Undecidability Results for Hybrid Systems *

Thomas A. Henzinger† and Peter W. Kopke‡

Computer Science Department

Cornell University, Ithaca, NY 14853§

February 7, 1995

Abstract. We illuminate the boundary between decidability and undecidability for hybrid systems. Adding any of the following decorations to a timed automaton makes the reachability problem undecidable:

- a single stopwatch with weak ($\leq, \geq$) edge guards,
- a single skewed clock with variable equality tests,
- a single two-slope clock with weak edge guards,
- a single memory cell with weak edge guards.

As a corollary, we obtain undecidability for linear hybrid systems with triangular differential inclusions, which have invariants of the form $\dot{x} \leq \dot{y}$.

1 Introduction

Hybrid automata—an extension of timed automata [AD90]—have become a standard model for hybrid systems [ACHH93, AHH93, NOSY93, HH94]. There have been several decidability results [Cer92, ACHH93, KPSY93, PV94] and several undecidability results [ACHH93, KPSY93] for hybrid automata. In particular, in [PV94] it is shown that the reachability problem is decidable if the dynamics (slope, or slope interval) of all variables remain constant throughout the automaton, and the transition guards may not compare the values of variables. Both requirements are necessary: in [KPSY93], the reachability problem for hybrid automata with two three-slope variables (slopes 0, 1, and -1) is shown to be undecidable; in [ACHH93], the reachability problem with two slope-2 variables and variable equality tests. In [ACH93, BES93, KPSY93, BER94], it is shown that, under various strong side conditions, reachability is decidable for timed automata with one stopwatch, but the general problem is left open. We sharpen these results, laying bare the precise boundary between decidability and undecidability: a single stopwatch or other two-slope variable with weak transition guards causes undecidability, as does a single skewed clock with variable equality tests.

*Presented at the Workshop on *Hybrid Systems and Autonomous Control*, Ithaca, NY, October 1994.

†E-mail address: tah@cs.cornell.edu

‡Supported in part by the U.S. Army Research Office through the Mathematical Sciences Institute of Cornell University, Contract Number DAAL03-91-C-0027. E-mail address: pkpk@cs.cornell.edu

§Supported in part by the National Science Foundation under grant CCR-9200794, by the United States Air Force Office of Scientific Research under contract F49620-93-1-0056, and by the Defense Advanced Research Projects Agency under grant NAG2-892.

As a corollary, we obtain undecidability for hybrid systems with triangular differential inclusions ($\dot{x} \leq \dot{y}$).

The undecidability proofs are based on doubling the value of one clock while keeping the value of another clock constant. With this capability, we may reduce the halting problem for two-counter machines to the reachability problem for the given class of hybrid automata, rendering the latter unsolvable. Because we use only weak predicates in our constructions, the digitization technique [HMP92], used in [KPSY93, BER94] to prove decidability of certain one-stopwatch automata, does not generalize to the full class.

2 Hybrid Automata

Hybrid predicates

Let V be a finite set of elements called *hybrid variables.* A *weak hybrid constraint* is a formula of the form $x \leq n$ or $x \geq n$, where $x \in V$ and $n \in \mathbb{N}$. A *variable equality test* is a formula of the form $x = y$, where $x, y \in V$. A conjunction of weak hybrid constraints is called a *weak predicate.* The set of weak predicates is denoted $\mathcal{W}(V)$. A conjunction of weak hybrid constraints and variable equality tests is called a *weak predicate with variable equality tests*, and the set of all such is denoted $\mathcal{W}_{=}(V)$. A *weak doubling predicate with equality tests* is a formula from $\mathcal{W}_{=}(V)$ with some instances of variables x replaced by $2x$. The set of weak doubling predicates with equality tests is denoted $\mathcal{W}_{=2}(V)$. A *weak addition predicate with equality tests* is a formula from $\mathcal{W}_{=}(V)$ with some instances of variables x replaced by $x + y$, where y is also a variable. The set of weak addition predicates with equality tests is denoted $\mathcal{W}_{=+}(V)$. We will usually omit the V, and write simply $\mathcal{W}$, or $\mathcal{W}_{=2}$, etc. We refer to any of these formulas generically as a *predicate.*

Valuations

A *valuation* is a function from V into $\mathbb{R}$. Let μ be a valuation, and ϕ a predicate. We write $\mu \models \phi$, and say μ *satisfies* ϕ, if the statement obtained by replacing each variable x in ϕ by the value $\mu(x)$ is true. A *variable assignment* is a function mapping each variable either to a variable or to zero. Let μ be a valuation, and Λ a variable assignment. Define $\Lambda(\mu)$ to be the valuation ν such that for each $x \in V$, i) if $\Lambda(x) \in V$, then $\nu(x) = \mu(\Lambda(x))$; and ii) if $\Lambda(x) = 0$, then $\nu(x) = 0$.

We classify variable assignments by their actions on indiviual variables. Mapping variable x to itself is called a *pass.* Mapping variable x to 0 is called a *reset.* Mapping variable x to a variable y, where $y \neq x$, is called a *variable switch.* Now then, a *reset assignment* consists solely of passes and resets, and a *variable-to-variable assignment* consists of passes, resets, and variable switches. The sets of all reset assignments and variable-to-variable assignments are denoted respectively $\mathcal{R}(V)$ and $\mathcal{V}(V)$. We will write simply $\mathcal{R}$ and $\mathcal{V}$.

Hybrid automata

We begin by defining hybrid automata in some generality. For our undecidability results, we will not need much more power than that afforded by timed automata. Given a variable x, an *activity constraint* on x is either a differential equation in x, or a statement of the form $\dot{x} \in [a, b]$, where a and b are integers. A *hybrid automaton* consists of the following components:

- A finite set S of elements called *vertices*, or *nodes*,

- A subset S_0 of S of *initial vertices*,
- A finite set V of hybrid variables,
- A function *dif* mapping each element of $V \times S$ to an activity constraint,
- A finite set E of *edges*,
- An initial configuration $I : V \to \mathbb{Z}$.

An edge is a quadruple $(s, s', \varphi, \Lambda)$, where s and s' are vertices called respectively the *source* and *target*; φ is a predicate called the *enabling condition*; and Λ is a variable assignment called the *reset condition*. Occasionally it is convenient to equip a hybrid automaton with an invariant mapping *inv* mapping each vertex s to a predicate which must be maintained whenever control lies at s.

Let $(S, S_0, V, \mathit{dif}, E, I)$ be a hybrid automaton. We classify the variables by the activity constraints imposed by *dif*. A *clock* is a variable x such that for each $s \in S$, $\mathit{dif}(x, s)$ is $\dot{x} = 1$. A variable x is a skewed clock if there is some $g \in \mathbb{R}_+$ with $g \neq 0, 1$, such that for each $s \in S$, $\mathit{dif}(x, s)$ is $\dot{x} = g$. A variable x is a *memory cell* if for each $s \in S$, $\mathit{dif}(x, s)$ is $\dot{x} = 0$. Memory cells are interesting in conjunction with variable-to-variable assignments. A *stopwatch* is a variable x such that for each $s \in S$, $\mathit{dif}(x, s)$ is either $\dot{x} = 0$ or $\dot{x} = 1$. A variable x is a *two-slope variable* if there exist two distinct real numbers $g, h > 0$, such that for each $s \in S$, $\mathit{dif}(x, s)$ is either $\dot{x} = g$ or $\dot{x} = h$. A variable x is *linear* if for each $s \in S$, there exists an interval $[a, b]$ such that $\mathit{dif}(x, s)$ is $\dot{x} \in [a, b]$.

A *timed automaton* is a hybrid automaton in which each variable is a clock. A *multirate automaton* is a hybrid automaton in which each variable is a skewed clock whose slope is a nonnegative natural number. A hybrid automaton is *linear* if each of its variables is linear. A *linear hybrid automaton with rectangular differential inclusions* is a linear hybrid automaton such that for each variable x, $\mathit{dif}(x, s)$ is a constant function of s. A *linear hybrid automaton with triangular differential inclusions* is a linear hybrid automaton with rectangular differential inclusions equipped with an invariant mapping *inv* of a special form. The invariant *inv* must be a constant map whose value is a conjunction of inequalities of the form $\dot{x} \leq \dot{y}$, where x and y are variables. That is, a linear hybrid automaton with triangular differential inclusions is a linear hybrid automaton such that the constraint on the derivative of each variable never changes, and the invariant is a global restriction on the derivatives of the variables.

Computations of hybrid automata

Let $A = (S, S_0, V, \mathit{dif}, E, I)$ be a hybrid automaton. A *state* is a pair (s, μ), where $s \in S$ and μ is a valuation of V. We define two successor relations on the set of all states of A, an edge successor, corresponding to traversal of an edge, and a time successor, corresponding to the passage of time.

Edge successor. $(s, \mu) \triangleright (s', \nu)$ if there is an edge $e = (s, s', \phi, \Lambda) \in E$ such that $\mu \models \phi$ and $\nu = \Lambda(\mu)$. If A has an invariant mapping *inv*, we also require $\nu \models \mathit{inv}(s')$.

Time successor. $(s, \mu) \to (s, \nu)$ if there exists a $\delta \geq 0$, and for each $x \in V$, a function $f_x : [0, \delta] \to \mathbb{R}$ such that

1. $f_x(0) = \mu(x)$ and $f_x(\delta) = \nu(x)$,
2. f_x satisfies $\mathit{dif}(x, s)$.

If $dif(x,s)$ is a differential equation, then the latter restriction is clear. If $dif(x,s)$ is of the form $\dot{x} \in [a,b]$ for some a and b, then the latter restriction means $\dot{f}_x(\eta) \in [a,b]$ for each $\eta \in (0,\delta)$, and similarly for the right and left derivatives of f_x at 0 and δ respectively. If A has an invariant mapping inv, we require that for each $\eta \leq \delta$, the valuation ξ_η, assigning to each variable x the value $f_x(\eta)$, satisfies $inv(s)$.

A *computation* of A is a sequence of states $(s_n,\mu_n)_{n\in\omega}$ such that $s_0 \in S_0$, $\mu_0 = I$, and for each $n \in \mathbb{N}$, either $(s_i,\mu_i) \triangleright (s_{i+1},\mu_{i+1})$, or $(s_i,\mu_i) \rightarrow (s_{i+1},\mu_{i+1})$. The *reachability problem* for hybrid automata is given a vertex s', is there a computation $(s_n,\mu_n)_{n\in\omega}$ of A and an $n \in \mathbb{N}$ for which $s' = s_n$?

3 Undecidability Results

Reachability is decidable for linear hybrid automata with rectangular differential inclusions [PV94].[1] This is the strongest known decidability result for hybrid systems. We show that this, indeed, is the boundary of decidability. If we push farther by (1) admitting a single variable with nonconstant activities, such as a stopwatch or a two-slope variable, (2) admitting stronger enabling conditions, such as comparison of variables, (3) admitting stronger invariants, such as triangular differential inclusions, or (4) admitting variable-to-variable assignments, we obtain undecidability.

The proofs of all of these results are essentially the same. Each is based on doubling the value of one clock, while keeping the value of another clock constant. If this can be done, then a two-counter machine can be simulated by two clocks, where counter value k is represented by clock value 2^{-k}. In fact we use clock value 2^{1-k} so that we can distinguish between counter values 0 and 1 with the weak predicates ≥ 2 and ≤ 1.

We will use the following conventions in our figures. Vertices are represented by circles, and edges are represented by directed arcs between circles. The enabling condition of every edge leaving any vertex s will be of the form $\phi \wedge \bigwedge_{x\in V}(x \leq 4g(x,s))$, where $g(x,s)$ is the maximum slope allowed to variable x in vertex s by the activity constraint $dif(x,s)$. We then display only ϕ, and the reset condition Λ, above the edge, in guarded command style $\phi \rightarrow \Lambda$. Passes are omitted from the display. If the slope of a variable is left unspecified, then the variable is a clock. When a variable t can take on more than one slope, we have placed the constraint $dif(s,t)$ inside the circle denoting vertex s. We omit the constraint if it is irrelevant.

Wrapping Lemma. *Consider the hybrid automaton fragment of Figure 1. Suppose that when edge e_1 is traversed into v_1, $c = \gamma$ and $d = \delta$, where $0 \leq \gamma \leq 2g$ and $0 \leq \delta \leq 2h$. Then the next time e_4 is traversed out of v_1, c has value γ and d has value δ.*

Proof. For each variable x and edge e, define $x(e)$ to be the value of x as the edge is e traversed, before any assignment is made to x. Exactly four units of time pass between traversal of e_1 and e_4. Since the enabling condition of e_4 (and every other edge) implies $c \leq 4g$, edge e_2 must be traversed exactly once in the interim. The enabling condition of e_2 implies that it is traversed exactly $\frac{1}{g}(4g - c(e_1))$ time units after traversal of e_1. Therefore $c(e_4) = g(4 - \frac{1}{g}(4g - c(e_1)) = 4g - 4g + c(e_1) = c(e_1)$. Similarly $d(e_1) = d(e_4)$. ■

The following device allows one to test if two variables with the same slope have the same value.

[1] Decidability is retained even when arbitrary boolean combinations of weak predicates are used for enabling conditions and invariants, and nondeterministic assignments to any value within an interval $[a,b]$ are allowed in reset conditions.

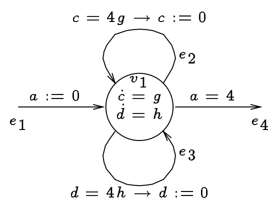

Figure 1: Wrapping lemma: both c and d retain their entry values upon exit

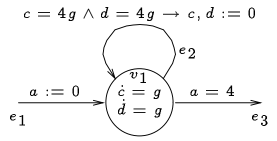

Figure 2: Equality lemma: testing $c = d$

Equality Lemma. *Consider the hybrid automaton fragment of Figure 2. Suppose that when edge e_1 is traversed into v_1, $c = \gamma$ and $d = \delta$, where $0 \leq \gamma, \delta \leq 2g$. Then edge e_3 may be traversed in the future iff $\gamma = \delta$. The next time e_3 is traversed, c and d both have the common value γ.*

Similarly, given two skewed clocks c and d, it is possible to assign to d the value of c multiplied by the ratio of their slopes.

Assignment Lemma. *Consider the hybrid automaton fragment of Figure 3. Suppose that when edge e_1 is traversed into v_1, $c = \gamma$, where $0 \leq \gamma \leq 2g$. Then the next time e_3 is traversed out of v_1, c has value γ, and d has value $\frac{h}{g}\gamma$.*

Theorem 3.1 *For every $g > 0$ with $g \neq 1$, the reachability problem for hybrid systems with*

- *one skewed clock of slope g, and any number of clocks,*
- *weak predicates with variable equality tests,*
- *reset assignments,*

is undecidable.

Proof. We reduce the halting problem for two-counter machines to this problem. Let M be a two-counter machine with counters m and n. We build a hybrid automaton with clocks c and d corresponding to m and n. For now we will assume $g = 2$. The counter value $m = k$ is represented by $c = 2^{1-k}$, and similarly for n and d. Figure 4 contains the implementation of the instruction ℓ_i: if $m = 0$ goto ℓ_j else goto ℓ_k.

Decrementing m corresponds to doubling c. We first check if $m = 0$ ($c = 2$) as in the implementation of the *if* instruction. If so, no decrement can take place, and so nothing remains to be done. If not, we proceed to the implementation contained in Figure 5. A skewed clock t of slope 2 and

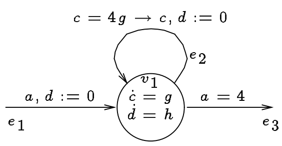

Figure 3: Assignment lemma: performing the assignment $d := \frac{h}{g}c$

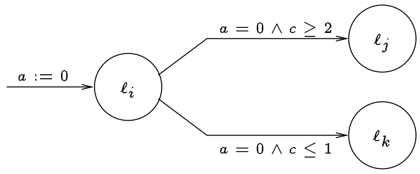

Figure 4: Implementation of ℓ_i : if $m = 0$ goto ℓ_j else goto ℓ_k

the assignment lemma construction are used to perform $t := 2c$. Then $c := t$ is performed by nondeterministically guessing a value for c, and checking the guess with a variable-variable equality test.

One of the nodes in the figure has an arc labeled with $t = 8 \to t := 0$ and $d = 4 \to d := 0$ underneath. This is a shorthand for two edges, one with enabling condition $t = 8$, resetting t, and one with enabling condition $d = 4$, resetting d. For each variable x and edge e, define $x(e)$ to be the value of x as the edge e is traversed, before any assignment is made to x. Then by two applications of wrapping, $d(e_1) = d(e_3)$, so the value of d is the same on exit as on entry to the fragment. Now let $\gamma = c(e_1)$. Then by the assignment lemma, $t(e_2) = 2\gamma$, and by wrapping, $t(e_3) = t(e_2) = 2\gamma$. Finally by the guard on e_3, $c(e_3) = 2\gamma$ as well.

To increment m, we need to halve c. To do this, we guess the half value with a clock x, assign twice this value to a clock y, using the above procedure for doubling, and then check $c = y$ using the equality lemma construction. Finally perform the assignment $c := x$ using the assignment lemma.

Notice that we can use any clock skew $g > 1$ and carry out a similar construction. First find an integer N large enough so that $N/g \leq N - 1$. Then use Ng^{-k} to encode counter value k. Then the weak predicates $c \geq N$ and $c \leq N - 1$ distinguish between $m = 0$ and $m > 1$. If $g < 1$, reverse the roles of multiplying and dividing by g. ■

Corollary 3.2 *For every $g > 0$ with $g \neq 1$, the reachability problem for hybrid systems with*

- *one skewed clock of slope g, and any number of clocks,*
- *weak predicates,*
- *variable-to-variable assignments,*

is undecidable.

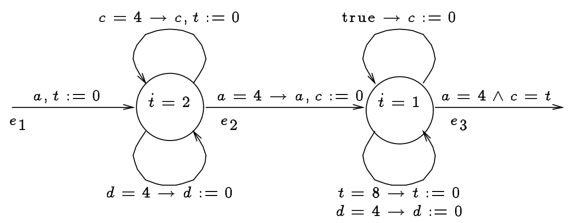

Figure 5: Doubling c using skewed clock t

Proof. After performing the assignment $t := gc$ using the assignment lemma as in Figure 5, the assignment $c := t$ can be done with a variable-to-variable assignment, rather than by guessing the value and using the equality lemma. ■

Corollary 3.3 [Alu91] *The reachability problem for hybrid systems with*

- *clocks,*
- *weak doubling predicates with variable equality tests,*
- *reset assignments,*

is undecidable. The reachability problem for hybrid systems with

- *clocks,*
- *weak addition predicates with variable equality tests,*
- *reset assignments,*

is undecidable.

Proof. For doubling predicates, in Figure 5 replace skewed clock t by clock t', replace occurrences of t in enabling conditions by $2t'$, and replace resets of t by resets of t'. Since doubling predicates are a special case of addition predicates, reachability is also undecidable for the latter. ■

Theorem 3.4 *The reachability problem for hybrid systems with*

- *one stopwatch, and any number of clocks,*
- *weak predicates,*
- *reset assignments,*

is undecidable.

Proof. The encoding of the two-counter machine is the same as before. In Figure 6, we give a hybrid automaton fragment that doubles the value of clock c while maintaining the value of clock d, using two synchronization clocks a and b, and stopwatch t. To avoid clutter, we have omitted the "reset edges" for clock d from the figure. That is, each node has an edge to itself with label $d = 4 \to d := 0$, which we have not shown in the figure. For each variable x and edge e, let

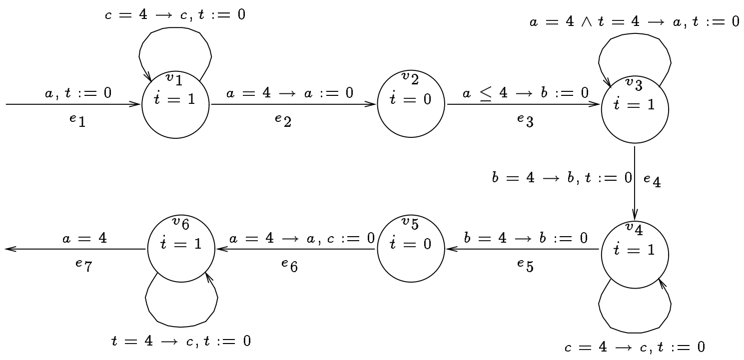

Figure 6: Doubling c using the stopwatch t

$x(e)$ be the value of x as the edge e is traversed, before any assignment is made to x. Let $\gamma = c(e_1)$ be the initial value of c. All further arithmetic will be done modulo 4. Vertex v_1 is an instance of the assignment lemma—it performs $t := c$. Hence $t(e_2) = c(e_2) = c(e_1) = \gamma$. Let $\delta = a(e_3)$ be the amount of time spent in vertex v_2. Then $c(e_3) = c(e_2) + \delta = \gamma + \delta$, and $t(e_3) = t(e_2) = \gamma$, because $\dot{t} = 0$ in v_2. Vertex v_3 is an instance of the equality lemma—it performs the test $a = t$. Therefore edge e_4 may be taken iff $\delta = a(e_3) = t(e_3) = \gamma$. Hence $c(e_3) = \gamma + \delta = 2\gamma$. By wrapping, $c(e_4) = c(e_3) = 2\gamma$. We would be done, except that the value of d has changed. Vertex v_4 is an instance of the assignment lemma—it performs $t := c$. Hence $t(e_5) = c(e_5) = c(e_4) = 2\gamma$. Since $\dot{t} = 0$ in vertex v_5, $t(e_6) = 2\gamma$ as well. Vertex v_6 is another instance of the assignment lemma—it performs $c := t$. Hence $c(e_7) = t(e_6) = 2\gamma$. But also, since $a(e_7) = 4$, the whole procedure has taken an amount of time equivalent to 0 modulo 4. Therefore by wrapping, $d(e_7) = d(e_1)$; the value of d has been maintained. ■

Theorem 3.5 *For every $g > 0$ with $g \neq 1$, the reachability problem for hybrid systems with*

- *one two-slope variable of slopes 1 and g, and any number of clocks,*
- *weak predicates,*
- *reset assignments,*

is undecidable.

Proof. Without loss of generality, we assume $g = 2$. Now doubling is provided by the automaton of Figure 7, where t is a two-slope variable, taking slopes 1 and 2. The first node is an instance of the assignment lemma, performing $t := 2c$. The second node is also an instance of the assignment lemma, performing $c := t$. ■

Corollary 3.6 *The reachability problem for linear hybrid automata with triangular differential inclusions is undecidable.*

Proof. We can redo the previous construction using linear variables s, t, and u of slope $[1, 2]$, with invariant $\dot{s} \leq \dot{t} \leq \dot{u}$. We use t to model the two-slope variable t from the previous proof. To force t to have slope 2 for the first four time units, we reset s on the incoming edge, and add the

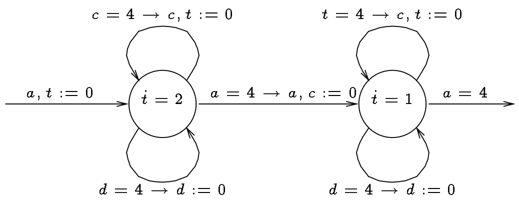

Figure 7: Doubling c using the two-slope clock t

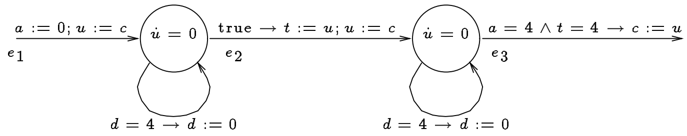

Figure 8: Doubling clock c with memory cell u

conjunct $s = 8$ to the edge leading into the node labeled $\dot{t} = 1$. To force t to have slope 1 for the second four time units, we reset u on the leading into the node labeled $\dot{t} = 1$, and add the conjunct $u = 4$ to the final edge. ■

Finally, Cerāns, in his thesis [Cer92], proves a result quite similar to the following, involving clocks that move backwards, using the doubling technique.

Theorem 3.7 *The reachability problem for hybrid systems with*

- *one memory cell, and any number of clocks,*
- *weak predicates,*
- *variable-to-variable assignments,*

is undecidable.

Proof. We use the same encoding as before. Figure 8 gives the doubling construction using memory cell u and one auxiliary clock t. The idea is to store the initial value $c(e_1)$ of c in u, and then wait nondeterministically for some number $a(t_2)$ of time units. For variable x and edge e, let $x(e)$ be the value of x as e is traversed before any assignment to x, and let $e(x)$ be the value of x as e is traversed after any assignment to x is made. Then $e_3(c) = e_2(u) = c(e_2) = c(e_1) + a(e_2)$, so it suffices to show $a(e_2) = c(e_1)$ in order to conclude $c(e_3) = 2c(e_1)$. The enabling condition of e_3 implies $e_2(a) = e_2(t)$, and therefore $a(e_2) = e_2(a) = e_2(t) = e_1(u) = c(e_1)$. ■

Summary

Figure 9 summarizes our results. The left column gives the type and number of variables used in addition to clocks. The middle column gives the allowable enabling conditions, and the last column gives the allowable reset conditions. In each case, the reachability problem is undecidable.

Hybrid Variables other than Clocks	Predicates	Assignments
none	$\mathcal{W}_{=2}$	$\mathcal{R}$
none	$\mathcal{W}_{=+}$	$\mathcal{R}$
one skewed clock	$\mathcal{W}_{=}$	$\mathcal{R}$
one skewed clock	$\mathcal{W}$	$\mathcal{V}$
one stopwatch	$\mathcal{W}$	$\mathcal{R}$
one two-slope variable	$\mathcal{W}$	$\mathcal{R}$
one memory cell	$\mathcal{W}$	$\mathcal{V}$
three linear variables with triangular diff. incl.	$\mathcal{W}$	$\mathcal{R}$

Figure 9: Undecidable reachability problems

References

[ACH93] R. Alur, C. Courcoubetis, and T.A. Henzinger. Computing accumulated delays in real-time systems. In C. Courcoubetis, editor, *CAV 93: Computer-aided Verification*, Lecture Notes in Computer Science 697, pages 181–193. Springer-Verlag, 1993.

[ACHH93] R. Alur, C. Courcoubetis, T.A. Henzinger, and P.-H. Ho. Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems. In R.L. Grossman, A. Nerode, A.P. Ravn, and H. Rischel, editors, *Hybrid Systems*, Lecture Notes in Computer Science 736, pages 209–229. Springer-Verlag, 1993.

[AD90] R. Alur and D.L. Dill. Automata for modeling real-time systems. In M.S. Paterson, editor, *ICALP 90: Automata, Languages, and Programming*, Lecture Notes in Computer Science 443, pages 322–335. Springer-Verlag, 1990.

[AHH93] R. Alur, T.A. Henzinger, and P.-H. Ho. Automatic symbolic verification of embedded systems. In *Proceedings of the 14th Annual Real-time Systems Symposium*, pages 2–11. IEEE Computer Society Press, 1993.

[Alu91] R. Alur. *Techniques for Automatic Verification of Real-time Systems*. PhD thesis, Stanford University, 1991.

[BER94] A. Bouajjani, R. Echahed, and R. Robbana. Verifying invariance properties of timed systems with duration variables. In H. Langmaack, W.-P. de Roever, and J. Vytopil, editors, *FTRTFT 94: Formal Techniques in Real-time and Fault-tolerant Systems*, Lecture Notes in Computer Science 863, pages 193–210. Springer-Verlag, 1994.

[BES93] A. Bouajjani, R. Echahed, and J. Sifakis. On model checking for real-time properties with durations. In *Proceedings of the Eighth Annual Symposium on Logic in Computer Science*, pages 147–159. IEEE Computer Society Press, 1993.

[Cer92] K. Cerāns. *Algorithmic Problems in Analysis of Real-time System Specifications*. PhD thesis, University of Latvia, 1992.

[HH94] T.A. Henzinger and P.-H. Ho. Model-checking strategies for linear hybrid systems. Presented at the Seventh International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, and at the Workshop on Hybrid Systems and Autonomous Control (Ithaca, NY), 1994.

[HMP92] T.A. Henzinger, Z. Manna, and A. Pnueli. What good are digital clocks? In W. Kuich, editor, *ICALP 92: Automata, Languages, and Programming*, Lecture Notes in Computer Science 623, pages 545–558. Springer-Verlag, 1992.

[KPSY93] Y. Kesten, A. Pnueli, J. Sifakis, and S. Yovine. Integration graphs: a class of decidable hybrid systems. In R.L. Grossman, A. Nerode, A.P. Ravn, and H. Rischel, editors, *Hybrid Systems*, Lecture Notes in Computer Science 736, pages 179–208. Springer-Verlag, 1993.

[NOSY93] X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. An approach to the description and analysis of hybrid systems. In R.L. Grossman, A. Nerode, A.P. Ravn, and H. Rischel, editors, *Hybrid Systems*, Lecture Notes in Computer Science 736, pages 149–178. Springer-Verlag, 1993.

[PV94] A. Puri and P. Varaiya. Decidability of hybrid systems with rectangular differential inclusions. In D.L. Dill, editor, *CAV 94: Computer-aided Verification*, Lecture Notes in Computer Science, pages 95–104. Springer-Verlag, 1994.