. Sci.*, 277(1-2):47–103, 2002.
11. Patrick Cousot. *Principles of Abstract Interpretation*. MIT Press, September 2021.
12. Alessandro D'Innocenzo, A. Agung Julius, George J. Pappas, Maria Domenica Di Benedetto, and Stefano Di Gennaro. Verification of temporal properties on hybrid automata by simulation relations. In *CDC*, pages 4039–4044. IEEE, 2007.
13. Laurent Doyen, Thomas A. Henzinger, and Jean-François Raskin. Automatic rectangular refinement of affine hybrid systems. In *FORMATS*, volume 3829 of *Lecture Notes in Computer Science*, pages 144–161. Springer, 2005.
14. Goran Frehse. On timed simulation relations for hybrid systems and compositionality. In *FORMATS*, volume 4202 of *Lecture Notes in Computer Science*, pages 200–214. Springer, 2006.
15. Antoine Girard, A. Agung Julius, and George J. Pappas. Approximate simulation relations for hybrid systems. *Discret. Event Dyn. Syst.*, 18(2):163–179, 2008.
16. Antoine Girard and George J. Pappas. Approximate bisimulation: A bridge between computer science and control theory. *Eur. J. Control*, 17(5-6):568–578, 2011.
17. Thomas A. Henzinger and Pei-Hsin Ho. A note on abstract interpretation strategies for hybrid automata. In *Hybrid Systems*, volume 999 of *Lecture Notes in Computer Science*, pages 252–264. Springer, 1994.
18. Thomas A. Henzinger, Peter W. Kopke, Anuj Puri, and Pravin Varaiya. What's decidable about hybrid automata? In *STOC*, pages 373–382. ACM, 1995.
19. Eugene Isaacson and Herbert Bishop Keller. *Analysis of Numerical Methods*. Dover, 1994.
20. Anatole Katok and Biros Hasselblatt. *Introduction to the Theory of Dynamical Systems*. Cambridge University Press, 1999.
21. Serge Lang. *Undergraduate Analysis*. Springer, 2 edition, 1997.
22. Daniel Liberzon. *Switching in Systems and Control*. Birkhäuser, 2003.
23. Nancy A. Lynch. Simulation techniques for proving properties of real-time systems. In *REX School/Symposium*, volume 803 of *Lecture Notes in Computer Science*, pages 375–424. Springer, 1993.
24. MathWorks. Simulation and model-based design. `https://www.mathworks.com/products/simulink.html`, 2022.
25. Larissa Meinicke and Ian J. Hayes. Continuous action system refinement. In *MPC*, volume 4014 of *Lecture Notes in Computer Science*, pages 316–337. Springer, 2006.
26. Robin Milner. An algebraic definition of simulation between programs. In *Proceedings IJCAI 1971*, pages 481–489, 1971.
27. Robin Milner. *Communication and concurrency*. PHI Series in computer science. Prentice Hall, 1989.
28. Harry Nyquist. Certain topics in telegraph transmission theory. *Proceedings of the IEEE*, 47(2):617–644, April 1928.
29. John G. Proakis and Dimitris G Manolakis. *Digital Signal Processing*. Pearson, 4 edition, 2006.

30. James C. Robinson. *An Introduction to Ordinary Differential Equations*. Cambridge University Press, 2004.
31. Mauno Rökkö, Anders P. Ravn, and Kaisa Sere. Hybrid action systems. *Theor. Comput. Sci.*, 290(1):937–973, 2003.
32. Davide Sangiorgi. *Introduction to Bisimulation and Coinduction*. Cambridge University Press, 2011.
33. Claude E. Shannon. Communication in the presence of noise. *Proceedings of the I.R.E.*, pages 10–21, January 1949.
34. Zahava Shmuely. The structure of Galois connections. *Pacific Journal of Mathematics*, 54(2):209—225, 1974.
35. Wen Su, Jean-Raymond Abrial, and Huibiao Zhu. Formalizing hybrid systems with Event-B and the Rodin platform. *Sci. Comput. Program.*, 94:164–202, 2014.
36. Yong Kiam Tan and André Platzer. An axiomatic approach to liveness for differential equations. In *FM*, volume 11800 of *Lecture Notes in Computer Science*, pages 371–388. Springer, 2019.
37. Alfred Tarski. A lattice theoretical fixpoint theorem and its applications. *Pacific J. of Math.*, 5:285–310, 1955.
38. Andrew K. Wright and Matthias Felleisen. A syntactic approach to type soundness. *Inf. Comput.*, 115(1):38–94, 1994.
39. Jun Zhang, Karl Henrik Johansson, John Lygeros, and Shankar Sastry. Dynamical systems revisited: Hybrid systems with zeno executions. In *HSCC*, volume 1790 of *Lecture Notes in Computer Science*, pages 451–464. Springer, 2000.