A Uniform Approach to Three-Valued Semantics for μ-Calculus on Abstractions of Hybrid Automata

K. Bauer, R. Gentilini, and K. Schneider

Department of Computer Science, University of Kaiserslautern, Germany
{k_bauer,gentilin,schneider}@informatik.uni-kl.de

Abstract. Abstraction/refinement methods play a central role in the analysis of hybrid automata, that are rarely decidable. Soundness (of evaluated properties) is a major challenge for these methods, since abstractions can introduce unrealistic behaviors.

In this paper, we consider the definition of a three-valued semantics for μ-calculus on abstractions of hybrid automata. Our approach relies on two steps: First, we develop a framework that is general in the sense that it provides a preservation result that holds for several possible semantics of the modal operators. In a second step, we instantiate our framework to two particular abstractions. To this end, a key issue is the consideration of both over- *and* under-approximated reachability analysis, while classic simulation-based abstractions rely only on overapproximations, and limit the preservation to the universal (μ-calculus') fragment. To specialize our general result, we consider (1) so-called discrete bounded bisimulation abstractions, and (2) modal abstractions based on may/must transitions.

1 Introduction

Hybrid automata [16,1] provide an appropriate modeling paradigm for systems where continuous variables interact with discrete modes. Such models are frequently used in complex engineering fields like embedded systems, robotics, avionics, and aeronautics [2,12,24,25]. In hybrid automata, the interaction between discrete and continuous dynamics is naturally expressed by associating a set of differential equations to every location of a finite automaton.

Finite automata and differential equations are well established formalisms in mathematics and computer science. Despite of their long-standing tradition, their combination in form of hybrid automata leads to surprisingly difficult problems that are often undecidable. In particular, the *reachability* problem is undecidable for most families of hybrid automata [1,14,20,21,22], and the few decidability results are built upon strong restrictions of the dynamics [3,17]. The reachability analysis of hybrid automata is a fundamental task, since checking *safety* properties of the underlying system can be reduced to a reachability problem for the set of bad configurations [16].

For this reason, a growing body of research is being developed on the issue of dealing with approximated reachability on undecidable – yet reasonably expressive – hybrid automata [9,11,23,25,26]. To this end, most of the techniques proposed so far either rely on bounded state-reachability or on the definition of finite abstractions. While the first approach suffers inherently of incompleteness, the quest for *soundness* is a key issue in

H. Chockler and A.J. Hu (Eds.): HVC 2008, LNCS 5394, pp. 38–52, 2009.

the context of methods based on abstractions. In fact, abstractions can introduce unrealistic behaviors that may yield to spurious errors being reported in the safety analysis. Usually, a simulation preorder is required to relate the abstraction to the concrete dynamics of the hybrid system under consideration, ensuring at least the correctness of each response of (abstract) *non* reachability.

In this work, we provide a *uniform* approach to the sound evaluation of *general* reactive properties on abstractions of hybrid automata. Here, 'general' refers to the fact that we specify properties by means of the highly expressive logic of μ-calculus, that covers in particular CTL and other specification logics. 'Uniform', instead, emphasizes that we consider different possible classes of abstractions, whose analysis permits to recover both under- and overapproximations of state-sets fulfilling a given reachability requirement. Intuitively, this requirement is a minimal prerequisite for recovering sound abstract evaluations of arbitrary μ-calculus formulas.

To achieve our results we proceed by two steps: We start with the development of a generic semantics scheme for the μ-calculus, where the meaning of the modal operators can be adapted to particular abstractions. Assuming certain constraints for the semantics of these operators, we can prove a preservation result for our generic semantics scheme, thus providing a general framework for different classes of abstractions. In a subsequent step, we specialize our framework to suitable abstractions. In particular, we demonstrate the applicability of our framework by considering (1) the class of so-called discrete bounded bisimulation (DBB) abstractions [10], and (2) a kind of *modal* abstractions based on may/must transitions. As a final contribution, we compare these instances of our framework with respect to the issue of *monotonicity* of preserved μ-calculus formulas.

The paper is organized as follows: Preliminaries are given in Section 2. Section 3 introduces the classes of abstractions used in Section 5 to instantiate the generic result on preservative three-valued μ-calculus semantics outlined in Section 4. The monotonicity issue is dealt with in Section 6, while Section 7 concludes the paper discussing its contributions. All proofs are given in the appendix and in [4].

2 Preliminaries

In this section, we introduce the basic notions used in the remainder of the paper.

Definition 1 (Hybrid Automata [3]). *A* Hybrid Automaton *is a tuple* $H = \langle L, E, X, Init, Inv, F, G, R\rangle$ *with the following components:*

- *a finite set of* locations L
- *a finite set of* discrete transitions *(or* jumps*)* $E \subseteq L \times L$
- *a finite set of* continuous variables $X = \{x_1, \ldots x_n\}$ *that take values in* $\mathbb{R}$
- *an initial set of conditions: Init* $\subseteq L \times \mathbb{R}^n$
- *Inv* $: L \to 2^{\mathbb{R}^n}$*, the* invariant location labeling
- $F : L \times \mathbb{R}^n \to \mathbb{R}^n$*, assigning to each location* $\ell \in L$ *a vector field* $F(\ell, \cdot)$ *that defines the evolution of continuous variables within* ℓ
- $G : E \to 2^{\mathbb{R}^n}$*, the* guard edge labeling
- $R : E \times \mathbb{R}^n \to 2^{\mathbb{R}^n}$*, the* reset edge labeling.

We write $\mathbf{v}$ to represent a valuation $(v_1, \ldots, v_n) \in \mathbb{R}^n$ of the variables' vector $\mathbf{x} = (x_1, \ldots, x_n)$, whereas $\dot{\mathbf{x}}$ denotes the first derivatives of the variables in $\mathbf{x}$ (they all depend on the time, and are therefore rather functions than variables). A *state* in H is a pair $s = (\ell, \mathbf{v})$, where $\ell \in L$ is called the *discrete component* of s and $\mathbf{v}$ is called the *continuous component* of s. A *run* of H is a path in the *time abstract transition system* of H, given in Definition 2.

Definition 2. *The* time abstract transition system *of the hybrid automaton $H = \langle L, E, X, Init, Inv, F, G, R\rangle$ is the transition system $T_H = \langle Q, Q_0, \ell_{\rightarrow}, \rightarrow\rangle$, where:*

- *$Q \subseteq L \times \mathbb{R}^n$ and $(\ell, \mathbf{v}) \in Q$ if and only if $\mathbf{v} \in Inv(\ell)$*
- *$Q_0 \subseteq Q$ and $(\ell, \mathbf{v}) \in Q_0$ if and only if $\mathbf{v} \in Init(\ell) \cap Inv(\ell)$*
- *$\ell_{\rightarrow} = \{e, \delta\}$ is the set of edge labels, that are determined as follows:*
  - *there is a* continuous transition *$(\ell, \mathbf{v}) \xrightarrow{\delta} (\ell, \mathbf{v}')$, if and only if there is a differentiable function $f : [0, t] \rightarrow \mathbb{R}^n$, with $\dot{f} : [0, t] \rightarrow \mathbb{R}^n$ such that:*
    1. *$f(0) = \mathbf{v}$ and $f(t) = \mathbf{v}'$*
    2. *for all $\varepsilon \in (0, t)$, $f(\varepsilon) \in Inv(\ell)$, and $\dot{f}(\varepsilon) = F(\ell, f(\varepsilon))$.*
  - *there is a* discrete transition *$(\ell, \mathbf{v}) \xrightarrow{e} (\ell', \mathbf{v}')$ if and only if there exists an edge $(\ell, \ell') \in E$, $\mathbf{v} \in G(\ell)$ and $\mathbf{v}' \in R((\ell, \ell'), \mathbf{v})$.*

Definition 3 and Definition 4 recapitulate the syntax and the semantics of the μ-calculus language L_μ on hybrid automata, respectively [6,7].

Definition 3 (L_μ Syntax). *The set of μ-calculus preformulas for a set of labels $a \in \{e, \delta\}$ and propositions $p \in AP$ is defined by the following syntax:*

$$\phi := p \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid \phi_1 \wedge \phi_2 \mid \langle a\rangle\phi \mid [a]\phi \mid E(\phi_1 \underline{U} \phi_2) \mid A(\phi_1 \underline{U} \phi_2) \mid \mu Z.\phi \mid \nu Z.\phi$$

The set L_μ of μ-calculus formulas is defined as the subset of pre-formulas, where each subformula of the type $\mu Z.\phi$ and $\nu Z.\phi$ satisfies that all occurrences of Z in ϕ occur under an even number of negation symbols.

Definition 4 (Semantics of L_μ on Hybrid Automata). *Let AP be a finite set of propositional letters, let $p \in AP$ and consider $H = \langle L, E, X, Init, Inv, F, G, R\rangle$. Given $\ell_{\mathsf{AP}} : Q \rightarrow 2^{\mathsf{AP}}$ and $\phi \in L_\mu$, the function $[\![\phi]\!] : Q \rightarrow \{0, 1\}$ is inductively defined:*

- *$[\![p]\!](q) = 1$ iff $p \in l_{AP}(q)$*
- *$[\![\neg\phi]\!] := \neg\, [\![\phi]\!]$*
- *$[\![\phi \diamond \psi]\!] := [\![\phi]\!] \diamond [\![\psi]\!]$ for $\diamond \in \{\vee, \wedge\}$*
- *$[\![E(\phi \underline{U} \psi)]\!](q) = 1$ iff there exists a run ρ departing from q that admits a prefix $\rho^* := q_1 \xrightarrow{a_1} \ldots \xrightarrow{a_{n-1}} q_n$, where $q = q_1$, $a_i \in \{e, \delta\}$, $q_i = (l, v_i)$, satisfying:*
  - *$[\![\psi]\!](q_n) = 1$ and for $1 \leq i < n$: $[\![\phi]\!](q_i) = 1$*
  - *for $a_i = \delta$: $\exists$ a differentiable function $f : [0, t] \rightarrow \mathbb{R}^n$, for which:*
    1. *$f(0) = \mathbf{v_i}$ and $f(t) = \mathbf{v_{i+1}}$*
    2. *for all $\varepsilon \in (0, t)$, $f(\varepsilon) \in Inv(\ell)$, and $\dot{f}(\varepsilon) = F(\ell, f(\varepsilon))$*
    3. *for all $\varepsilon \in (0, t)$, $q' = (l_i, f(\varepsilon))$ satisfies $[\![\phi \vee \psi]\!](q') = 1$*
- *$[\![A(\phi \underline{U} \psi)]\!](q_1) = 1$ iff for all runs ρ departing from q there exists a prefix $\rho^* := q_1 \xrightarrow{a_1} \ldots \xrightarrow{a_{n-1}} q_n$, where $q = q_1$, $a_i \in \{e, \delta\}$, $q_i = (l, v_i)$, satisfying:*

  - $[\![\psi]\!](q_n) = 1$ *and for* $1 \leq i < n$: $[\![\phi]\!](q_i) = 1$
  - *for* $a_i = \delta$: $\exists$ *a differentiable function* $f : [0,t] \to \mathbb{R}^n$, *for which:*
    1. $f(0) = \mathbf{v_i}$ *and* $f(t) = \mathbf{v_{i+1}}$
    2. *for all* $\varepsilon \in (0,t)$, $f(\varepsilon) \in Inv(\ell)$, *and* $\dot{f}(\varepsilon) = F(\ell, f(\varepsilon))$
    3. *for all* $\varepsilon \in (0,t)$, $q' = (l_i, f(\varepsilon))$ *satisfies* $[\![\phi \vee \psi]\!](q') = 1$
- $[\![\langle a \rangle \phi]\!](q) = 1$ *iff* $\exists q \xrightarrow{a} q' : [\![\phi]\!](q') = 1$
  $[\![[a]\phi]\!](q) = 1$ *iff* $\forall q \xrightarrow{a} q' : [\![\phi]\!](q') = 1$
- *The fixpoint operators are defined in the following way:*
  Let $[\phi]_Z^{\psi}$ *be the formula obtained by replacing all occurrences of* Z *with* ψ. *Given a fixpoint formula* $\sigma Z.\phi$ *with* $\sigma \in \{\mu, \nu\}$ *its k-th approximation* $apx_k(\sigma Z.\phi)$ *is recursively defined as follows:*

$$apx_0(\mu Z.\phi) := 0 \text{ and } apx_{k+1}(\mu Z.\phi) := [\phi]_Z^{apx_k(\mu Z.\phi)}$$
$$apx_0(\nu Z.\phi) := 1 \text{ and } apx_{k+1}(\nu Z.\phi) := [\phi]_Z^{apx_k(\nu Z.\phi)}$$

  Then smallest and greatest fixpoints $[\![\sigma Z.\phi]\!]$ *are defined by*
  - *smallest fixpoint:* $[\![\mu Z.\phi]\!] := \bigvee_{k \in \mathbb{N}} [\![apx_k(\mu Z.\phi)]\!]$
  - *greatest fixpoint:* $[\![\nu Z.\phi]\!] := \bigwedge_{k \in \mathbb{N}} [\![apx_k(\mu Z.\phi)]\!]$

$H \vDash \phi$ *iff* $\forall q_0 \in Q_0 : [\![\phi]\!](q_0) = 1$.

The following definition recalls the notion of *simulation* relation, that plays a central role in the context of hybrid automata abstractions.

Definition 5 (Simulation). *Let* $T_1 = \langle Q^1, Q_0^1, \ell_{\rightarrow}, \rightarrow^1 \rangle$, $T_2 = \langle Q^2, Q_0^2, \ell_{\rightarrow}, \rightarrow^2 \rangle$, $Q^1 \cap Q^2 = \emptyset$, *be two edge-labeled transition systems and let* $\mathcal{P}$ *be a partition on* $Q_1 \cup Q_2$. *A* simulation *from* T_1 *to* T_2 *is a non-empty relation on* $\rho \subseteq Q^1 \times Q^2$ *such that, for all* $(p,q) \in \rho$*:*

- $p \in Q_0^1$ *iff* $q \in Q_0^2$ *and* $[p]_{\mathcal{P}} = [q]_{\mathcal{P}}$.
- *for each label* $a \in \ell_{\rightarrow}$, *if there exists* p' *such that* $p \xrightarrow{a} p'$, *then there exists* q' *such that* $(p', q') \in \rho$ *and* $q \xrightarrow{a} q'$.

If there exists a simulation from T_1 *to* T_2, *then we say that* T_2 simulates T_1, *denoted* $T_1 \leq_S T_2$. *If* $T_1 \leq_S T_2$ *and* $T_2 \leq_S T_1$, *then* T_1 *and* T_2 *are said* similar, *denoted* $T_1 \equiv_S T_2$. *If* ρ *is a simulation from* T_1 *to* T_2, *and the inverse relation* ρ^{-1} *is a simulation from* T_2 *to* T_1, *then* T_1 *and* T_2 *are said* bisimilar, *denoted* $T_1 \equiv_B T_2$

3 Abstractions of Hybrid Automata for Parallel over- and Underapproximated Reachability Analysis

In this section, we introduce two kinds of abstractions that we will use in the sequel to specialize our general preservation result for μ-calculus semantics.

Most of the abstraction/refinement methods for hybrid automata in the literature are based on overapproximations of the reachable states[1]. In particular, they rely on a generic notion of abstractions based on the simulation preorder. The latter is required to relate the abstraction to the dynamics of the hybrid automaton, as formalized below.

[1] Note that the reachability problem is undecidable for most classes of hybrid automata.

Definition 6 (Abstraction). *Let H be a hybrid automaton. An abstraction of H is a finite transition system $A = \langle R, R_0, \xrightarrow{\delta}, \xrightarrow{e} \rangle$ in which*

1. *R is a finite partition of the state space of H, $R_0 \subseteq R$ is a partition of the initial states, $\xrightarrow{\delta} \subseteq R \times R$ and $\xrightarrow{e} \subseteq R \times R$*
2. *$A^* := \langle R, R_0 \xrightarrow{\delta *}, \xrightarrow{e} \rangle$ simulates the time abstract transition system T_H associated to H, where $\xrightarrow{\delta *}$ denotes the transitive closure of the continuous transitions $\xrightarrow{\delta}$*

Since this basic notion of abstraction gives only an overapproximation of the hybrid automaton's reachable states, its usage is inherently limited to the universal fragment of the μ-calculus [5]. As we are interested in unrestricted μ-calculus properties, we need a more powerful abstraction/refinement approach. To this end, a minimum requirement is the combination of both over- and underapproximations of state-sets satisfying a given reachability property. The consideration of parallel over- and underapproximated reachability on hybrid automata is quite new: In [10], discrete bounded bisimulation (DBB) abstractions, briefly recalled in Subsection 3.1, were designed for this purpose. Another approach that leads to over- and underapproximations is given by *modal abstractions* for hybrid automata, that we develop in Subsection 3.2 (generalizing the definitions given in context of discrete systems [13]).

3.1 Discrete Bounded Bisimulation (DBB) Abstractions

It is well known that the classic bisimulation equivalence can be characterized as a coarsest partition stable with respect to a given transition relation [18]. Bounded bisimulation imposes a bound on the number of times each edge can be used for partition refinement purposes. For the equivalence of discrete bounded bisimulation (DBB), the latter bound applies only to the *discrete* transitions of a given hybrid automaton, as recalled in Definition 7, below.

Definition 7 (Discrete Bounded Bisimulation [10]). *Let H be an hybrid automaton, and consider the partition $\mathcal{P}$ on the state-space Q of $T_H = \langle Q, Q_0, \ell_{\rightarrow}, \rightarrow \rangle$. Then:*

1. *$\equiv_0 \in Q \times Q$ is the maximum relation on Q such that for all $p \equiv_0 q$*
   (a) $[p]_P = [q]_P$ and $p \in Q_0$ iff $q \in Q_0$
   (b) $\forall p \xrightarrow{\delta} p' \exists q' : p' \equiv_0 q' \wedge q \xrightarrow{\delta} q'$
   (c) $\forall q \xrightarrow{\delta} p' \exists q' : p' \equiv_0 q' \wedge p \xrightarrow{\delta} p'$
2. *$\equiv_n \in Q \times Q$ is the maximum relation on Q such that for all $p \equiv_n q$*
   (a) $p \equiv_{n-1} q$
   (b) $\forall p \xrightarrow{\delta} p' \exists q' : p' \equiv_n q' \wedge q \xrightarrow{\delta} q'$
   (c) $\forall q \xrightarrow{\delta} p' \exists p' : p' \equiv_n q' \wedge p \xrightarrow{\delta} p'$
   (d) $\forall p \xrightarrow{e} p' \exists q' : p' \equiv_{n-1} q' \wedge q \xrightarrow{e} q'$
   (e) $\forall q \xrightarrow{e} q' \exists p' : p' \equiv_{n-1} q' \wedge p \xrightarrow{e} p'$

For $n \in \mathbb{N}$, the relation $\equiv_n$ will be called n-DBB equivalence.

The succession of n-DBB equivalences over an hybrid automaton H naturally induces a series of abstractions for H, as stated in Definition 8.

Definition 8 (Series of DBB Abstractions [10]). *Let H be a hybrid automaton and $T_H = \langle Q, Q_0, l_{\rightarrow}, \rightarrow \rangle$ be the associated time abstract transition system. Let $\mathcal{P}$ be a partition of Q and consider the n-DBB equivalence $\equiv_n$. Then, the n-DBB abstraction $H_{\equiv_n} = \langle Q', Q'_0, l_{\rightarrow} \rightarrow' \rangle$ is defined as follows:*

- $Q' = Q_{/\equiv_n}$, $Q'_0 = Q_{0/\equiv_n}$
- $\forall \alpha, \beta \in Q'$:
  - $\alpha \xrightarrow{e} \beta$ *iff* $\exists a \in \alpha \exists b \in \beta : a \xrightarrow{e} b$
  - $\alpha \xrightarrow{\delta} \beta$ *iff* $\exists a \in \alpha \exists b \in \beta : a \xrightarrow{\delta} b$ *and the path* $a \rightsquigarrow b$ *only traverses* α *and* β

The existence of a simulation preorder relating successive elements in a series of DBB abstractions allows the refinement of overapproximations of reachable sets in the considered hybrid automaton [10]. Moreover, $H_{\equiv_n}$ preserves the reachability of a given region of interest (in the initial partition) whenever the latter can be established on H following a path that traverses at most n locations [10]. On this ground, it is also possible to use the succession of DBB abstractions to obtain $\subseteq$-monotonic underapproximations of the set of states fulfilling a given reachability requirement.

3.2 Modal Abstractions Based on May/Must Relations

For discrete systems [13], a *may*-transition between two abstract classes r and r' encodes that for at least some state in r there is a transition to some state in r'. In turn, a *must*-transition between r and r' states that all states in r have a transition to a state in r'. Naturally, may-transitions (resp. must-transitions) refer to overapproximated (resp. underapproximated) transitions among classes of an abstract system. The above ideas can be extended to the context of hybrid automata as formalized in Definition 9.

Definition 9 (Modal Abstractions). *Let $A = \langle R, R_0, \xrightarrow{\delta}, \xrightarrow{e} \rangle$ be an abstraction of the hybrid automaton H. Then A is a* modal abstraction *(or* may/must abstraction*) of H iff the following properties hold:*

- $\xrightarrow{\delta} \supseteq \xrightarrow{\delta}_{must}$, *where* $\xrightarrow{\delta}_{must}$ *is defined as follows:*
  $r \xrightarrow{\delta}_{must} r'$ *iff for all $x \in r$ there exists an $x' \in r'$ such that H can evolve continuously from the state x to the state x' by traversing the only regions r and r'.*
- $\xrightarrow{e} \supseteq \xrightarrow{e}_{must}$ *where* $\xrightarrow{e}_{must}$ *is defined as follows:*
  $r \xrightarrow{e}_{must} r'$ *iff for all $x \in r$ there exists an $x' \in r'$ s.t. $x \xrightarrow{e} x'$ in H.*

The subautomaton $\langle R, R_0, \xrightarrow{\delta}_{must}, \xrightarrow{e}_{must} \rangle$ of A is called A_{must}.

Given the modal abstraction A for the hybrid automaton H, Lemma 1 states that A_{must} is simulated by the time abstract transition system T_H of H.

Lemma 1. *Let H be a hybrid automaton and let A be a may/must abstraction of H. Then, the subautomaton A_{must} of A is simulated by T_H, i.e. $A_{must} \leq_S T_H \leq_S A^*$.*

On this ground, may/must abstractions can be used not only to overapproximate, but also to underapproximate the set of states modeling a given reachability property, as stated in Corollary 1.

Corollary 1. *Let $A = \langle R, R_0 \xrightarrow{\delta}, \xrightarrow{e} \rangle$ be a modal abstraction for the hybrid automaton H, and let F be a set of (final) states in H. Assume that F is consistent with respect to R, i.e. for all $r \in R : r \cap F = r \ \vee \ r \cap F = \emptyset$. If $r \in R$ admits a path to $r' \subseteq F$ in $\mathcal{A}_{must}$, then for all $s \in r$, exists $s' \in r'$ such that H admits a run from s to s'.*

4 A Generic Semantics for μ-Calculus on Abstractions of Hybrid Automata

In this section, we present one of the main ingredients of our approach: a *generic three-valued* semantics for μ-calculus on abstractions of hybrid automata. Here, two keywords deserve our attention: Generic and three-valued.

The choice of a three-valued logic as the base of our semantics is motivated by the broad family of abstractions that we consider for our framework. In fact, the abstractions we have in mind are in general less precise than a bisimulation (which allows for exact reachability analysis, but is seldom finite), and more precise than a simulation (that allows only for overapproximated reachability analysis). Hence, we can not expect that *any* μ-calculus formula is preserved, however it should be possible to recover at least all true universal μ-calculus subformulas[2]. By means of a three-valued logic, we can use the third value $\perp$ to distinguish the cases for which it is not possible to derive a boolean truth value, due to the coarseness of the abstraction. Instead, the preservation applies to all the boolean results established using the abstract semantics. In the following, we write $\neg_3$, $\vee_3$, $\wedge_3$ for the three-valued extensions of the boolean operations $\neg$, $\vee$, $\wedge$, respectively[3].

The second keyword – generic – is better understood as a way of establishing a link between (1) the quest for soundness in our semantics, and (2) the variety of patterns according to which different abstractions split the information over their regions. Our generic semantics is an abstract semantics scheme, where the evaluation is fixed for some operators (namely boolean and fixpoint operators), and only subject to some constraints for the others. The constraints are sufficient to establish a general preservation result, though the semantics scheme can be adapted to several classes of abstractions.

Given the above premises, we are now ready to formalize in Definition 10 our three-valued generic semantics for μ-calculus on abstractions of hybrid automata. Note that for a μ-calculus formula ϕ, we distinguish between the semantics $[\![\phi]\!]_H$ on a hybrid automaton H (as given in Definition 4) and the semantics $[\![\phi]\!](r)$ on the region r of an abstraction of H.

Definition 10 (Generic μ-Calculus Semantics). *Let H be a hybrid automaton whose state space is partitioned into finitely many regions of interest by the labeling function $l_{AP} : Q \to 2^{AP}$, where AP is a finite set of propositional letters. Let ϕ be a*

[2] Recall that bisimulation preserves the whole μ-calculus, while simulation preserves the only true universally quantified formulas.

[3] We use Kleene's definition of three-valued logic [19].

μ-calculus formula with atomic propositions AP, and consider the abstraction $A = \langle R, R_0, l_{\rightarrow}, \rightarrow\rangle$ where R is assumed to refine[4] *the regions of interest in H.*

1. *If ϕ is an atomic proposition, then* $[\![\phi]\!](r) = \begin{cases} 1 & \phi \in l_{AP}(r) \\ 0 & \textit{otherwise} \end{cases}$
2. *If $\phi = \neg\varphi$, then $[\![\neg\varphi]\!] = \neg_3[\![\varphi]\!]$*
3. *If $\phi = \varphi \vee \psi$, then $[\![\varphi \vee \psi]\!] = [\![\varphi]\!] \vee_3 [\![\psi]\!]$*
4. *If $\phi = \varphi \wedge \psi$, then $[\![\varphi \wedge \psi]\!] = [\![\varphi]\!] \wedge_3 [\![\psi]\!]$*
5. *If $\phi \in \{\langle\delta\rangle\varphi, \langle e\rangle\varphi, [\delta]\varphi, [e]\varphi, E(\varphi \underline{U} \psi), A(\varphi \underline{U} \psi)\}$, then $[\![\phi]\!]$ is required to fulfill the following conditions:*

$$[\![\phi]\!](r) = 1 \Rightarrow \forall\, x \in r :\ [\![\phi]\!]_H(x) = 1$$
$$[\![\phi]\!](r) = 0 \Rightarrow \forall\, x \in r :\ [\![\phi]\!]_H(x) = 0$$

6. *Let $\phi \in \{\mu Z.\varphi, \nu Z.\varphi\}$ be a fixpoint formula. Let $[\varphi]_Z^{\psi}$ be the formula obtained by replacing all occurrences of Z with ψ. Given a fixpoint formula $\sigma Z.\varphi$ with $\sigma \in \{\mu, \nu\}$, its k-th approximation $apx_k(\sigma Z.\varphi)$ is recursively defined as follows:*
   - *$apx_0(\mu Z.\varphi) := 0$ and $apx_{k+1}(\mu Z.\varphi) := [\varphi]_Z^{apx_k(\mu Z.\varphi)}$*
   - *$apx_0(\nu Z.\varphi) := 1$ and $apx_{k+1}(\nu Z.\varphi) := [\varphi]_Z^{apx_k(\nu Z.\varphi)}$*

   The semantics of least and greatest fixpoints $[\![\sigma Z.\varphi]\!]$ are defined by $[\![apx_{\hat{k}}\sigma Z.\varphi]\!]$ where $\hat{k}$ is the smallest index where $[\![apx_{\hat{k}}(\sigma Z.\varphi)]\!] = [\![apx_{\hat{k}+1}(\sigma Z.\varphi)]\!]$ holds.

Let ϕ be a μ-calculus formula and let $A = \langle R, R_0 \xrightarrow{\delta}, \xrightarrow{e}\rangle$ be an abstraction of the hybrid automaton H. On the ground of Definition 10, we can define a three-valued relation $\models_3$ stating whether A is a model of the formula ϕ:

$$A \models_3 \phi = \begin{cases} 1 & \forall r \in R_0 : [\![\phi]\!](r) = 1 \\ 0 & \exists r \in R_0 : [\![\phi]\!](r) = 0 \\ \bot & otherwise \end{cases}$$

Theorem 1 below states that both results true and false established on A via $\models_3$ are preserved on the underlying hybrid automaton. Note that Theorem 1 has a sort of *uniform* character, since $\models_3$ subsumes indeed many possible effective semantics for μ-calculus, the latter recovered by specializing the semantics of the modal operators according to the properties of different classes of abstractions. For the rest of this work let $\preceq$ be the partial order over $\{0, 1, \bot\}$ defined by the reflexive closure of $\{(\bot, 0), (\bot, 1)\}$.

Theorem 1 (Uniform Preservation Theorem). *Let H be a hybrid automaton, let A be an abstraction of H. Then, for any μ-calculus formula ϕ, we have $A \models_3 \phi \preceq H \models \phi$.*

Hence, if $A \models_3 \phi$ is 1, so is $H \models \phi$, and if $A \models_3 \phi$ is 0, so is $H \not\models \phi$, and if $A \models_3 \phi$ is $\bot$, then $H \models \phi$ is completely unknown. For this reason, both valid and invalid subformulas can be preserved with our framework as long as the abstraction is not too coarse.

[4] Note that our assumption (the partition of Q into regions of interest is refined by the abstraction $A = \langle R, R_0, l_{\rightarrow}, \rightarrow\rangle$) implies that $\forall r \in R\, \forall x_1, x_2 \in R :\ l_{AP}(x_1) = l_{AP}(x_2)$ holds. Thus, the labeling function can be easily extended to $l_{AP} : R \rightarrow 2^{AP}$.

5 Instantiation to DBB- and May/Must-Abstractions

In this section, we specialize the general preservation result given in Section 4 to two particular instances, namely to modal and DBB abstractions. As a result, we obtain two preservative abstraction/refinement frameworks for μ-calculus on hybrid automata.

5.1 Semantics Completion for May/Must-Abstractions

In modal abstractions, each $\xrightarrow{\delta}_{must}$ (resp. $\xrightarrow{e}_{must}$) edge underapproximates a continuous (resp. discrete) evolution for the underlying hybrid automata. In turn, each $\xrightarrow{\delta}$ (resp. $\xrightarrow{e}$) edge overapproximates a continuous (resp. discrete) evolution for the considered hybrid automaton. The above considerations can be used to properly instantiate the semantics for the modal operators on may/must abstractions, completing the semantics scheme given in Definition 10. Consider for example the modal operator $\langle\delta\rangle$. According to the adaptive semantics scheme in Definition 10, we should instantiate the semantics $[\![\langle\delta\rangle\varphi]\!]$ in such a way that whenever $[\![\langle\delta\rangle\varphi]\!]$ evaluates to 1 (resp. 0) on an abstract region, then it evaluates to 1 (resp. 0) on all the states of the region. This constraint is naturally guaranteed on modal abstractions if we use only $\xrightarrow{\delta}_{must}$ edges (resp. $\xrightarrow{\delta}$ edges) to inspect for true (resp. false) evaluations. A similar way of reasoning allows to completely adapt the semantics scheme in Definition 10 to the case of modal abstractions, as formalized in Definition 11.

Definition 11. *Let H be a hybrid automaton, $A = \langle R, R_0 \xrightarrow{\delta}, \xrightarrow{e}\rangle$ be a may/must abstraction of H and let φ and ψ be μ-calculus formulas. Then, the semantics of the three-valued μ-calculus of Definition 10 for $a, a_i \in \{\delta, e\}$ is completed by:*

- $[\![\langle e\rangle\varphi]\!](r) \begin{cases} 1 & \exists r \xrightarrow{e}_{must} r' : [\![\varphi]\!](r') = 1 \\ 0 & \forall r \xrightarrow{e} r' : \quad [\![\varphi]\!](r') = 0 \\ \bot & \textit{otherwise} \end{cases}$
- $[\![\langle\delta\rangle\varphi]\!](r) \begin{cases} 1 & \exists r \xrightarrow{\delta}_{must} r' : [\![\varphi]\!](r') = 1 \\ 0 & \forall r \xrightarrow{\delta}{}^{*} r' : \quad [\![\varphi]\!](r') = 0 \\ \bot & \textit{otherwise} \end{cases}$
- $[\![[a]\phi]\!] = [\![\neg(\langle a\rangle\neg\phi)]\!]$
- *Let $\{r_n\}_{n\in\mathbb{N}}$ (resp. $\{r_n\}^{must}_{n\in\mathbb{N}}$) denote an infinite path of A (resp. A_{must}) starting in $r = r_0$. Then:*

$$[\![E(\varphi \underline{U} \psi)]\!](r) \begin{cases} 1 & \exists \{r_n\}^{must}_{n\in\mathbb{N}} \exists k \in \mathbb{N} : [\![\psi]\!](r_k) = 1 \wedge [\![\varphi]\!](r_{i<k}) = 1 \\ 0 & \forall \{r_n\}_{n\in\mathbb{N}} \forall k \in \mathbb{N} : \ [\![\psi]\!](r_k) \neq 0 \Rightarrow \exists i < k : [\![\varphi]\!](r_i) = 0 \\ \bot & \textit{otherwise} \end{cases}$$

$$[\![A(\varphi \underline{U} \psi)]\!](r) \begin{cases} 1 & \forall \{r_n\}_{n\in\mathbb{N}} \exists k \in \mathbb{N} : \ [\![\psi]\!](r_k) = 1 \ \wedge \ [\![\varphi]\!](r_{i<k}) = 1 \\ 0 & \exists \{r_n\}^{must}_{n\in\mathbb{N}} \forall k \in \mathbb{N} : [\![\psi]\!](r_k) \neq 0 \Rightarrow \exists i < k : [\![\varphi]\!](r_i) = 0 \\ \bot & \textit{otherwise} \end{cases}$$

Lemma 2, below, states the correctness of our instantiation, namely it ensures that the semantics for the modal operators on may/must abstractions in Definition 11 fulfill the constraints provided in Definition 10.

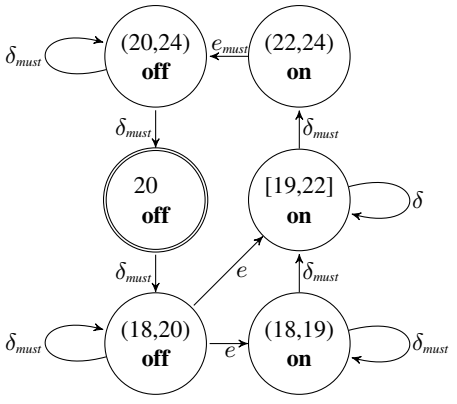

Fig. 1. May/Must Abstraction A_1 of the Heating Controller

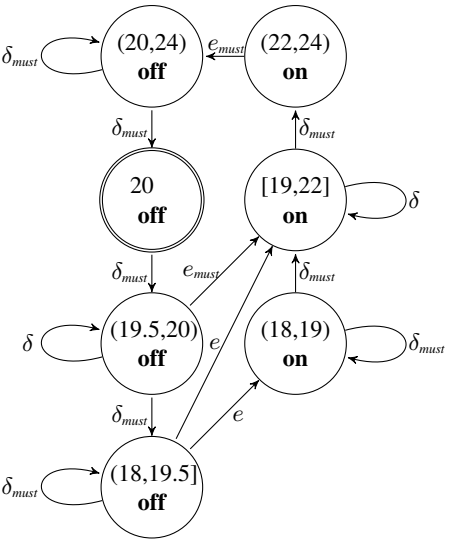

Fig. 2. May/Must Abstraction A_2 of the Heating Controller

Lemma 2. *Let A be a modal abstraction for the hybrid automaton H, and assume to interpret μ-calculus formulas according to Definition 11. Then, for any formula $\phi \in \{\langle\delta\rangle\varphi, \langle e\rangle\varphi, [\delta]\varphi, [e]\varphi, E(\varphi \underline{U} \psi), A(\varphi \underline{U} \psi)\}$, we have:*

$$[\![\phi]\!](r) = 1 \Rightarrow \forall\, x \in r :\ [\![\phi]\!]_H(x) = 1$$
$$[\![\phi]\!](r) = 0 \Rightarrow \forall\, x \in r :\ [\![\phi]\!]_H(x) = 0$$

On this ground, the uniform preservation theorem given in the previous section (cfr. Theorem 1) applies to our specialized semantics, as stated in Corollary 2.

Corollary 2. *Let A be a modal abstraction for the hybrid automaton H, and assume to interpret μ-calculus formulas according to Definition 11. Then $A \models_3 \phi \preceq H \models \phi$.*

We conclude this subsection by providing a concrete example, which illustrates our three-valued semantics on modal abstractions.

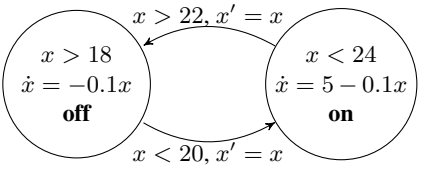

Fig. 3. Heating Controller

Figure 3 depicts a heating controller consisting of the two discrete states **off** and **on**. While the heating is off, the temperature x falls via the differential rule $\dot{x} = -0.1x$. Conversely, while the heating is on, the temperature rises via $\dot{x} = 5 - 0.1x$. The location **off** may be left, when the temperature falls below 20 degree and it has to be left, when x falls below 18 degree. Symmetric conditions hold for **on**. Initially, the heating controller starts at the location **off** with a

temperature of 20 degrees. Figure 1 and Figure 2 depict two different modal abstractions A_1 and A_2 for the heating controller. Consider the formula $\psi = \mu Z.\phi \vee \Diamond Z$, where ϕ denotes a propositional letter being true for the abstract state $(\textbf{off}, (20, 24))$. This formula holds in the states that can reach a configuration where the temperature is between 20 and 24 degree and the heating is off. Applying the semantics scheme on A_1, this formula can not be proven since A_1 does not admit a must-path from the initial region to $(\textbf{off}, (20, 24))$. Conversely, ψ can not be falsified, since there exists a may-path from the initial region to the target region. Using A_2 instead we can establish $A_2 \vDash \psi$, since A_2 contains a must-path leading to $(\textbf{off}, (20, 24))$. This yields $H \vDash \psi$, by our preservation theorem.

5.2 Semantics Completion for DBB Abstractions

We now turn out to the consideration of DBB abstractions, providing a further specialization of the uniform preservation result discussed in section 4.

DBB abstractions encode the information for parallel over- and underapproximated reachability analysis differently from modal abstractions. In particular, there is no distinction between edges that over-estimate (resp. under-estimate) the evolution of the underlying hybrid automaton. Rather, a discrete edge between the abstract states $[r]_{\equiv_n}$ and $[r]_{\equiv_n}$ in $H_{\equiv_n}$ means that H can evolve from $[r]_{\equiv_n}$ to $[r']_{\equiv_{n-1}} \supseteq [r']_{\equiv_n}$, via a discrete edge. The continuous edges in $H_{\equiv_n}$ represent instead with fidelity the continuous evolution along the regions of the abstraction. These considerations are useful to understand the ratio underlying the development of the exact semantics for the modal operators on DBB abstractions, given in Definition 12.

Definition 12. *Let H be a hybrid automaton and $H_{\equiv_n} = \langle Q_{/\equiv n}, Q_{0/\equiv n}, l_{\rightarrow}, \rightarrow_{/\equiv n}\rangle$ be its n-DBB abstraction. Then the semantics scheme in Definition 10 is completed by the following rules:*

- *The value of $[\![\langle\delta\rangle\varphi]\!]_{\equiv n}([x]_{\equiv n})$ is given by*

$$\begin{cases} 1 & \exists [x']_{\equiv n} \in Q_{/\equiv n} : [x]_{\equiv n} \xrightarrow{\delta} [x']_{\equiv n} \wedge [\![\varphi]\!]_{\equiv n}([x']_{\equiv n}) = 1 \\ 0 & \nexists [x']_{\equiv n} \in Q_{/\equiv n} : [x]_{\equiv n} \xrightarrow{\delta}{}^{*} [x']_{\equiv n} \wedge [\![\varphi]\!]_{\equiv n}([x']_{\equiv n}) = 1 \\ \bot & otherwise \end{cases}$$

- *The value of $[\![\langle e\rangle\varphi]\!]_{\equiv n}([x]_{\equiv n})$ is given by*

$$\begin{cases} 1 & \exists [x']_{\equiv n} \in Q_{/\equiv n} : [x]_{\equiv n} \xrightarrow{\delta} [x']_{\equiv n} \wedge [\![\varphi]\!]_{\equiv n-1}([x']_{\equiv n}) = 1 \\ 0 & \nexists [x']_{\equiv n} \in Q_{/\equiv n} : [x]_{\equiv n} \xrightarrow{e} [x']_{\equiv n} \wedge [\![\varphi]\!]_{\equiv n}([x']_{\equiv n}) \neq 0 \\ \bot & otherwise \end{cases}$$

- $[\![[a]\varphi]\!]_{\equiv n} := [\![\neg(\langle a\rangle\neg\varphi)]\!]_{\equiv n}$ *for* $a \in \{e, \delta\}$
- *Let us use the notation $\{[x_i]_{\equiv n}\}$ to represent an infinite path in $H_{\equiv n}$. Then: The value of $[\![E(\varphi \underline{U} \psi)]\!]_{\equiv n}([x_0]_{\equiv n})$ is given by*

$$\begin{cases} 1 & \exists\{[x_i]_{\equiv n}\}\exists k \in \mathbb{N} : 1.\ [x_{i<k}]_{\equiv n} \xrightarrow{\delta} [x_{i+1}]_{\equiv n} \wedge [\![\varphi \vee \psi]\!]_{\equiv n}([x_i]_{\equiv n}) = 1 \\ & \qquad\qquad 2.\ [\![\psi]\!]_{\equiv n}([x_k]_{\equiv n}) = 1 \text{ or} \\ & \qquad\qquad\quad [x_k]_{\equiv n} \xrightarrow{e} [x_{k+1}]_{\equiv n} \wedge [\![E(\varphi \underline{U} \psi)]\!]_{\equiv n-1}([x_{k+1}]_{\equiv n-1}) = 1 \\ 0 & \forall\{[x_i]_{\equiv n}\}\forall k \in \mathbb{N} : [\![\psi]\!]_{\equiv n}([x_k]_{\equiv n}) \neq 0 \Rightarrow \exists j < k :\ [\![\varphi \vee \psi]\!]_{\equiv n}([x_j]_{\equiv n}) = 0 \\ \bot & otherwise \end{cases}$$

The value of $[\![A(\varphi \underline{U} \psi)]\!]_{\equiv n}([x_0]_{\equiv n})$ *is given by*

$$\begin{cases} 1 & \forall\{[x_i]_{\equiv n}\}\exists k \in \mathbb{N} : [\![\psi]\!]_{\equiv n}([x_k]_{\equiv n}) = 1 \ \wedge \ [\![\varphi \vee \psi]\!]_{\equiv n}([x_{i<k}]_{\equiv n}) = 1 \\ 0 & \exists\{[x_i]_{\equiv n}\}\exists k \in \mathbb{N} : 1.\ [x_{i<k}]_{\equiv n} \xrightarrow{\delta} [x_{i+1}]_{\equiv n} \wedge [\![\varphi \wedge \neg\psi]\!]_{\equiv n}([x_i]_{\equiv n}) = 1 \\ & \qquad\qquad\ 2.\ [\![\varphi \vee \psi]\!]_{\equiv n}([x_k]_{\equiv n}) = 0 \textit{ or} \\ & \qquad\qquad\quad [x_k]_{\equiv n} \xrightarrow{e} [x_{k+1}]_{\equiv n} \wedge [\![A(\varphi \underline{U} \psi)]\!]_{\equiv n-1}([x_{k+1}]_{\equiv n-1}) = 0 \\ \bot & \textit{otherwise} \end{cases}$$

On the ground of Lemma 3 the uniform preservation theorem in Section 4 applies also to our specialized semantics for DBB abstractions, as formally stated in Corollary 3.

Lemma 3. *Let $H_{\equiv n}$ be an n-DBB abstraction for the hybrid automaton H, and assume to interpret μ-calculus formulas according to Definition 12. Then, for any formula $\phi \in \{\langle\delta\rangle\varphi, \langle e\rangle\varphi, [\delta]\varphi, [e]\varphi, E(\varphi \underline{U} \psi), A(\varphi \underline{U} \psi)\}$:*

$$[\![\phi]\!](r) = 1 \Rightarrow \ \forall\, x \in r :\ [\![\phi]\!]_H(x) = 1$$
$$[\![\phi]\!](r) = 0 \Rightarrow \ \forall\, x \in r :\ [\![\phi]\!]_H(x) = 0$$

Corollary 3. *Let $H_{\equiv n}$ be an n-DBB abstraction for the hybrid automaton H, and assume to interpret μ-calculus formulas according to Definition 12. Then for any formula $\phi \in L_\mu$: $H_{\equiv_n} \models_3 \phi \preceq H \models \phi$*

The following example illustrates the instantiation of the semantic framework to DBB abstractions described so far.

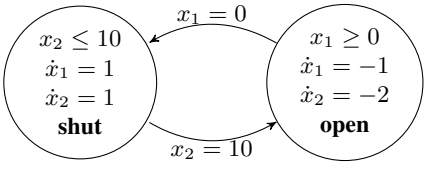

Fig. 4. Water Level Controller

The hybrid automaton depicted in Fig. 4 models a water level controller with two variables. The first variable x_1 represents a clock, while the second variable x_2 models the water level in the tank. When the valve at the bottom of the tank is closed, the water level increases by $1ms^{-1}$, otherwise it decreases by $2ms^{-1}$. Intuitively, the clock allows to establish that the valve remains open as long as it was closed in the previous step. This hybrid automaton does not belong to any known decidable class , and it yields infinite bisimulations for suitable initial partitions [15]. This is the case e.g. for the initial partition

$$P = \{\textbf{shut} \times X,\ \textbf{shut} \times Y,\ \textbf{open} \times X,\ \textbf{open} \times Y\}$$

where $X = [0, 6] \times \{10\}$ and $Y = [0, \infty) \times (-\infty, 10] \setminus X$. However, the above automaton is fully O-minimal and thus the construction of DBB-Abstractions terminates [10].

Consider the following question 'When starting in *Init* = **open** $\times\, [0, 6] \times \{10\}$, does the water level controller always admit an evolution to r = **shut** $\times\, [0, 6] \times \{10\}$?'. Such a question corresponds to compute whether $H \models \psi$, where $\psi = \mu Z.r \vee \Diamond Z$. We use DBB abstractions to falsify the above property. Figure 5 and Fig. 6 depict the 0-DBB and 1-DBB abstraction, respectively. In the 0-DBB abstraction the formula ψ evaluates to 1 on r_1, r_2, r_3 and s_1, and is indefinite elsewhere. Thus, $(H \models \psi) = \bot$ since $[\![\psi]\!](s_2) = \bot$ for the only initial region s_2 of $H_{\equiv 0}$. In the 1-DBB abstraction $H_{\equiv 1}$ the region s_2 gets split to $\langle t_2, t_3\rangle$ and ψ evaluates to 0 on t_3. Since all a paths starting in t_3 do not allow to reach a region, where ψ evaluates to 1 or $\bot$, we can conclude that $(H_{\equiv 1} \models \psi) = 0$. Thus, due to the preservation theorem we can state that $H \not\models \psi$.

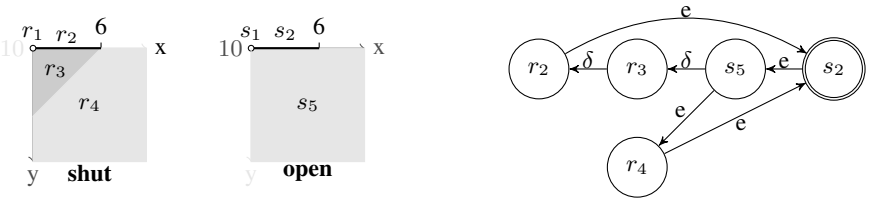

Fig. 5. 0-DBB Abstraction: Partitioning of the Regions and Control Graph of the Abstraction (for simplification the cycle $r_1 \leftrightarrow s_1$ is left out)

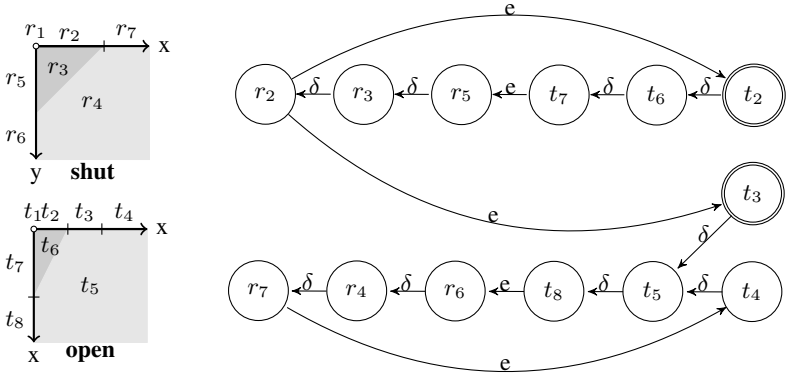

Fig. 6. 1-DBB Abstraction: Partitioning of the Regions and Control Graph of the Abstraction (for simplification the cycle $r_1 \leftrightarrow s_1$ is left out)

6 Abstraction Refinement and Monotonicity

A key issue in the context of three-valued abstract semantics for μ-calculus on hybrid automata is related to *monotonicity*. Given an abstraction-refinement framework, it is desirable that the set of formulas evaluating to $\perp$ decreases monotonically in its size along any succession of finer abstractions. Such a requirement is reminiscent of the usual *regularity* property for Kleene's three-valued logics [8,19].

In this section, we compare the two abstraction refinement frameworks based on DBB-abstractions and modal abstractions with respect to monotonicity. Theorem 2 proves that the DBB succession of abstractions allows to monotonically recover true/false μ-calculus formulas along the series of refining abstractions.

Theorem 2 (Monotonicity). *Let $H_{\equiv n}$ and $H_{\equiv k}$ with $n > k$ be DBB abstractions of the hybrid system H, and let ϕ be a μ-calculus formula. Then, $(H_{\equiv k} \vDash \phi) \preceq (H_{\equiv n} \vDash \phi)$.*

The following example shows instead that the abstraction/refinement framework based on modal abstractions does not behave well with respect to monotonicity.

Example 1. Let us consider the abstraction A_3 depicted in Fig. 7 which is a refinement of the abstraction A_2 given in Fig. 2. In Section 5.1 we were able to establish the result

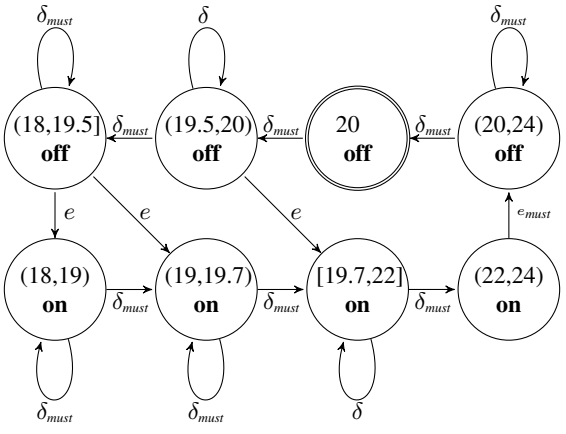

Fig. 7. Abstraction A_3 with may/must of the heating controller

$(A_2 \models_3 \mu Z.\phi \vee \Diamond Z) = 1$, where ϕ is a propositional letter being true on $(20, 24) \times$ **off**. However, we cannot prove $(A_3 \models_3 \mu Z.\phi \vee \Diamond Z) = 1$ since there exist no $\xrightarrow{e}_{must}$-transitions from the configuration **off** to the configuration **on**.

7 Conclusions

In this paper, we developed a framework for inferring general μ-calculus properties on abstractions of hybrid automata. Based on the definition of a sound three-valued semantics on abstractions, our framework does not feature the inherent limitations of bounded model checking or techniques using the simulation preorder. In particular, our method can *both* prove and disprove arbitrary μ-calculus properties on abstractions over- and underapproximating (unbounded) evolutions of the system. To cope with the variety of candidate abstractions for our framework, we rely on a top-down approach in which we (1) fix the semantics of boolean and fixpoint operators while only constraining the modal operators, and (2) consider suitable classes of abstractions to instantiate the modal operators according to the constraints. We finally show that, despite of the generality of the preservation result, the choice of the abstraction is relevant for the *monotonic* preservation of true/false evaluations along abstraction refinements.

References

1. Alur, R., Dill, D.L.: A theory of timed automata. Theoretical Computer Science 126(2), 183–235 (1994)
2. Alur, R., Henzinger, T.A., Ho, P.: Automatic symbolic verification of embedded systems. In: IEEE Real-Time Systems Symposium, pp. 2–11 (1993)
3. Alur, R., Henzinger, T.A., Lafferriere, G., Pappas, G.J.: Discrete abstractions of hybrid systems. Proc. of the IEEE 88, 971–984 (2000)
4. Bauer, K.: Three-valued μ-calculus on hybrid automata. Master's thesis, Master Thesis, University of Kaiserslautern, Department of Computer Science (2008)

5. Bensalem, S., Bouajjani, A., Loiseaux, C., Sifakis, J.: Property preserving simulations. In: von Bochmann, G., Probst, D. (eds.) CAV 1992. LNCS, vol. 663, pp. 260–273. Springer, Heidelberg (1993)
6. Davoren, J.: On hybrid systems and the modal μ-calculus. In: Antsaklis, P.J., Kohn, W., Lemmon, M.D., Nerode, A., Sastry, S.S. (eds.) HS 1997. LNCS, vol. 1567, pp. 38–69. Springer, Heidelberg (1999)
7. Davoren, J., Nerode, A.: Logics for hybrid systems. Proc. of the IEEE 88, 985–1010 (2000)
8. Fitting, M.: Kleene's three valued logics and their children. Fund. Inf. 20, 113–131 (1994)
9. Fränzle, M.: What will be eventually true of polynomial hybrid automata? In: Kobayashi, N., Pierce, B.C. (eds.) TACS 2001. LNCS, vol. 2215, pp. 340–359. Springer, Heidelberg (2001)
10. Gentilini, R., Schneider, K., Mishra, B.: Successive abstractions of hybrid automata for monotonic CTL model checking. In: Artemov, S.N., Nerode, A. (eds.) LFCS 2007. LNCS, vol. 4514, pp. 224–240. Springer, Heidelberg (2007)
11. Ghosh, R., Tiwari, A., Tomlin, C.: Automated symbolic reachability analysis with application to delta-notch signaling automata. In: Maler, O., Pnueli, A. (eds.) HSCC 2003. LNCS, vol. 2623, pp. 233–248. Springer, Heidelberg (2003)
12. Ghosh, R., Tomlin, C.J.: Lateral inhibition through delta-notch signaling: A piecewise affine hybrid model. In: Di Benedetto, M.D., Sangiovanni-Vincentelli, A.L. (eds.) HSCC 2001. LNCS, vol. 2034, pp. 232–245. Springer, Heidelberg (2001)
13. Godefroid, P., Huth, M., Jagadeesan, R.: Abstraction-based model checking using modal transition systems. In: Larsen, K.G., Nielsen, M. (eds.) CONCUR 2001. LNCS, vol. 2154, pp. 426–440. Springer, Heidelberg (2001)
14. Henzinger, M.R., Henzinger, T.A., Kopke, P.W.: Computing simulations on finite and infinite graphs. In: Proc. of 36th Ann. Symp. on Found. of Comp. Sc., p. 453. IEEE, Los Alamitos (1995)
15. Henzinger, T.: Hybrid automata with finite bisimulations. In: Fülöp, Z., Gecseg, F. (eds.) ICALP 1995. LNCS, vol. 944, pp. 324–335. Springer, Heidelberg (1995)
16. Henzinger, T.A.: The theory of hybrid automata. In: Proc. of the 11th IEEE Symp. on Logic in Comp. Science, pp. 278–292. IEEE Computer Society, Los Alamitos (1996)
17. Henzinger, T.A., Kopke, P.W., Puri, A., Varaiya, P.: What's decidable about hybrid automata? In: Proc. of the 27th Symp. on Theory of Computing, pp. 373–382. ACM, New York (1995)
18. Kannellakis, P.C., Smolka, S.A.: CCS expressions, finite state processes, and three problems of equivalence. Information and Computation 86(1), 43–68 (1990)
19. Kleene, S.C.: Introduction to Metamathematics. Wolters-Noordhoff, Groningen (1971)
20. Lafferriere, G., Pappas, G., Sastry, S.: O-minimal hybrid systems. Mathematics of Control, Signals, and Systems 13, 1–21 (2000)
21. Lafferriere, G., Pappas, J., Yovine, S.: A new class of decidable hybrid systems. In: Vaandrager, F.W., van Schuppen, J.H. (eds.) HSCC 1999. LNCS, vol. 1569, pp. 137–151. Springer, Heidelberg (1999)
22. Miller, J.: Decidability and complexity results for timed automata and semi-linear hybrid automata. In: Lynch, N.A., Krogh, B.H. (eds.) HSCC 2000. LNCS, vol. 1790, pp. 296–309. Springer, Heidelberg (2000)
23. Piazza, C., Antoniotti, M., Mysore, V., Policriti, A., Winkler, F., Mishra, B.: Algorithmic algebraic model checking i: Challenges from systems biology. In: Etessami, K., Rajamani, S.K. (eds.) CAV 2005. LNCS, vol. 3576, pp. 5–19. Springer, Heidelberg (2005)
24. Alur, C.C.R., Henzinger, T.A.: Computing accumulated delays in real-time systems. Formal Methods in System Design 11, 137–156 (1997)
25. Ratschan, S., She, Z.: Safety verification of hybrid systems by constraint propagation based abstraction refinement. In: Morari, M., Thiele, L. (eds.) HSCC 2005. LNCS, vol. 3414, pp. 573–589. Springer, Heidelberg (2005)
26. Tiwari, A., Khanna, G.: Series of abstractions for hybrid automata. In: Tomlin, C.J., Greenstreet, M.R. (eds.) HSCC 2002. LNCS, vol. 2289, pp. 465–478. Springer, Heidelberg (2002)