Hybrid Automata with Finite Mutual Simulations*

Thomas A. Henzinger and Peter W. Kopke

Computer Science Department
Cornell University, Ithaca, NY 14853

(tah|pkpk)@cs.cornell.edu

Abstract. Many decidability results for hybrid automata rely upon the finite region bisimulation of timed automata [AD94]. Rectangular automata do not have finite bisimulations [Hen95], yet have many decidable verification problems [PV94, HKPV95]. We prove that every two-dimensional rectangular automaton A with positive-slope variables has a finite *mutual simulation* relation, which is the intersection of the region bisimulations defined by the extremal slopes of the variables of A. While the mutual simulation is infinite for two-dimensional automata with one variable taking both positive and negative slopes, it forms a regular tesselation of the plane, and therefore can be encoded by one counter. As a corollary, we obtain the decidability of model checking linear temporal logic on these automata.

1 Introduction

Existing decision procedures for model checking hybrid automata inevitably appeal to the finite region bisimulation on timed automata [ACD90, OSY94, HH95, ACH$^+$95]. Even so, most work has concentrated on the reachability problem [ACH$^+$95], for which the existence of a finite bisimulation is unnecessary. Recently, in [PV94, HKPV95], the reachability problem for the class of rectangular hybrid automata, in which each continuous variable travels within a linear envelope, was proven decidable. These hybrid automata do not have finite bisimulations [Hen95]. Even so, every two-dimensional rectangular automaton with positive-slope variables has a finite computable *mutual simulation relation*. Mutually similar states satisfy the same $\forall CTL^*$ formulas [DGG93], and therefore model checking $\forall CTL^*$, and the familiar linear temporal logic which it subsumes, may be done on a finite quotient of the hybrid automaton. A two-dimensional rectangular automaton with one variable taking both positive and negative slopes has a mutual simulation relation that tiles the plane in a regular manner. Therefore model checking these automata may be accomplished by use of pushdown automata.

*This research 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, the Defense Advanced Research Projects Agency under grant NAG2-892, and by the U.S. Army Research Office through the Mathematical Sciences Institute of Cornell University, Contract Number DAAL03-91-C-0027.

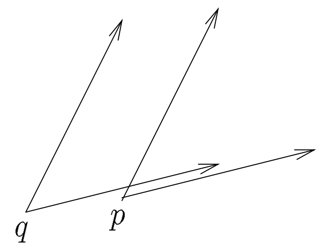

Figure 1: $cone(p)$ lies beneath $cone(q)$

2 Definitions

Let S be a set, let $\equiv$ be an equivalence relation on S, and let $\mathcal{R}$ be a set of binary relations on S. A *simulation* with respect to $\equiv$ and $\mathcal{R}$ on S [Mil71, LV92] is a binary relation $\triangleright$ on S such that

- for all $p, q \in S$, if $p \triangleright q$ then $p \equiv q$; and
- for all $p, q, q' \in S$, and every $R \in \mathcal{R}$, if qRq', then there exists a $p' \in S$ such that pRp' and $p' \triangleright q'$.

A simulation that is also an equivalence relation is called a *bisimulation*. A *mutual simulation* with respect to $\equiv$ and $\mathcal{R}$ on S is an equivalence relation $\approx$ such that $p \approx q$ implies that $p \triangleright_1 q$ and $q \triangleright_2 p$ for some simulations $\triangleright_1$ and $\triangleright_2$ on S with respect to $\equiv$ and $\mathcal{R}$.

Define $\mathbb{Z}^{-1}$ to be the set of reciprocals of integers. We fix two numbers $a, b \in \mathbb{Z} \cup \mathbb{Z}^{-1}$, with $a \leq b$, for the entire exposition. The domain of discourse is the plane region $[0,1)^2$, sometimes considered as the torus T^2, on which players 0 and 1 play a game. Each player controls a particle lying in $[0,1)^2$. A move for player i consists of changing the position of his particle, either by jumping to one of the coordinate axes, or by moving continuously to a point in $[0,1]^2$ that lies within the cone, rooted at the initial position of the particle, with rays of slope b and a. The sets $\{0\} \times [0,1]$, $\{1\} \times [0,1]$, $[0,1] \times \{0\}$, and $[0,1] \times \{1\}$ are called *boundaries*. The torus equivalence relation $\sim_{\bigcirc}$ on $[0,1]^2$ is defined by $p \sim_{\bigcirc} q$ iff $p_x - \lfloor p_x \rfloor = q_x - \lfloor q_x \rfloor$ and $p_y - \lfloor p_y \rfloor = q_y - \lfloor q_y \rfloor$. A move to one of the boundaries is considered as a move to the opposite boundary via the $\sim_{\bigcirc}$ correspondence.

For $p \in \mathbb{R}^2$, p_x denotes the first coordinate of p, and p_y denotes the second coordinate. The underlying equivalence relation will be the *cube equivalence* $\sim_{\square}$ on $[0,1)^2$, defined by $p \sim_{\square} q$ iff a) $p_x = 0$ iff $q_x = 0$; and b) $p_y = 0$ iff $q_y = 0$. Given any point $p \in [0,1)^2$, we define the *cone* of p by

$$cone(p) = \{q \in [0,1)^2 \mid q = p \text{ or } \exists p', q' \in [0,1]^2 . p \sim_{\bigcirc} p',\ q \sim_{\bigcirc} q',\ a \leq \frac{q'_y - p'_y}{q'_x - p'_x} \leq b, \text{ and } q'_x > p'_x\},$$

the set of points reachable from p by moving at a slope between a and b. We write $p \rightsquigarrow q$ iff $q \in cone(p)$. The *open cone* $ocone(p)$ is the topological interior of $cone(p)$, that is, the set of points reachable from p by moving at a slope strictly between a and b.

For $p, q \in T^2$, we say *$cone(p)$ lies beneath $cone(q)$*, and write $p\mathcal{B}q$, if $p \notin ocone(q)$ and no point on the upper boundary of $cone(q)$ lies in $cone(p)$. See Figure 1. We say *$cone(p)$ lies above $cone(q)$*, and write $p\mathcal{A}q$, if $p \notin ocone(q)$ and no point on the lower boundary of $cone(q)$ lies in $cone(p)$. We have the following lemma.

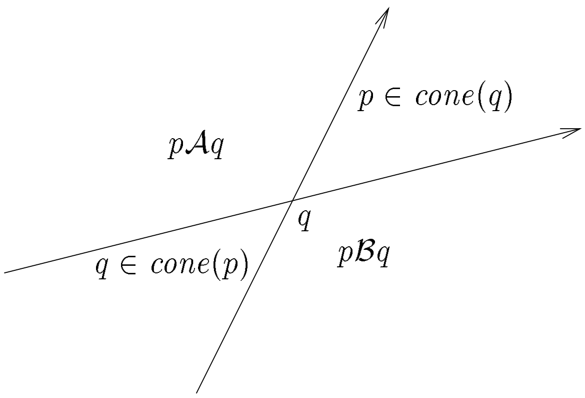

Figure 2: Illustration of Lemma 2.1: The lines through q of slopes a and b divide the plane into four regions

Lemma 2.1 *For every $p, q \in [0,1)^2$, either $p\mathcal{B}q$, $p\mathcal{A}q$, $cone(p) \subset cone(q)$, or $cone(p) \supset cone(q)$.*

For $p \in [0,1)^2$, define $R_x(p) = (0, p_y)$ and $R_y(p) = (p_x, 0)$. Here R_x is the familiar reset or jump operation of hybrid automata, which resets the first variable to zero, and similarly R_y resets the second variable to zero. Given a positive slope c, define the binary relation $\rightsquigarrow_c$ on $[0,1)^2$ by $p \rightsquigarrow_c q$ iff there is a $q' \sim_{\bigcirc} q$ such that $q'_x \geq p_x$ and $c(q'_x - p_x) = q'_y - p_y$. Define the relation $\leftarrow$ on $[0,1)^2$ by $p \leftarrow q$ iff $q = R_x(p)$. Similarly define the relation $\downarrow$ on $[0,1)^2$ by $p \downarrow q$ iff $q = R_y(p)$. The *region bisimulation* $\equiv_c$ on $[0,1)^2$ is the coarsest bisimulation on $[0,1)^2$ with respect to the relations $\{\rightsquigarrow_c, \leftarrow, \downarrow\}$ and the underlying equivalence relation $\sim_{\square}$ [AD94, ACD90, Alu91]. See Figure 3. This is the region equivalence for a two-dimensional system whose x-coordinate changes with slope 1, and whose y-coordinate changes with slope c. The equivalence classes of $\equiv_c$ are called *regions.* A region is an *interior region* if it is an open subset of the plane; otherwise it is a *boundary region.*

Lemma 2.2 *Suppose $p, q \in [0,1)^2$, $c = a$ or $c = b$, and $p \equiv_c q$. Then $R_x(p) \equiv_c R_x(q)$, and $R_y(p) \equiv_c R_y(q)$.*

3 A Finite Mutual Simulation Relation

We first give two technical lemmas. The first states that if p is beneath q, and q "moves" to q', without entering the cone of p, then every point p' reachable from p at maximum slope is beneath q'. The second gives a simple consequence of the relationship $a < 0 < b$

Lemma 3.1 *For every $p, q \in [0,1)^2$, if $p\mathcal{B}q$, $q \rightsquigarrow q'$, $q' \notin cone(p)$, and $p \rightsquigarrow_b p'$, then $p'\mathcal{B}q'$.*

Lemma 3.2 *For every $p, q \in [0,1)^2$, if $a < 0 < b$, $p\mathcal{B}q$, and $p_x > q_x$, then $p_y < q_y$.*

Proof . If $p_y \geq q_y$, then either $p \in ocone(q)$ or $p\mathcal{A}q$, both of which are impossible since $p\mathcal{B}q$. ■

In the sequel, we assume $b > 0$, for all results for $a < b \leq 0$ follow by symmetry from the corresponding results for $b > a \geq 0$. For $p, q \in [0,1)^2$, define $p \triangleright_a^b q$ if

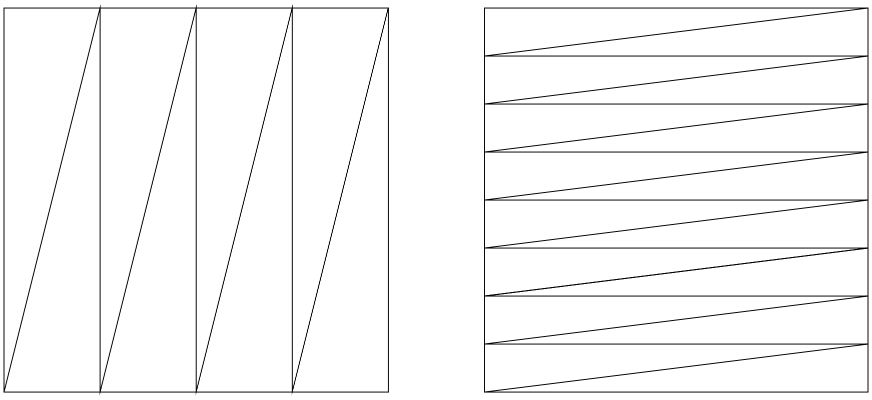

Figure 3: The region bisimulations $\equiv_4$ and $\equiv_{\frac{1}{8}}$

1. $p \sim_\square q$,
2. $p_y > q_y$ implies $R_x(p) \equiv_a R_x(q)$, and
3. $p_y < q_y$ implies $R_x(p) \equiv_b R_x(q)$, and
4. $p_x > q_x$ implies $R_y(p) \equiv_b R_y(q)$, and
5. $p_x < q_x$ implies $R_y(p) \equiv_a R_y(q)$, and
6. - $p\mathcal{A}q$ and $p \equiv_a q$, or
   - $p\mathcal{B}q$ and $p \equiv_b q$, or
   - $p \equiv_a q$ and $p \equiv_b q$.

Theorem 3.1 *The relation $\rhd_a^b$ is a simulation relation on $[0,1)^2$ with respect to the equivalence relation $\sim_\square$ and the set $\{\rightsquigarrow, \downarrow, \leftarrow\}$ of operations.*

Proof. Notice that if $p \equiv_b q$, $p_x \geq q_x$, and $p_y \leq q_y$, then by Lemma 2.2, p and q satisfy conditions 1-5 of the definition of $p \rhd_a^b q$. We will call this observation the *basic remark.* Notice also that if conditions 2-5 hold between p and q, then they also hold between $R_x(p)$ and $R_x(q)$, and between $R_y(p)$ and $R_y(q)$. We will call this the *reset remark.*

Suppose $p \rhd_a^b q$, $p\mathcal{B}q$, and $p \equiv_b q$. We use a game metaphor for our proof. Player 1 begins at p and simulates Player 0, who begins at q, and moves to q'. We show that if qRq' for some $q \in [0,1)^2$ and $R \in \{\rightsquigarrow, \downarrow, \leftarrow\}$, then there is a $p' \in [0,1)^2$ such that pRp' and $p' \rhd_a^b q'$, by distinguishing between several types of moves by Player 0. Item 1 handles the two cases $q \downarrow q'$ and $q \leftarrow q'$, while Items 2-7 handle $q \rightsquigarrow q'$. If $n < m$, then move type n takes precedence over move type m: so when we say Player 0 makes a move of type m, we mean that his move is not of any type n with $n < m$. For a move that is not of type 1 or 2, we assume that Player 0 moves from one $\equiv_b$ region to another, without passing through any intermediate $\equiv_b$ regions. A move that does pass through intermediate $\equiv_b$ regions is split into several submoves, each consisting of one change of region. The responses for Player 1 to the submoves are then concatenated into a single move. The basic strategy for Player 1 is to move at maximum slope until Player 0 moves into his cone.

1. Player 0 resets one variable (i.e., $q \downarrow q'$ or $q \leftarrow q'$)
2. Player 0 moves into $cone(p)$
3. Player 0 moves to a cube boundary
4. Player 0 moves from an interior $\equiv_b$ region to a boundary $\equiv_b$ region
5. Player 0 moves from a boundary $\equiv_b$ region to an interior $\equiv_b$ region
6. Player 0's move lies completely within an interior $\equiv_b$ region
7. Player 0's move lies entirely within a boundary $\equiv_b$ region

If Player 0 performs a reset, then Player 1 resets the same variable, arriving at the point p'. By conditions 2-5 of the definition of $p \rhd_a^b q$, either $p'\mathcal{A}q'$ and $p' \equiv_a q'$, or $p'\mathcal{B}q'$ and $p' \equiv_b q'$. Now by the reset remark, $p' \rhd_a^b q'$.

If Player 0 moves to $q' \in cone(p)$, then Player 1 also moves to q'.

If Player 0 plays a move of type 3, then Player 1 moves to the same boundary at maximum slope, reaching point p' satisfying $p' \sim_\Box q'$. By Lemma 3.1, $p'\mathcal{B}q'$. Since $p \equiv_b q$, and the equivalence classes of $\equiv_b$ are convex, $p' \equiv_b q'$. Therefore by Lemma 2.2, conditions 3 and 4 of the definition of $p' \rhd_a^b q'$ obtain. Since $q' \notin cone(p)$, it follows that $p'\mathcal{B}q'$, so condition 6 holds. Conditions 2 and 5 hold vacuously, since $p'\mathcal{B}q'$ and p' and q' lie on the same cube boundary. Therefore $p' \rhd_a^b q'$.

There are four types of boundary $\equiv_b$ region: vertical (if $b > 1$), horizontal (if $b < 1$), slope-b, and the point $(0,0)$. If Player 0 plays a move of type 4 to a vertical boundary $\equiv_b$ region, then Player 1 moves at maximum slope to the same boundary region, arriving at p'. Then $p' \equiv_b q'$, because if q^* is the point on the same boundary region such that $q \rightsquigarrow_b q^*$, then because $q \equiv_b p$, $q^* \equiv_b p'$. Because the $\equiv_b$ regions are convex, $p' \equiv_b q'$ as well Hence also $p' \sim_\Box q'$, so condition 1 of the definition of $p' \rhd_a^b q'$ obtains (we will omit mention of condition 1 in the sequel). Then $p'\mathcal{B}q'$ by Lemma 3.1, and so condition 6 holds. Since the target region is vertical, $p'_x = q'_x$. This and $p'\mathcal{B}q'$ together imply $q'_y > p'_y$. Now by the basic remark, conditions 2-5 are satisfied. Hence $p' \rhd_a^b q'$.

If Player 0 plays a move of type 4 to a horizontal boundary $\equiv_b$ region, then again Player 1 moves at maximum slope to the same boundary region, arriving at p' satisfying $p' \equiv_b q'$. Condition 6 holds between p' and q' as before. Since the target region is horizontal, $p'_y = q'_y$. This and $p'\mathcal{B}q'$ together imply $q'_x > p'_x$. Now by the basic remark, conditions 2-5 are satisfied, and so $p' \rhd_a^b q'$.

Player 0 may not play a move of type 4 to a $\equiv_b$ boundary region of slope b without entering $cone(p)$. Similarly, Player 0 may not move to the point $(0,0)$ without entering $cone(p)$, and so the analysis of move type 4 is complete.

Move type 5 also dissolves into four cases, the same two impossible, based upon the slope of the boundary $\equiv_b$ region. If Player 0 begins at q on a vertical boundary region, then $p_x = q_x$, and $p\mathcal{B}q$ implies $p_y \leq q_y$. Player 1 moves at maximum slope b to a point p' with $p'_x = q'_x$ and $p'_y \leq q'_y$. Then $p' \equiv_b q'$, because if Player 1 crossed any horizontal region boundary by moving to q', then $p'_y \leq q'_y$ implies Player 0 crossed this boundary as well, a contradiction, and if Player 1 crossed any vertical region boundary by moving to q', then $p'_x = q'_x$ implies Player 0 crossed this boundary as well, again a contradiction. By Lemma 3.1, $p'\mathcal{B}q'$, so condition 6 holds, and by the basic remark, so do conditions 2-5.

If Player 0 begins at q on a horizontal boundary region, then $p_y = q_y$, and $p\mathcal{B}q$ implies $p_x \geq q_x$. This case is slightly more difficult than the previous, because Player 1 may not be able to move at maximum slope to a point p' with $p' \equiv_b q'$ and $p'_y = q'_y$, because the right cube boundary may be too close to p. If he can, then by the same analysis as in the previous case, $p' \triangleright_a^b q'$. Now suppose no such move is available. If $q'_x \geq p_x$, then Player 1 moves at maximum slope b to a point p' with $p' \equiv_b q'$, $p'_x = q'_x$, and $p'_y \leq q'_y$. Then by Lemma 3.1 and the basic remark, $p' \triangleright_a^b q'$. Finally, suppose $q'_x < p_x$. Notice that in this case it must be that $a \geq 0$, by Lemmas 3.1 and 3.2. Therefore $q'_y > p_y$. Player 1 now makes a very small move at maximum slope—small enough so that he reaches a point p' with $p'_y < q'_y$, $p'_x \geq q'_x$, and $p' \equiv_b q'$. By the basic remark, $p' \triangleright_a^b q'$.

We deal with case 6 in a hierarchical manner.

- Suppose $p_x > q'_x$. Then Player 1 remains at p. Since $q' \equiv_b q$ and $q \equiv_b p$, $p \equiv_b q'$. By Lemma 3.1, $p\mathcal{B}q'$. So condition 6 is satisfied between p and q'. By Lemma 2.2, $R_y(p) \equiv_b R_y(q')$, and so conditions 4 and 5 are satisfied.
  - If $a < 0 < b$, then by Lemma 3.2, $p_y \leq q'_y$; so by the basic remark, $p \triangleright_a^b q'$.
  - If $0 \leq a \leq b$, then $q'_y \geq q_y$.
    * If $p_y \leq q_y$, then $p_y \leq q'_y$, so by the basic remark, $p \triangleright_a^b q'$.
    * If $p_y > q_y$, then because $p \triangleright_a^b q$, it must be that $R_x(p) \equiv_a R_x(q)$. Since $q_y \leq q'_y \leq p_y$, $R_x(p) \equiv_a R_x(q')$ as well. So conditions 2 and 3 are satisfied, and $p \triangleright_a^b q'$.
- Now suppose $p_x \leq q'_x$. Then Player 1 moves at maximum slope to a point p' such that $p'_x = q'_x$, $p' \equiv_b q'$, and $p'\mathcal{B}q'$. By the basic remark, $p' \triangleright_a^b q'$.

At last we reach move type 7. Player 0 may not begin in a vertical region, for then he could not move and remain in the region. He may not begin in a slope-b region, for then p is in the same slope-b region, in which case it cannot be that $p\mathcal{B}q$. If Player 0 begins on a horizontal region, then it must be that $p_y = q_y$ and $p_x \geq q_x$. Notice $p_y = q'_y$ as well, because q' lies in the same horizontal region. Since $q \notin cone(p)$, it must be that $q'_x < p_x$. In this case Player 1 does not move. By the basic remark, $p \triangleright_a^b q'$.

We have now finished the case $p\mathcal{B}q$ and $p \equiv_b q$. A symmetric argument handles the case $p\mathcal{A}q$ and $p \equiv_a q$. Now suppose $p \equiv_a q$ and $p \equiv_b q$, while neither $p\mathcal{A}q$ nor $p\mathcal{B}q$ obtains. If Player 0 performs a reset, then Player 1 resets the same variable, arriving at a point p', which, by Lemma 2.2, satisfies $p' \equiv_a q'$ and $p' \equiv_b q'$. Conditions 1-5 are immediately satisfied, and so $p' \triangleright_a^b q'$. If Player 0 moves into $cone(p)$, then Player 1 may move to q', and of course $q' \triangleright_a^b q'$. Otherwise, by Lemma 2.1, $p \in cone(q')$. It must then be that p and q do not lie in a boundary $\equiv_a$ region or a boundary $\equiv_b$ region. For otherwise either $p\mathcal{A}q$ or $p\mathcal{B}q$ obtains.

The line segment from q to q' must do one of the following:

1. cross a horizontal region boundary,
2. cross a vertical region boundary,
3. cross a slope-a region boundary,
4. cross a slope-b region boundary, or
5. lie completely within one $\equiv_a$ region and one $\equiv_b$ region.

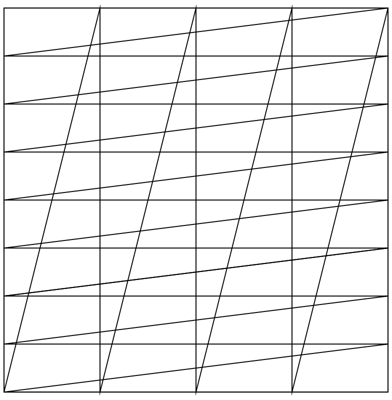

Figure 4: The mutual simulation relation $\triangleright^{4}_{\frac{1}{8}}$

For cases 1-4, let the crossing point be q^*. Player 1 will first make a submove to a point p^* such that $p^* \triangleright^b_a q^*$, and then play as dictated by the strategy against a move from q^* to q'. If 1, then Player 1 moves to this region, at minimal slope if the y-coordinate of the region is less than p_y, at maximal slope the y-coordinate of the region is greater than p_y. If 2, then Player 1 move to the same region, at minimum or maximum slope, so as to minimize $|p^*_y - q^*_y|$. If 3, then Player 1 moves at maximum slope b to the same boundary. If 4, then Player 1 moves at minimum slope a to the same boundary. If 5, then Player 1 does not move (i.e., $p^* = p$). In each case, $p^* \equiv_a q^*$ and $p^* \equiv_b q^*$. ■

Define $\equiv^b_a$ on $[0,1)^2$ to be the intersection of $\equiv_a$ and $\equiv_b$. I.e., $p \equiv^b_a q$ iff $p \equiv_a q$ and $p \equiv_b q$.

Corollary 3.2 *The intersection $\equiv^b_a$ of the two bisimulations $\equiv_a$ and $\equiv_b$ is the coarsest finite mutual simulation relation on $[0,1)^2$.*

Proof. Lemma 2.2 and Condition 6 in the definition of $\triangleright^b_a$ show that $\equiv^b_a$ is a mutual simulation relation. To see that it is the coarsest, suppose $p \not\equiv_b q$. Suppose Player 0 begins at position q, and Player 1 begins at position p. If $\lfloor bq_x \rfloor < \lfloor bp_x \rfloor$, then by performing a $\downarrow$ move (i.e., resetting y), and then moving at maximum slope, Player 0 can make more consecutive $\rightsquigarrow$ moves to the boundary $[0,1] \times \{1\}$ than can Player 1. If $\lfloor bp_x \rfloor = \lfloor bp_y \rfloor$, we have two cases, predicated on how many players lie on or on either side of the line defined by the equation $y = bx - b\lfloor bp_x \rfloor$. If Player 0 is above this line, and Player 1 is on or below, then Player 0 can make more consecutive moves to the boundary $[0,1) \times \{1\}$ by playing at maximum slope. If Player 0 is on the line, and Player 1 is below, then Player 0 can make more consecutive moves to the boundary $[0,1] \times \{1\}$ by playing at maximum slope. The case $p \not\equiv_a q$ is handled similarly. ■

The mutual simulation $\triangleright^{4}_{\frac{1}{8}}$ is shown in Figure 4.

Let $\Delta \subset \mathbb{R}$ be a set of slopes. Define $\rightsquigarrow_\Delta$ on $[0,1)^2$ by $p \rightsquigarrow_\Delta q$ iff for some $c \in \Delta$, $p \rightsquigarrow_c q$. If $a = \inf \Delta$ and $b = \sup \Delta$, then any $\rightsquigarrow_\Delta$ move may be implemented as one $\rightsquigarrow_a$ move followed by one $\rightsquigarrow_b$ move. Hence we have the following corollary.

Corollary 3.3 *Suppose $\Delta \subset \mathbb{R}$ contains $a = \inf \Delta$ and $b = \sup \Delta$. with $b \in \mathbb{N}$ and either $a \in \mathbb{Z}$ or $a^{-1} \in \mathbb{N}$. Then the relation $\equiv_a^b$ is the coarsest mutual simulation on $[0,1)^2$ with respect to the equivalence relation $\sim_\square$ and the set $\{\rightsquigarrow_\Delta, \downarrow, \leftarrow\}$ of operations.*

4 Model Checking LTL on Two-Dimensional Rectangular Automata with Positive Slopes

Two-Dimensional Rectangular Hybrid Automata

Let V be a set of variables. An *atomic proposition over V* is a boolean combination of formulas of the form $x \sim c$ or $y \sim c$, where $x \in V$, c is an integer, and $\sim \in \{<, >, \leq, \geq\}$. The set of all atomic propositions is denoted $\mathcal{P}_V$. A *rectangular hybrid automaton* is a hybrid automaton with variable set V such that (1) every edge guard and location invariant is an atomic predicate over V; (2) the constraint on the derivative of each variable x_i is of the form $\dot{x}_i \in [a_{x_i}, b_{x_i}]$, where $a_{x_i}, b_{x_i} \in \mathbb{Q}$, and moreover this constraint is the same at every automaton location; and (3) variables may be assigned only to integer values. A variable x_i is *positive* if $a_{x_i} \geq 0$. A *two-dimensional positive rectangular hybrid automaton* is a rectangular hybrid automaton with exactly two continuous variables x and y, both positive, such that $\frac{b_y}{a_x}, \frac{a_y}{b_x} \in \mathbb{N} \cup \mathbb{N}^{-1}$ (where $\mathbb{N}^{-1}$ is the set of reciprocals of natural numbers).

Let A be a 2D positive rectangular hybrid automaton. A *state* of A is a triple (u, x, y), where u is a location of A and $x, y \in \mathbb{R}_{\geq 0}$. The set of states of A is denoted Σ_A. A *run* of A is a partial function $\rho \colon \mathbb{N} \times \mathbb{R}_{\geq 0} \to \Sigma_A$ such that

- $\rho(0,0)$ is an initial state
- for each $n \in \mathbb{N}$, $domain(\rho) \cap \{n\} \times \mathbb{R}_{\geq 0}$ is $\{n\} \times [0, t_n]$ for some $t_n \in \mathbb{R}_{\geq 0}$
- for each $n \in \mathbb{N}$, the function $f : [0, t_n] \to \mathbb{R}^2_{\geq 0}$, defined by omitting the location component of $\lambda s.\rho(n, s)$, satisfies $\dot{f}_1 \in [a_x, b_x]$ and $\dot{f}_2 \in [a_y, b_y]$, where $f = (f_1, f_2)$
- for each $n \in \mathbb{N}$, $\rho(n+1, 0)$ follows from $\rho(n, t_n)$ by traversal of an edge.

The elements of $domain(\rho)$ are called *positions*, and the positions are linearly ordered lexicographically.

We extend the cube equivalence relation $\sim_\square$ to Σ_A by $(u, x, y) \sim_\square (u', x', y')$ iff $u = u'$, and for every atomic proposition p, (x, y) satisfies p iff (x', y') satisfies p. An equivalence class of $\sim_\square$ is called a *cube region.*

Linear Temporal Logic

Fix a 2D rectangular hybrid automaton A. For each location u of A, we introduce a new atomic proposition p_u, which is satisfied by a state (v, x, y) iff $v = u$. Let $\mathcal{P}'_V = \mathcal{P}_V \cup \{p_u | u \in Q\}$, where Q is the set of locations of A. The formulas of linear temporal logic (*LTL*) are generated by the following grammar:

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \phi \mid \phi \mathcal{U} \phi$$

Here p may be any element of $\mathcal{P}'_V$. The formulas of *LTL* are interpreted over runs.

Let ρ be a run of A. For every position (n, t), let $\rho^{(n,t)}$ be the trace of A defined by ignoring every position of ρ before (n, t). I.e.,

$$\rho^{(n,t)}(m, s) = \begin{cases} (0, s-t) & \text{if } m = n \\ (m-n, s) & \text{if } m > n \end{cases}$$

We give the significant clause of the inductive definition of satisfaction.

$$\rho \models_A \psi_1 \mathcal{U} \psi_2 \quad \text{iff} \quad \begin{array}{l} \text{there is a position } (n,t) \text{ such that } \rho^{(n,t)} \models_A \psi_2, \\ \text{and for every position } (m,s) < (n,t),\ \rho^{(m,s)} \models_A \psi_1 \vee \psi_2. \end{array}$$

The *model-checking problem* for *LTL* on a class $\mathcal{C}$ of hybrid automata is, given a hybrid automaton $A \in \mathcal{C}$, and given an *LTL* formula ϕ, does every run of A satisfy ϕ?

Corollary 4.1 *The model-checking problem for LTL on two-dimensional positive rectangular hybrid automata is decidable.*

Proof . Let A be a two-dimensional rectangular hybrid automaton, and let ϕ be a formula of *LTL*. Let M be the absolute value of the largest constant appearing in ϕ or in any edge guard, location invariant, or variable assignment of A. Let $\sim_\Box^M$ be the equivalence relation on Σ_A defined by $(u, x, y) \sim_\Box (u', x', y')$ iff $u = u'$ and for every atomic proposition p using no constant N with $|N| > |M|$, (x, y) satisfies p iff (x', y') satisfies p. Define $(u, x, y) \rightsquigarrow_A (u', x', y')$ iff (u', x', y') follows from (u, x, y) by a time-step or by traversal of an edge.

Define $(u, x, y) \approx_\Box^M (u', x', y')$ iff $(u, x, y) \sim_\Box^M (u', x', y')$ and $(x - \lfloor x \rfloor, y - \lfloor y \rfloor) \equiv_a^b (x' - \lfloor x' \rfloor, y' - \lfloor y' \rfloor)$. That is, the states are indistinguishable without constants larger than M, and their fractional parts lie in the same mutual simulation class. Then by Corollary 3.2, the equivalence relation $\approx_\Box^M$ is a finite mutual simulation relation on Σ_A with respect to the equivalence relation $\sim_\Box^M$ and the operation $\rightsquigarrow_A$.

Define a finite automaton $\hat{A}$ whose state space is the set of equivalence classes of $\approx_\Box^M$, and whose transition relation $\rightarrow_{\hat{A}}$ is defined by $\mu \rightarrow_{\hat{A}} \mu'$ iff there are states $(u, x, y) \in \mu$ and $(u', x', y') \in \mu'$ such that $(u, x, y) \rightsquigarrow_A (u', x', y')$. Then by a theorem of [DGG93], all runs of A satisfy ϕ iff all runs of $\hat{A}$ satisfy ϕ. ■

A *bounded-constraint automaton* is a hybrid automaton with the same properties as a rectangular hybrid automaton, except that the continuous behavior of each variable x_i is globally constrained (i.e., the constraint is the same in each automaton location) to satisfy $\dot{x}_i \in \Delta_i$ during time steps, where Δ_i is any set containing its inf and sup. A variable x_i is *positively constrained* if $\Delta_i \subset \mathbb{R}_{\geq 0}$. A *two-dimensional positive bounded-constraint automaton* is a bounded-constraint automaton A with exactly two continuous variables, both positively constrained, such that $a = \frac{\min \Delta_2}{\max \Delta_1}$ and $b = \frac{\max \Delta_2}{\min \Delta_1}$ are elements of $\mathbb{N} \cup \mathbb{N}^{-1}$. We have the following counterpart to Corollary 3.3.

Corollary 4.2 *The model-checking problem for LTL on two-dimensional positive bounded-constraint automata is decidable.*

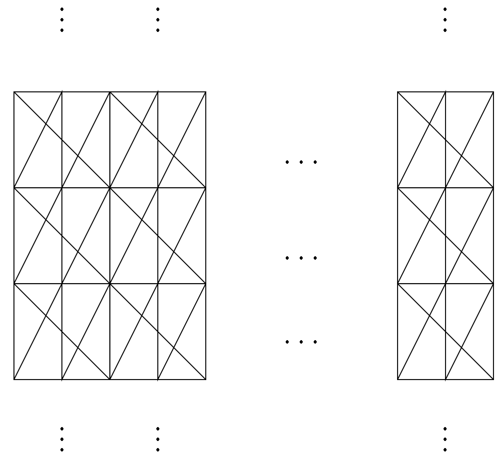

Figure 5: The Infinite Mutual Simulation $\approx_{\Box}^{M}$

5 Model Checking Linear Temporal Logic on Two-Dimensional Rectangular Automata with Positive and Negative Slopes

When both positive and negative slopes are considered, it no longer suffices to consider equivalent all points in the plane outside of a bounded box. If the minimal slope a is a negative integer, then the mutual simulation relation tesselates the strip $\{p \in \mathbb{R}^2 \mid -M \leq p_x \leq M\}$, where M is the largest relevant constant. Consequently a pushdown automaton [CG77] may be used to simulate a 2D rectangular automaton with only one positive variable. This technique was used in [BR95] to verify untimed properties of two-dimensional hybrid automata with one clock, and one variable whose slope is constrained to be one particular element of $\{-1, 0, 1\}$ in each location.

A *two-dimensional rectangular hybrid automaton with positive and negative slopes* is a rectangular hybrid automaton with exactly two continuous variables x and y, one positive, such that $\frac{b_y}{a_x}, \frac{a_y}{b_x} \in \mathbb{Z} \cup \mathbb{Z}^{-1}$.

Corollary 5.1 *The model-checking problem for LTL on two-dimensional rectangular automata with positive and negative slopes is decidable.*

Proof . Let A be a two-dimensional rectangular hybrid automaton with positive and negative slopes, and let ϕ be a formula of *LTL*. Let x be the positive variable of A. Let M be the absolute value of the largest constant appearing in ϕ or in any edge guard, location invariant, or variable assignment of A. Let $\sim_{\Box}^{M}$ be the equivalence relation on Σ_A defined by $(u, x, y) \sim_{\Box} (u', x', y')$ iff $u = u'$ and for every atomic proposition p comparing x against no constant N with $|N| > |M|$, (x, y) satisfies p iff (x', y') satisfies p. Define the operation $\rightsquigarrow_A$ as in the proof of Theorem 4.1. Define $(u, x, y) \approx_{\Box}^{M} (u', x', y')$ iff $(u, x, y) \sim_{\Box}^{M} (u', x', y')$ and $(x - \lfloor x \rfloor, y - \lfloor y \rfloor) \equiv_a^b (x' - \lfloor x' \rfloor, y' - \lfloor y' \rfloor)$. Then by Corollary 3.2, the equivalence relation $\approx_{\Box}^{M}$ is an infinite mutual simulation relation on Σ_A with respect to the equivalence relation $\sim_{\Box}^{M}$ and the operation $\rightsquigarrow_A$. The equivalence classes of $\approx_{\Box}^{M}$ tile the strip $\{p \in \mathbb{R}^2 \mid -M \leq p_x \leq M\}$, replicating the cube mutual simulation $\equiv_a^b$ on each unit cube in the strip. See Figure 5. Therefore we may build an ω-pushdown automaton $\hat{A}$ which generates all sequences of $\approx_{\Box}^{M}$-equivalence classes generated by runs of A. The discrete state of the automaton gives the $\equiv_a^b$-equivalence class, and the length of the stack gives the integer part of y. We use $\{-\infty, -M, -M+1, \ldots, -1, 0, 1, \ldots, M, \infty\}$ for the stack alphabet.

Since ϕ describes an ω-regular property, the model-checking problem is reduced to the inclusion problem between ω-pushdown automata and ω-automata. The latter is decidable [CG77], and so the proof is complete. ■

We may now strengthen Corollary 4.2. A *two-dimensional bounded-constraint automaton with positive and negative slopes* is a bounded-constraint automaton with exactly two continuous variables, one positively constrained, such that $a = \frac{\min \Delta_2}{\max \Delta_1}$ and $b = \frac{\max \Delta_2}{\min \Delta_1}$ are elements of $\mathbb{Z} \cup \mathbb{Z}^{-1}$.

Corollary 5.2 *The model-checking problem for LTL on two-dimensional bounded-constraint automata with positive and negative slopes is decidable.*

References

[ACD90] R. Alur, C. Courcoubetis, and D.L. Dill. Model checking for real-time systems. In *Proceedings of the Fifth Annual Symposium on Logic in Computer Science*, pages 414–425. IEEE Computer Society Press, 1990.

[ACH+95] R. Alur, C. Coucoubetis, N. Halbwachs, T.A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138:3–34, 1995.

[AD94] R. Alur and D.L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.

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

[BR95] A. Bouajjani and R. Robbana. Verifying ω-regular properties for subclasses of linear hybrid systems. Submitted for publication, 1995.

[CG77] R.S. Cohen and A.Y. Gold. Theory of ω-languages. i: Characterizations of ω-context-free languages. *Journal of Computer and System Sciences*, 15:169–184, 1977.

[DGG93] D. Dams, O. Grumberg, and R. Gerth. Generation of reduced models for checking fragments of ctl. In C. Courcoubetis, editor, *CAV 93: Computer-aided Verification*, Lecture Notes in Computer Science 697, pages 479–490. Springer-Verlag, 1993.

[Hen95] T.A. Henzinger. Hybrid automata with finite bisimulations. To appear at ICALP, 1995.

[HH95] T.A. Henzinger and P.-H. Ho. Algorithmic analysis of nonlinear hybrid systems. Submitted, 1995.

[HKPV95] T.A. Henzinger, P. Kopke, A. Puri, and P. Varaiya. What's decidable about hybrid automata. To appear at STOC, 1995.

[LV92] N.A. Lynch and F. Vaandrager. Forward and backward simulations for timing-based systems. In J.W. de Bakker, K. Huizing, W.-P. de Roever, and G. Rozenberg, editors, *Real Time: Theory in Practice*, Lecture Notes in Computer Science 600, pages 397–446. Springer-Verlag, 1992.

[Mil71] R. Milner. An algebaic definition of simulation between programs. In *Second International Joint Conference on Artificial Intelligence*, pages 481–489. The British Computer Society, 1971.

[OSY94] A. Olivero, J. Sifakis, and S. Yovine. Using abstractions for the verification of linear hybrid systems. In D.L. Dill, editor, *CAV 94: Computer-aided Verification*, Lecture Notes in Computer Science 818, pages 81–94. Springer-Verlag, 1994.

[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.