International Journal of Foundations of Computer Science

The Existence of ω-Chains for Transitive Mixed Linear Relations and Its Applications

Zhe Dang*
School of Electrical Engineering and Computer Science,
Washington State University, Pullman, Washington 99164, USA

and

Oscar H. Ibarra
Department of Computer Science,
University of California, Santa Barbara, California 93106, USA

Received (received date)
Revised (revised date)
Communicated by Editor's name

ABSTRACT

We show that it is decidable whether a transitive mixed linear relation has an ω-chain. Using this result, we study a number of liveness verification problems for generalized timed automata within a unified framework. More precisely, we prove that (1) the mixed linear liveness problem for a timed automaton with dense clocks, reversal-bounded counters, and a free counter is decidable, and (2) the Presburger liveness problem for a timed automaton with discrete clocks, reversal-bounded counters, and a pushdown stack is decidable.

Keywords: Mixed linear relations; ω-chains; timed automata; liveness; safety.

1. Introduction

In the area of model-checking, the search for efficient techniques for verifying infinite-state systems has been an ongoing research effort. Much work has been devoted to investigating various restricted models of infinite-state systems that are amenable to automatic verification for some classes of temporal properties, e.g., safety and liveness. A timed automaton is one such model. A timed automaton [2] is a finite automaton (over finitely many control states) augmented with dense clocks. The clocks can be reset or progress at the same rate, and can be tested against clock constraints in the form of clock regions (i.e., comparisons of a clock or the difference of two clocks against an integer constant, e.g., $x - y < 6$, where x and y are clocks.). The most important result in the theory of timed automata is that region reachability for timed automata is decidable [2]. This result has

*Corresponding author (zdang@eecs.wsu.edu).

been used in defining various real-time logics, model checking algorithms and tools [1, 3, 16, 17, 21, 22, 23, 25] for verifying real-time systems.

However, region reachability is not strong enough to verify many complex timing requirements not in the form of clock regions (e.g., "$x_1 - x_2 > 2(x_3 - x_4)$ is always true") for timed automata. Recently, decidable binary reachability (i.e., the set of all pairs of configurations such that one can reach the other) characterizations for timed automata and their generalizations were obtained [8, 9, 10]. The characterizations opened the door for automatic verification of various real-time models against complex timing requirements. For instance, a flattening technique was used by Comon and Jurski [8] to establish that the binary reachability of timed automata is definable in the additive theory of the reals and integers. A timed automaton can be augmented with other unbounded discrete data structures such as a free counter and reversal-bounded counters. A (free) counter is an integer variable that can be incremented by 1, decremented by 1, and tested against 0. A counter is reversal-bounded if the number of times it alternates between nondecreasing and nonincreasing mode and vice-versa is bounded by some fixed number independent of the computation [19]. A pattern technique was proposed by Dang [9] to obtain a decidable binary reachability characterization on some "storage-augmented" timed automata. For instance, suppose that $\mathcal{A}$ is a timed automaton (with dense clocks x_1 and x_2) augmented with two reversal-bounded counters y_1 and y_2, and a free counter y_3. The result of Dang [9] implies that the binary reachability of $\mathcal{A}$ is definable in the additive theory of the reals and integers. Therefore, we can automatically verify the following safety property, which contains linear constraints on *both* dense variables and unbounded discrete variables,

> "Given two control states s_1 and s_2, if $\mathcal{A}$ starts at s_1 in a configuration satisfying $x_1 - 2x_2 + y_1 - 2y_2 + y_3 > 5$, then whenever $\mathcal{A}$ reaches s_2, its configuration must satisfy $x_1 + x_2 < y_2 - 2y_3 + 2$."

In contrast to safety properties, liveness properties considered in this paper involve properties on infinite executions of $\mathcal{A}$. For instance, consider an infinite execution that passes some control state for infinitely many times. A mixed linear constraint on clocks and counters in $\mathcal{A}$ may or may not be satisfied whenever $\mathcal{A}$ passes the control state. Is there an infinite execution on which the constraint is satisfied for infinitely many times at the control state? An example liveness property would be like below:

> "Given two control states s_1 and s_2, if $\mathcal{A}$ starts at s_1 in some configuration satisfying $x_1 - 2x_2 + y_1 - 2y_2 + y_3 > 5$, then $\mathcal{A}$ has an infinite execution on which $x_1 + x_2 < y_2 - 2y_3 + 2$ is satisfied at s_2 for infinitely many times."

This kind of liveness properties have a lot of applications such as whether concurrent real-time processes are livelock-free, starvation-free, etc. Can this liveness property be automatically verified for $\mathcal{A}$?

We approach this question by looking at mixed linear relations R that are relations on real and integer variables definable in the additive theory of the reals and integers. We first prove the main theorem that the existence of an ω-chain for R is decidable when R is transitive. This proof is done by eliminating quantifiers from R using a recent result of [24] and expressing R into mixed linear constraints. The decidable result follows from the fact that the existence of an ω-chain for R forces R to have a special format. Notice that the transitivity of R is critical; removing it from R obviously causes the existence of an ω-chain undecidable (e.g., encoding the one-step transition relations of a two-counter machine into R).

Recall that the binary reachability of $\mathcal{A}$ is a transitive mixed linear relation. The above liveness question can be reduced to the existence of an ω-chain for some mixed linear relation easily constructed from the binary reachability. Therefore, a direct application of the main theorem gives a positive answer to the question. We may also use the main theorem to verify a class of pushdown systems. For instance, suppose that $\mathcal{P}$ is a pushdown automaton. Consider the following Presburger liveness property:

> "Given two states s_1 and s_2, from some configuration at s_1 satisfying $n_a - 2n_b > n_c$, $\mathcal{P}$ has an infinite execution on which $n_a + n_b < 3n_c$ holds at s_2 for infinitely many times,"

where count variable n_a indicates the number of symbol a's in the stack word in a configuration. This paper provides a technique to reduce this property into the existence of an ω-chain for some Presburger relation, which is a special form of mixed linear relations. Therefore, using the main theorem, the above property can be automatically verified for $\mathcal{P}$. In fact, we show the result for a more powerful class of pushdown systems: $\mathcal{P}$ can be a pushdown automaton augmented with reversal-bounded counters and integer-valued clocks. This class of pushdown systems can be used to model a class of real-time recursive programs. The Presburger liveness properties for this class of pushdown systems then contain Presburger formulas on count variables, reversal-bounded counters and discrete clocks.

The techniques presented in this paper are different from our previous papers [12, 11] on liveness verification. In those two papers, we only deal with the Presburger liveness problems for discrete timed automata (i.e., timed automata with integer-valued clocks) [12] and for reversal-bounded counter machines with a free counter (NCMFs) [11], respectively. Both of the papers are based upon analyzing loops in the machines. In particular, the key idea in [12] is to make discrete timed automata static (i.e., enabling conditions can be removed) and memoryless (i.e., two integer clock values are somewhat unrelated if they are separated by a large number of clock resets). But, the idea cannot be easily extended to dense clocks. The key idea in [11] is to partition an execution of an NCMF into phases such that reversal-bounded counters are monotonic in each phase. Then, a technique is used to reduce the NCMF into one with only one free counter, with respect to the liveness property. But, we were not able to extend the idea when the free counter is replaced by a pushdown stack. The techniques presented in this paper, however, allows us to handle, in a unified framework, a stronger class of systems: timed automata with dense clocks, reversal-bounded counters, and a free counter. In addition, we can deal with a class of generalized pushdown systems.

The paper is organized as follows. Section 2 gives the basic definitions and preliminary results that are used in the paper. Sections 3 through 5 present the proof of the main theorem; i.e., it is decidable whether a transitive mixed linear relation has an ω-chain. Section 6 applies the main theorem in showing the decidable results on the mixed linear liveness problem for a timed automaton augmented with reversal-bounded counters and a free counter and on the Presburger liveness problem for a discrete timed automaton augmented with reversal-bounded counters and a pushdown stack. Finally, Section 7 concludes with some remarks.

2. Preliminaries

Let m and n be positive integers. Consider a formula

$$\sum_{1\leq i\leq m} a_i x_i + \sum_{1\leq j\leq n} b_j y_j \sim c,$$

where each x_i is a real variable, each y_j is an integer variable, each a_i, each b_j and c are integers, $1 \leq i \leq m, 1 \leq j \leq n$, and $\sim$ is $=$, $>$, or $\equiv_d$ for some integer $d > 0$. The formula is a *mixed linear constraint* if $\sim$ is $=$ or $>$. The formula is called a *dense linear constraint* if $\sim$ is $=$ or $>$ and each $b_j = 0$, $1 \leq j \leq n$. The formula is called a *discrete linear constraint* if $\sim$ is $>$ and each $a_i = 0$, $1 \leq i \leq m$. The formula is called a *discrete mod constraint*, if each $a_i = 0$, $1 \leq i \leq m$, and $\sim$ is $\equiv_d$ for some integer $d > 0$.

A formula is *definable in the additive theory of reals and integers (resp. reals, integers)* if it is the result of applying quantification ($\exists$) and Boolean operations ($\neg$ and $\wedge$) over mixed linear constraints (resp. dense linear constraints, discrete linear constraints); the formula is called a *mixed formula (resp. dense formula, Presburger formula)*. It is decidable whether the formula is satisfiable. It is well-known that a Presburger formula can always be written, after quantifier elimination, as a disjunctive normal form of discrete linear constraints and discrete mod constraints. It is also known that a dense formula can always be written as a disjunctive normal form of dense linear constraints. Can we eliminate quantifiers in mixed formulas? The answer is not obvious. This is because a mixed formula like $\exists y(x_1 - x_2 = y)$, after eliminating all the quantifiers, is not always in the form of a Boolean combination of mixed linear constraints.

A real variable x can be treated as the sum of an integer variable (the integral part of x) $x^{\mathbf{Int}}$ and a real variable (the fractional part of x) $x^{\mathbf{Frac}}$ with $x = x^{\mathbf{Int}} + x^{\mathbf{Frac}}$ and $0 \leq x^{\mathbf{Frac}} < 1$. A mixed formula $R(x_1, \cdots, x_m, y_1, \cdots, y_n)$, where $x_1, \cdots, x_m$, $y_1, \cdots, y_n$ are the free variables, can therefore be translated into another mixed formula $\hat{R}$ (called R's *separation*):

$$R(x_1^{\mathbf{Int}} + x_1^{\mathbf{Frac}}, \cdots, x_m^{\mathbf{Int}} + x_m^{\mathbf{Frac}}, y_1, \cdots, y_n) \wedge 0 \leq x_1^{\mathbf{Frac}} < 1 \wedge \cdots \wedge 0 \leq x_m^{\mathbf{Frac}} < 1.$$

Notice that the separation $\hat{R}$ contains real variables $x_1^{\mathbf{Frac}}, \cdots, x_m^{\mathbf{Frac}}$ and integer variables $x_1^{\mathbf{Int}}, \cdots, x_m^{\mathbf{Int}}$, $y_1, \cdots, y_n$. The following result can be easily obtained from [24], in which the separation can be written into a Boolean combination of dense linear constraints, discrete linear constraints, and discrete mod constraints. A nice property of the Boolean combination is that real variables and integer variables are separated: each constraint in the combination either contains real variables $x_1^{\mathbf{Frac}}, \cdots, x_m^{\mathbf{Frac}}$ only or contains integer variables $x_1^{\mathbf{Int}}, \cdots, x_m^{\mathbf{Int}}$, $y_1, \cdots, y_n$ only.

Theorem 1 *The separation of any mixed formula can be written into a Boolean combination of dense linear constraints, discrete linear constraints, and discrete mod constraints.*

Definition 1 *R is a* mixed linear relation *if it is a mixed formula $R(\mathbf{X}, \mathbf{Y}, \mathbf{X}', \mathbf{Y}')$ over $2m$ real variables $\mathbf{X} = x_1, \cdots, x_m$* and *$\mathbf{X}' = x'_1, \cdots, x'_m$ and $2n$ integer variables $\mathbf{Y} = y_1, \cdots, y_n$* and *$\mathbf{Y}' = y'_1, \cdots, y'_n$.*

We use $\mathbf{U}$ to denote an m-ary real vector and use $\mathbf{V}$ to denote an n-ary integer vector.

Definition 2 *A mixed linear relation R is* transitive *if for all $\mathbf{U}, \mathbf{V}, \mathbf{U}', \mathbf{V}', \mathbf{U}'', \mathbf{V}''$, $R(\mathbf{U}, \mathbf{V}, \mathbf{U}', \mathbf{V}') \wedge R(\mathbf{U}', \mathbf{V}', \mathbf{U}'', \mathbf{V}'')$ implies $R(\mathbf{U}, \mathbf{V}, \mathbf{U}'', \mathbf{V}'')$. An infinite sequence $(\mathbf{U}^0, \mathbf{V}^0), \cdots, (\mathbf{U}^k, \mathbf{V}^k), \cdots$ is* an ω-chain *of R if $R(\mathbf{U}^k, \mathbf{V}^k, \mathbf{U}^{k+1}, \mathbf{V}^{k+1})$ holds for all $k \geq 0$. The sequence is a* strong ω-chain *of R if it is an ω-chain of R satisfying $R(\mathbf{U}^{k_1}, \mathbf{V}^{k_1}, \mathbf{U}^{k_2}, \mathbf{V}^{k_2})$ for all $0 \leq k_1 < k_2$.*

Notice that, if R is transitive, then any subsequence

$$(\mathbf{U}^{i_0}, \mathbf{V}^{i_0}), \cdots, (\mathbf{U}^{i_k}, \mathbf{V}^{i_k}), \cdots$$

(with $0 \leq i_0 < \cdots < i_k < \cdots$) of an ω-chain $(\mathbf{U}^0, \mathbf{V}^0), \cdots, (\mathbf{U}^k, \mathbf{V}^k), \cdots$ is also an ω-chain of R. According to the definition of the separation $\hat{R}$ (which is also a mixed linear relation) of a mixed linear relation R and Theorem 1, the following lemma can be proved.

Lemma 1 *(1). A mixed linear relation is transitive iff its separation is transitive. (2). A mixed linear relation has an ω-chain iff its separation has an ω-chain.*

3. A Technical Lemma

We will show that it is decidable whether a transitive mixed linear relation R has an ω-chain. From Lemma 1, it suffices to work on the separation of R; i.e., from Theorem 1, we assume that R itself is already in the form of a Boolean combination of dense linear constraints (with each real variable taking values in $[0, 1)$), discrete linear constraints, and discrete mod constraints. That is, $R(\mathbf{X}, \mathbf{Y}, \mathbf{X}', \mathbf{Y}')$ can be written as a disjunction

$$R_1 \vee \cdots \vee R_p \tag{1}$$

for some p, where each R_i is a conjunction of

$$S_i \wedge T_i.$$

Each S_i is a conjunction of l dense linear equations

$$\bigwedge_{1 \leq j \leq l} P_{ij}^1(\mathbf{X}) + Q_{ij}^1(\mathbf{X}') = c_{ij}^1, \tag{2}$$

followed by l dense linear inequalities

$$\bigwedge_{1\le j\le l} P^2_{ij}(\mathbf{X}) + Q^2_{ij}(\mathbf{X}') > c^2_{ij}, \tag{3}$$

with $\mathbf{X}$ and $\mathbf{X}'$ taking values in $[0,1)^m$. Each T_i is a conjunction of l discrete linear inequalities

$$\bigwedge_{1\le j\le l} P^3_{ij}(\mathbf{Y}) + Q^3_{ij}(\mathbf{Y}') > c^3_{ij}, \tag{4}$$

followed by l discrete mod constraints

$$\bigwedge_{1\le j\le l} P^4_{ij}(\mathbf{Y}) + Q^4_{ij}(\mathbf{Y}') \equiv_{d_{ij}} c^4_{ij}. \tag{5}$$

Notice that discrete linear equations like $y_1 + 2y_2 = 3$ can be expressed in discrete linear inequalities such as $y_1 + 2y_2 > 2 \wedge -y_1 - 2y_2 > -4$. Also notice that the negation of a discrete mod constraint like $y_1 + 2y_2 \not\equiv_5 3$ can be expressed into a finite disjunction of mod constraints in (5). Each P^h_{ij} and each Q^h_{ij} for $h = 1,2$ (resp. $h = 3,4$) are linear combinations (with integer coefficients) over real variables (resp. integer variables).

Mod constraints in (5) can be eliminated using the following procedure. Take

$$d = \prod_{1\le i\le p, 1\le j\le l} d_{ij}.$$

Let $\mathbf{d}$ be an n-ary integer vector taking values in $\{0,\cdots,d-1\}^n$. Let $R'(\mathbf{X},\mathbf{Z},\mathbf{X}',\mathbf{Z}')$ be

$$\bigvee_{\mathbf{d},\mathbf{d}'} R(\mathbf{X}, d\mathbf{Z}+\mathbf{d}, \mathbf{X}', d\mathbf{Z}'+\mathbf{d}')$$

by substituting $\mathbf{Y}$ with $d\mathbf{Z}+\mathbf{d}$ and $\mathbf{Y}'$ with $d\mathbf{Z}'+\mathbf{d}'$ in $R(\mathbf{X},\mathbf{Y},\mathbf{X}',\mathbf{Y}')$, for all possible choices of $\mathbf{d}$ and $\mathbf{d}'$. Clearly,

- R is transitive iff R' is transitive, and
- R has an ω-chain iff R' has an ω-chain.

In R', there are no mod-constraints, since, after the substitution, the truth value of each mod-constraint in (5) is known (according to the choice of $\mathbf{d}$ and $\mathbf{d}'$). Hence, we may assume that R itself does not contain mod-constraints in (5).

Consider an infinite sequence $\mathcal{C}^\omega$

$$(\mathbf{U}^0,\mathbf{V}^0),\cdots,(\mathbf{U}^k,\mathbf{V}^k),\cdots.$$

Let $f(\mathbf{X},\mathbf{Y})$ be a term that is a linear combination of real variables $\mathbf{X}$ and integer variables $\mathbf{Y}$. The term is *increasing* (resp. *decreasing, flat*) on $\mathcal{C}^\omega$ if $f(\mathbf{U}^k,\mathbf{V}^k) < f(\mathbf{U}^{k+1},\mathbf{V}^{k+1})$ (resp. $f(\mathbf{U}^k,\mathbf{V}^k) > f(\mathbf{U}^{k+1},\mathbf{V}^{k+1})$, $f(\mathbf{U}^k,\mathbf{V}^k) = f(\mathbf{U}^{k+1},\mathbf{V}^{k+1})$), for each $k \ge 0$. The term is *bounded increasing* (resp. *bounded decreasing*) on $\mathcal{C}^\omega$ if f is increasing (resp. decreasing) on $\mathcal{C}^\omega$ and there is a number b such that

$f(\mathbf{U}^k, \mathbf{V}^k) < b$ (resp. $f(\mathbf{U}^k, \mathbf{V}^k) > b$) for all $k \geq 0$. The term is *unbounded increasing* (resp. *unbounded decreasing*) on $\mathcal{C}^\omega$ if f is increasing (resp. decreasing) on $\mathcal{C}^\omega$ and f is not bounded increasing (resp. decreasing) on $\mathcal{C}^\omega$. The term of f could (but need not) be in one of the following five *modes* on $\mathcal{C}^\omega$:

(mode1) unbounded increasing,

(mode2) unbounded decreasing,

(mode3) flat,

(mode4) bounded increasing,

(mode5) bounded decreasing.

Clearly, when f only contains real variables, (mode1) and (mode2) are impossible (since each real variables is assumed in $[0, 1)$); when f only contains integer variables, (mode4) and (mode5) are impossible.

We observe that, since R is transitive, R has an ω-chain iff R has an ω-chain $\mathcal{C}^\omega$ on which each real variable $x \in \mathbf{X}$ (as well as each integer variable $y \in \mathbf{Y}$, and each term P^h_{ij} and Q^h_{ij}, $h = 1, 2, 3$, $1 \leq i \leq p, 1 \leq j \leq l$) is in one of the five modes on $\mathcal{C}^\omega$. A *mode vector* $\mathcal{M}$ is used to indicate the chosen mode for each of the variables and the terms. There are at most $3^m 3^n 3^{3pl} 3^{3pl}$ distinct mode vectors. Therefore, in order to decide whether R has an ω-chain, we only need to decide whether R has an ω-chain with some mode vector $\mathcal{M}$. In the sequel, we use the following abbreviation.

Definition 3 *An ω-chain is* monotonic of mode $\mathcal{M}$ *(or simply,* monotonic *when $\mathcal{M}$ is understood) if the chain is with mode vector $\mathcal{M}$.*

Now, we are ready to prove the following lemma using the pigeon-hole principle.

Lemma 2 *Suppose that R is a transitive mixed linear relation in the form of $R = R_1 \vee \cdots \vee R_p$ where each R_i is a conjunction of atomic formulas in (2,3,4,5). Then, R has an ω-chain iff R_i has a monotonic and strong ω-chain for some $1 \leq i \leq p$ and some mode vector $\mathcal{M}$.*

Proof. ($\Rightarrow$). Assume that R has an ω-chain $\mathcal{C}^\omega$

$$(\mathbf{U}^0, \mathbf{V}^0), \cdots, (\mathbf{U}^k, \mathbf{V}^k), \cdots \tag{6}$$

that is monotonic for some mode vector $\mathcal{M}$. $R(\mathbf{U}^{k_1}, \mathbf{V}^{k_1}, \mathbf{U}^{k_2}, \mathbf{V}^{k_2})$ holds for any $0 \leq k_1 < k_2$, since R is transitive. Recall that $R = R_1 \vee \cdots \vee R_p$. Notice that each R_i is not necessarily transitive. The following technique generalizes the one presented in [11]. We use a predicate $I(k_1, k_2, i)$ to indicate $0 \leq k_1 < k_2 \wedge R_i(\mathbf{U}^{k_1}, \mathbf{V}^{k_1}, \mathbf{U}^{k_2}, \mathbf{V}^{k_2})$. Clearly, for any k_1, k_2 with $0 \leq k_1 < k_2$, there is an i ($1 \leq i \leq p$) such that $I(k_1, k_2, i)$ holds. Define $I'(k_1, i)$ as $\forall k \exists k_2 (k_2 > k \wedge I(k_1, k_2, i))$. Hence, $I'(k_1, i)$ is true iff there are infinitely many k_2 satisfying $I(k_1, k_2, i)$. Since i is bounded (i.e., $1 \leq i \leq p$), for each k_1, there is an i satisfying $I'(k_1, i)$. Therefore, there is an i_0 ($1 \leq i_0 \leq p$), such that

$$\forall k \exists k_1 (k_1 > k \wedge I'(k_1, i_0)). \tag{7}$$

That is, there are infinitely many k_1 satisfying $I'(k_1, i_0)$. According to the definition of I' and I, formula (7) can be translated back to the following formula:

$$\forall k \exists k_1 > k \forall k' > k_1 \exists k_2 > k' R_{i_0}(\mathbf{U}^{k_1}, \mathbf{V}^{k_1}, \mathbf{U}^{k_2}, \mathbf{V}^{k_2}). \tag{8}$$

Since $\mathcal{C}^\omega$ is monotonic, there is a $\mathbf{U} \in [0,1]^m$ such that $\lim \mathbf{U}^k = \mathbf{U}$. In addition, $Q^1_{i_0 j}(\mathbf{U}^k)$, $Q^2_{i_0 j}(\mathbf{U}^k)$, and $Q^3_{i_0 j}(\mathbf{V}^k)$ in R_{i_0} (R_{i_0} is given in the form of (2),(3), and (4)) are all monotonic wrt k. Hence, formula (8) can be strengthened into

$$\forall k \exists k_1 > k \exists k' > k_1 \forall k_2 > k' R_{i_0}(\mathbf{U}^{k_1}, \mathbf{V}^{k_1}, \mathbf{U}^{k_2}, \mathbf{V}^{k_2}). \tag{9}$$

That is, there are infinitely many k_1 such that, for each of these k_1, there is a $k' > k_1$ satisfying $R_{i_0}(\mathbf{U}^{k_1}, \mathbf{V}^{k_1}, \mathbf{U}^{k_2}, \mathbf{V}^{k_2})$ for all $k_2 > k'$. From these infinitely many k_1's, we select any strictly increasing infinite sequence

$$k_1^0, \cdots, k_1^q, \cdots.$$

For each k_1^q, we can pick a k_2^q from (9) (treating k_1^q as k_1 and k_2^q as k_2). By making each k_2^q large enough, we can obtain a strictly increasing infinite sequence

$$k_2^0, \cdots, k_2^q, \cdots.$$

Notice that, from (9), for each q,

$$\forall k \geq k_2^q R_{i_0}(\mathbf{U}^{k_1^q}, \mathbf{V}^{k_1^q}, \mathbf{U}^k, \mathbf{V}^k). \tag{10}$$

Now, we define a sequence of indices as follows. Let $t_0 = 0$. Pick t_1 as any number satisfying $t_0 < t_1$ and $k_2^{t_0} < k_1^{t_1}$. Pick t_2 as any number satisfying $t_1 < t_2$ and $k_2^{t_1} < k_1^{t_2}$, and so on. The existence of each t_q is guaranteed by the monotonicity of the two sequences $k_1^0, \cdots, k_1^q, \cdots$ and $k_2^0, \cdots, k_2^q, \cdots$. It is easy to verify

$$R_{i_0}(\mathbf{U}^{k_1^{t_q}}, \mathbf{V}^{k_1^{t_q}}, \mathbf{U}^{k_1^{t_{q+1}}}, \mathbf{V}^{k_1^{t_{q+1}}})$$

holds for each $q \geq 0$ according to the choice of each t_q and (10). Hence,

$$(\mathbf{U}^{k_1^{t_0}}, \mathbf{V}^{k_1^{t_0}}), \cdots, (\mathbf{U}^{k_1^{t_q}}, \mathbf{V}^{k_1^{t_q}}), \cdots$$

is an ω-chain of R_{i_0}, which is also monotonic of mode $\mathcal{M}$. Notice that the ω-chain is also a strong ω-chain of R_{i_0}. This is because of the definition of t_q and (10). Therefore, we have already shown that, if R has an ω-chain, then R_{i_0} has a monotonic and strong ω-chain for some i_0 and $\mathcal{M}$.

($\Leftarrow$). Obvious. □

Recall that $R_i = S_i \wedge T_i$ where S_i contains only dense variables and T_i contains only integer variables. Therefore, for any $\mathcal{M}$, R_i has a monotonic and strong ω-chain iff both S_i and T_i have a monotonic and strong ω-chain. Hence, from now on, we will focus on S_i and T_i separately by looking at the following two problems:

1. whether S has a monotonic and strong ω-chain, where S is a conjunction of dense linear equations in (2) and inequalities in (3);

2. whether T has a monotonic and strong ω-chain, where T is a conjunction of integer linear inequalities in (4).

Notice that S and T are not necessarily transitive. Solutions to the problems are given in the following two sections.

4. The Existence of ω-chains for Dense Linear Equations and Inequalities

Assume that S is a conjunction of l dense linear equations $P_j^1(\mathbf{X})+Q_j^1(\mathbf{X}')=c_j^1$ and l dense linear inequalities $P_j^2(\mathbf{X})+Q_j^2(\mathbf{X}')>c_j^2$. Each dense variable takes values in $[0,1)$. Let $\mathcal{M}$ be a mode vector (on each dense variable, each term P_j^1, Q_j^1, P_j^2, Q_j^2, $1\le j\le l$). We use "$\nearrow$", "$\rightarrow$" and "$\searrow$" to stand for "bounded increasing", "flat" and "bounded decreasing", respectively (the other two modes "unbounded increasing" and "unbounded decreasing" are not possible for dense variables and dense terms). Assume that

$$\mathbf{U}^0,\cdots,\mathbf{U}^k,\cdots$$

is a monotonic and strong ω-chain $\mathbf{U}^\omega$ of S, for a given $\mathcal{M}$. Therefore, $S(\mathbf{U}^{k_1},\mathbf{U}^{k_2})$ holds for any $0\le k_1<k_2$ (notice that S itself is not necessarily transitive.). Since dense variables take values in $[0,1)$, we have $\lim \mathbf{U}^k=\mathbf{U}$ for some $\mathbf{U}\in[0,1]^m$.

A number of observations can be made on $\mathbf{U}^\omega$ and $\mathcal{M}$. For instance, each variable $x\in\mathbf{X}$ (as well as each term P_j^1, Q_j^1, P_j^2, Q_j^2) has a mode (given in $\mathcal{M}$) on $\mathbf{U}^\omega$. In particular, for a linear equation like $P_j^1(\mathbf{X})+Q_j^1(\mathbf{X}')=c_j^1$, the mode of P_j^1 and the mode of Q_j^1 must be flat. How about a linear inequality like $P_j^2(\mathbf{X})+Q_j^2(\mathbf{X}')>c_j^2$? Let us consider the case when $\mathcal{M}(P_j^2)=\searrow$ and $\mathcal{M}(Q_j^2)=\nearrow$. In this case, since $\lim\mathbf{U}^k=\mathbf{U}$, we can easily conclude that, for any $k_1<k_2$, $P_j^2(\mathbf{U}^{k_1})>P_j^2(\mathbf{U}^{k_2})>P_j^2(\mathbf{U})$, $Q_j^2(\mathbf{U}^{k_1})<Q_j^2(\mathbf{U}^{k_2})<Q_j^2(\mathbf{U})$, $P_j^2(\mathbf{U})+Q_j^2(\mathbf{U})\ge c_j^2$. Similar conclusions can be made for all the other possible choices for $\mathcal{M}(P_j^2)$ and $\mathcal{M}(Q_j^2)$. Combining all these observations, we obtain that, for any $k_1<k_2$, $H(\mathbf{U},\mathbf{U}^{k_1},\mathbf{U}^{k_2},\mathcal{M})$ holds, where H is defined as follows:

- $\mathbf{U}^{k_1}$ and $\mathbf{U}^{k_2}$ are consistent to the mode $\mathcal{M}(x)$ for each $x\in\mathbf{X}$. That is, for all $x\in\mathbf{X}$, $\mathbf{U}^{k_1}(x)<\mathbf{U}^{k_2}(x)$ (resp. $=$, $>$) and $\mathbf{U}^{k_2}(x)\le\mathbf{U}(x)$ (resp. $=$, $\ge$) if $\mathcal{M}(x)=\nearrow$ (resp. $\rightarrow$, $\searrow$), where $\mathbf{U}^{k_1}(x)$ is the component for variable x in vector $\mathbf{U}^{k_1}$.

- For each linear equation $P_j^1(\mathbf{X})+Q_j^1(\mathbf{X}')=c_j^1$, both $\mathcal{M}(P_j^1)$ and $\mathcal{M}(Q_j^1)$ must be flat. In this case, $P_j^1(\mathbf{U})+Q_j^1(\mathbf{U})=c_j^1$, $P_j^1(\mathbf{U}^{k_1})=P_j^1(\mathbf{U}^{k_2})=P_j^1(\mathbf{U})$, $Q_j^1(\mathbf{U}^{k_1})=Q_j^1(\mathbf{U}^{k_2})=Q_j^1(\mathbf{U})$.

- For each linear inequality $P_j^2(\mathbf{X})+Q_j^2(\mathbf{X}')>c_j^2$, according to each possible combination of $\mathcal{M}(P_j^2)$ and $\mathcal{M}(Q_j^2)$, one of the following nine cases is satisfied:

  - $\mathcal{M}(P_j^2)=\nearrow$ and $\mathcal{M}(Q_j^2)=\nearrow$. $P_j^2(\mathbf{U}^{k_1})<P_j^2(\mathbf{U}^{k_2})<P_j^2(\mathbf{U})$, $Q_j^2(\mathbf{U}^{k_1})<Q_j^2(\mathbf{U}^{k_2})<Q_j^2(\mathbf{U})$, and $P_j^2(\mathbf{U})+Q_j^2(\mathbf{U})>c_j^2$,
  - $\mathcal{M}(P_j^2)=\nearrow$ and $\mathcal{M}(Q_j^2)=\rightarrow$. $P_j^2(\mathbf{U}^{k_1})<P_j^2(\mathbf{U}^{k_2})<P_j^2(\mathbf{U})$, $Q_j^2(\mathbf{U}^{k_1})=Q_j^2(\mathbf{U}^{k_2})=Q_j^2(\mathbf{U})$, $P_j^2(\mathbf{U})+Q_j^2(\mathbf{U})>c_j^2$,
  - $\mathcal{M}(P_j^2)=\nearrow$ and $\mathcal{M}(Q_j^2)=\searrow$. $P_j^2(\mathbf{U}^{k_1})<P_j^2(\mathbf{U}^{k_2})<P_j^2(\mathbf{U})$, $Q_j^2(\mathbf{U}^{k_1})>Q_j^2(\mathbf{U}^{k_2})>Q_j^2(\mathbf{U})$, $P_j^2(\mathbf{U})+Q_j^2(\mathbf{U})>c_j^2$,
  - $\mathcal{M}(P_j^2)=\rightarrow$ and $\mathcal{M}(Q_j^2)=\nearrow$. $P_j^2(\mathbf{U}^{k_1})=P_j^2(\mathbf{U}^{k_2})=P_j^2(\mathbf{U})$, $Q_j^2(\mathbf{U}^{k_1})<Q_j^2(\mathbf{U}^{k_2})<Q_j^2(\mathbf{U})$, $P_j^2(\mathbf{U})+Q_j^2(\mathbf{U})>c_j^2$,

- $\mathcal{M}(P_j^2) = \rightarrow$ and $\mathcal{M}(Q_j^2) = \rightarrow$. $P_j^2(\mathbf{U}^{k_1}) = P_j^2(\mathbf{U}^{k_2}) = P_j^2(\mathbf{U})$, $Q_j^2(\mathbf{U}^{k_1}) = Q_j^2(\mathbf{U}^{k_2}) = Q_j^2(\mathbf{U})$, $P_j^2(\mathbf{U}) + Q_j^2(\mathbf{U}) > c_j^2$,
- $\mathcal{M}(P_j^2) = \rightarrow$ and $\mathcal{M}(Q_j^2) = \searrow$. $P_j^2(\mathbf{U}^{k_1}) = P_j^2(\mathbf{U}^{k_2}) = P_j^2(\mathbf{U})$, $Q_j^2(\mathbf{U}^{k_1}) > Q_j^2(\mathbf{U}^{k_2}) > Q_j^2(\mathbf{U})$, $P_j^2(\mathbf{U}) + Q_j^2(\mathbf{U}) \geq c_j^2$,
- $\mathcal{M}(P_j^2) = \searrow$ and $\mathcal{M}(Q_j^2) = \nearrow$. $P_j^2(\mathbf{U}^{k_1}) > P_j^2(\mathbf{U}^{k_2}) > P_j^2(\mathbf{U})$, $Q_j^2(\mathbf{U}^{k_1}) < Q_j^2(\mathbf{U}^{k_2}) < Q_j^2(\mathbf{U})$, $P_j^2(\mathbf{U}) + Q_j^2(\mathbf{U}) \geq c_j^2$,
- $\mathcal{M}(P_j^2) = \searrow$ and $\mathcal{M}(Q_j^2) = \rightarrow$. $P_j^2(\mathbf{U}^{k_1}) > P_j^2(\mathbf{U}^{k_2}) > P_j^2(\mathbf{U})$, $Q_j^2(\mathbf{U}^{k_1}) = Q_j^2(\mathbf{U}^{k_2}) = Q_j^2(\mathbf{U})$, $P_j^2(\mathbf{U}) + Q_j^2(\mathbf{U}) \geq c_j^2$,
- $\mathcal{M}(P_j^2) = \searrow$ and $\mathcal{M}(Q_j^2) = \searrow$. $P_j^2(\mathbf{U}^{k_1}) > P_j^2(\mathbf{U}^{k_2}) > P_j^2(\mathbf{U})$, $Q_j^2(\mathbf{U}^{k_1}) > Q_j^2(\mathbf{U}^{k_2}) > Q_j^2(\mathbf{U})$, $P_j^2(\mathbf{U}) + Q_j^2(\mathbf{U}) \geq c_j^2$.

Since $\lim \mathbf{U}^k = \mathbf{U}$, we have

$$\forall \delta > 0 \exists \mathbf{U}' \in [0,1)^m \forall \delta' > 0 \exists \mathbf{U}'' \in [0,1)^m$$

$$(H(\mathbf{U}, \mathbf{U}', \mathbf{U}'', \mathcal{M}) \wedge |\mathbf{U}' - \mathbf{U}| < \delta \wedge |\mathbf{U}'' - \mathbf{U}| < \delta') \tag{11}$$

Conversely, we can show the following lemma.

Lemma 3 *If there are a $\mathbf{U} \in [0,1]^m$ and a mode vector $\mathcal{M}$ satisfying formula (11), then S has a monotonic (of mode $\mathcal{M}$) and strong ω-chain in $[0,1)^m$.*

Proof. Assume (11) holds for some $\mathbf{U} \in [0,1]^m$ and a mode vector $\mathcal{M}$. That is, we can pick a sequence in $[0,1)^m$

$$\mathbf{W}^0, \cdots, \mathbf{W}^k, \cdots$$

such that,

- $\lim \mathbf{W}^k = \mathbf{U}$,
- $H(\mathbf{U}, \mathbf{W}^0, \mathbf{W}^k, \mathcal{M})$ for each $k \geq 1$.

According to the fact that $\lim \mathbf{W}^k = \mathbf{U}$ and the first item in the definition of H, we can always pick a subsequence of $\mathbf{W}^0, \cdots, \mathbf{W}^k, \cdots$ such that each $x \in \mathbf{X}$ has mode $\mathcal{M}(x)$ on the subsequence. Without loss of generality, we assume that $\mathbf{W}^0, \cdots, \mathbf{W}^k, \cdots$ itself is the subsequence.

From the definition of H, for each linear equation $P_j^1(\mathbf{X}) + Q_j^1(\mathbf{X}') = c_j^1$, $\mathcal{M}(P_j^1)$ and $\mathcal{M}(Q_j^1)$ must both be flat. In addition, $P_j^1(\mathbf{U}) + Q_j^1(\mathbf{U}) = c_j^1$, $P_j^1(\mathbf{W}^0) = P_j^1(\mathbf{W}^k) = P_j^1(\mathbf{U})$, $Q_j^1(\mathbf{W}^0) = Q_j^1(\mathbf{W}^k) = Q_j^1(\mathbf{U})$. Therefore, $\mathbf{W}^0, \cdots, \mathbf{W}^k, \cdots$ (as well as any subsequence) is already a strong ω-chain for the conjunction of these linear equations. Clearly, each P_j^1 and each Q_j^1 are in mode $\mathcal{M}(P_j^1) = \mathcal{M}(Q_j^1) = \rightarrow$ on the chain. In the rest of the proof, a "subsequence" always starts from $\mathbf{W}^0$.

For each linear inequality $P_j^2(\mathbf{X}) + Q_j^2(\mathbf{X}') > c_j^2$, we will show that a subsequence of $\mathbf{W}^0, \cdots, \mathbf{W}^k, \cdots$ can be picked such that the subsequence is a strong ω-chain of the linear inequality, and any subsequence of the subsequence is also a strong ω-chain of the linear inequality. In addition, P_j^2 and Q_j^2 are in modes $\mathcal{M}(P_j^2)$ and $\mathcal{M}(Q_j^2)$ on the subsequence, respectively. By working on each linear inequality one by one, a subsequence can be eventually picked which is a monotonic (of mode $\mathcal{M}$) and strong ω-chain of S. Once this is done, the lemma follows.

There are nine cases for the mode choices of $\mathcal{M}(P_j^2)$ and $\mathcal{M}(Q_j^2)$. We only prove the case when $\mathcal{M}(P_j^2) = \searrow$ and $\mathcal{M}(Q_j^2) = \nearrow$; all the other cases can be shown analogously. In the case, according to the definition of H, for each $k \geq 1$, $P_j^2(\mathbf{W}^0) > P_j^2(\mathbf{W}^k) > P_j^2(\mathbf{U})$, $Q_j^2(\mathbf{W}^0) < Q_j^2(\mathbf{W}^k) < Q_j^2(\mathbf{U})$, $P_j^2(\mathbf{U}) + Q_j^2(\mathbf{U}) \geq c_j^2$. Since $\lim Q_j^2(\mathbf{W}^k) = Q_j^2(\mathbf{U})$ and $\lim P_j^2(\mathbf{W}^k) = P_j^2(\mathbf{U})$, if we take $k^0 = 0$, then we can pick a large enough k^1 such that

- $P_j^2(\mathbf{W}^{k^0}) > P_j^2(\mathbf{W}^{k^1})$, and
- $Q_j^2(\mathbf{W}^{k^0}) < Q_j^2(\mathbf{W}^{k^1})$, and
- $P_j^2(\mathbf{W}^{k^0}) + Q_j^2(\mathbf{W}^{k^1}) > c_j^2$ (i.e., $(\mathbf{W}^{k^0}, \mathbf{W}^{k^1})$ satisfies the inequality).

Similarly, we can pick a large enough $k^2 > k^1$ such that

- $P_j^2(\mathbf{W}^{k^1}) > P_j^2(\mathbf{W}^{k^2})$, and
- $Q_j^2(\mathbf{W}^{k^1}) < Q_j^2(\mathbf{W}^{k^2})$, and
- $P_j^2(\mathbf{W}^{k^1}) + Q_j^2(\mathbf{W}^{k^2}) > c_j^2$ (i.e., $(\mathbf{W}^{k^1}, \mathbf{W}^{k^2})$ satisfies the inequality).

It can be checked that $(\mathbf{W}^{k^0}, \mathbf{W}^{k^2})$ also satisfies the inequality. This process can go on and, as a result, we obtain an infinite sequence

$$\mathbf{W}^{k^0}, \cdots, \mathbf{W}^{k^i}, \cdots$$

which satisfies:

- P_j^2 is in mode $\mathcal{M}(P_j^2) = \searrow$ on the sequence,
- Q_j^2 is in mode $\mathcal{M}(Q_j^2) = \nearrow$ on the sequence,
- $(\mathbf{W}^{k^{i_1}}, \mathbf{W}^{k^{i_2}})$ satisfies the linear inequality for all i_1 and i_2.

Therefore, the sequence (as well as any subsequence) is a strong ω-chain of the linear inequality. □

Thus, S has a monotonic (of mode $\mathcal{M}$) and strong ω-chain iff formula (11), which is definable in the additive theory of reals, is satisfied by some $\mathbf{U} \in [0, 1]^m$. Hence,

Lemma 4 *Let S be a conjunction of l dense linear equations $P_j^1(\mathbf{X}) + Q_j^1(\mathbf{X}') = c_j^1$ and l dense linear inequalities $P_j^2(\mathbf{X}) + Q_j^2(\mathbf{X}') > c_j^2$ defined in (2,3). Let $\mathcal{M}$ be a mode vector on $\mathbf{X}$, $P_j^1, Q_j^1, P_j^2, Q_j^2$, $1 \leq j \leq l$. Then, it is decidable whether S has a monotonic and strong ω-chain.*

5. The Existence of ω-chains for Discrete Linear Inequalities

Assume that T is a conjunction of l discrete linear inequalities $P_j(\mathbf{Y}) + Q_j(\mathbf{Y}') > c_j$. Let $\mathcal{M}$ be a mode vector (on each integer variable, each term P_j, Q_j, $1 \leq j \leq$

l). We use "$\nearrow$", "$\rightarrow$" and "$\searrow$" to stand for "unbounded increasing", "flat" and "unbounded decreasing" modes, respectively. Assume that

$$\mathbf{V}^0, \cdots, \mathbf{V}^k, \cdots$$

is a monotonic and strong ω-chain $\mathbf{V}^\omega$ of T. Therefore,

$$\text{for any } k_1 < k_2, T(\mathbf{V}^{k_1}, \mathbf{V}^{k_2}). \tag{12}$$

(12) implies that, for each $1 \le j \le l$, the mode $\mathcal{M}(P_j)$ and the mode $\mathcal{M}(Q_j)$ only have the following five combinations (all the others are not possible):

- $\mathcal{M}(P_j) =\nearrow$ and $\mathcal{M}(Q_j) =\nearrow$,
- $\mathcal{M}(P_j) =\rightarrow$ and $\mathcal{M}(Q_j) =\nearrow$,
- $\mathcal{M}(P_j) =\searrow$ and $\mathcal{M}(Q_j) =\nearrow$,
- $\mathcal{M}(P_j) =\nearrow$ and $\mathcal{M}(Q_j) =\rightarrow$,
- $\mathcal{M}(P_j) =\rightarrow$ and $\mathcal{M}(Q_j) =\rightarrow$.

If $\mathcal{M}(P_j) =\rightarrow$ (resp. $\mathcal{M}(Q_j) =\rightarrow$), we use p_j (resp. q_j) to stands for $P_j(\mathbf{V}^0)$ (resp. $Q_j(\mathbf{V}^0)$). Similarly, if $\mathcal{M}(y) =\rightarrow$, we use v_y to denote the component of y in $\mathbf{V}^0$. Suppose $1 \le j_1 \ne j_2 \le l$, $\mathcal{M}(P_{j_1}) =\searrow$ and $\mathcal{M}(Q_{j_1}) =\nearrow$, $\mathcal{M}(P_{j_2}) =\nearrow$ and $\mathcal{M}(Q_{j_2}) =\rightarrow$. That is, $\lim P_{j_1}(\mathbf{V}^k) = -\infty$, $\lim Q_{j_1}(\mathbf{V}^k) = +\infty$, $\lim P_{j_2}(\mathbf{V}^k) = +\infty$, and for all k, $Q_{j_2}(\mathbf{V}^k) = q_{j_2}$. From (12), for all $k \ge 0$, we can pick $\mathbf{V}^{k_1}$ and $\mathbf{V}^{k_2}$ such that $T(\mathbf{V}^{k_1}, \mathbf{V}^{k_2})$, and

- $-k > P_{j_1}(\mathbf{V}^{k_1}) > P_{j_1}(\mathbf{V}^{k_2})$, and
- $k < Q_{j_1}(\mathbf{V}^{k_1}) < Q_{j_1}(\mathbf{V}^{k_2})$,

and

- $k < P_{j_2}(\mathbf{V}^{k_1}) < P_{j_2}(\mathbf{V}^{k_2})$, and
- $Q_{j_2}(\mathbf{V}^{k_1}) = Q_{j_2}(\mathbf{V}^{k_2}) = q_{j_2}$.

Similar statement can be made for all the valid choices of $\mathcal{M}(P_j)$ and $\mathcal{M}(Q_j)$, $1 \le j \le l$, as well as for $\mathcal{M}(y)$, $y \in \mathbf{Y}$. That is, for all $k \ge 0$, there are $\mathbf{V}^{k_1}$ and $\mathbf{V}^{k_2}$ such that

- $T(\mathbf{V}^{k_1}, \mathbf{V}^{k_2})$,
- $\mathbf{V}^{k_1}$ and $\mathbf{V}^{k_2}$ are consistent with mode $\mathcal{M}(y)$ for each $y \in \mathbf{Y}$. That is, for all $y \in \mathbf{Y}$, $\mathbf{V}^{k_1}(y) < \mathbf{V}^{k_2}(y)$ (resp. $=$, $>$) and $k < \mathbf{V}^{k_1}(y)$ (resp. $v_y = \mathbf{V}^{k_1}(y)$, $-k > \mathbf{V}^{k_1}(y)$) if $\mathcal{M}(y) =\nearrow$ (resp. $\rightarrow$, $\searrow$), where $\mathbf{V}^{k_1}(y)$ is the component for y in vector $\mathbf{V}^{k_1}$.
- For each $1 \le j \le l$, one of the following items holds:

- $\mathcal{M}(P_j) = \nearrow$ and $\mathcal{M}(Q_j) = \nearrow$. In this case, $k < P_j(\mathbf{V}^{k_1}) < P_j(\mathbf{V}^{k_2})$ and $k < Q_j(\mathbf{V}^{k_1}) < Q_j(\mathbf{V}^{k_2})$.
- $\mathcal{M}(P_j) = \rightarrow$ and $\mathcal{M}(Q_j) = \nearrow$. In this case, $P_j(\mathbf{V}^{k_1}) = P_j(\mathbf{V}^{k_2}) = p_j$ and $k < Q_j(\mathbf{V}^{k_1}) < Q_j(\mathbf{V}^{k_2})$.
- $\mathcal{M}(P_j) = \searrow$ and $\mathcal{M}(Q_j) = \nearrow$. In this case, $-k > P_j(\mathbf{V}^{k_1}) > P_j(\mathbf{V}^{k_2})$ and $k < Q_j(\mathbf{V}^{k_1}) < Q_j(\mathbf{V}^{k_2})$.
- $\mathcal{M}(P_j) = \nearrow$ and $\mathcal{M}(Q_j) = \rightarrow$. In this case, $k < P_j(\mathbf{V}^{k_1}) < P_j(\mathbf{V}^{k_2})$ and $Q_j(\mathbf{V}^{k_1}) = Q_j(\mathbf{V}^{k_2}) = q_j$.
- $\mathcal{M}(P_j) = \rightarrow$ and $\mathcal{M}(Q_j) = \rightarrow$. In this case, $P_j(\mathbf{V}^{k_1}) = P_j(\mathbf{V}^{k_2}) = p_j$ and $Q_j(\mathbf{V}^{k_1}) = Q_j(\mathbf{V}^{k_2}) = q_j$.

The above statement (replacing $\mathbf{V}^{k_1}$ with $\mathbf{V}$ and $\mathbf{V}^{k_2}$ with $\mathbf{V}'$) can be written as

$$\forall k \exists \mathbf{V} \exists \mathbf{V}' \; G(k, \mathbf{C}, \mathbf{V}, \mathbf{V}', \mathcal{M}) \tag{13}$$

where $\mathbf{C}$ represents the tuple of all the constant values p_j and q_j, $1 \le j \le l$, and v_y, $y \in \mathbf{Y}$. Clearly, G is a Presburger formula. Conversely, we can show the following lemma.

Lemma 5 *If there are a* $\mathbf{C}$ *and a mode vector* $\mathcal{M}$ *satisfying (13), then* T *has a monotonic and strong* ω*-chain.*

Proof. Assume (13) holds for some $\mathbf{C}$ and a mode vector $\mathcal{M}$. For $k = 0$, according to (13), we pick $\mathbf{V}_0, \mathbf{V}_0'$ satisfying $G(0, \mathbf{C}, \mathbf{V}_0, \mathbf{V}_0', \mathcal{M})$. Take

$$k = \max_{1 \le j \le l} \{|P_j(\mathbf{V}_0')|, |Q_j(\mathbf{V}_0')|\}.$$

For this k, according to (13), we pick any $\mathbf{V}_1, \mathbf{V}_1'$ satisfying $G(k, \mathbf{C}, \mathbf{V}_1, \mathbf{V}_1', \mathcal{M})$. What is the relationship among $\mathbf{V}_0, \mathbf{V}_0', \mathbf{V}_1, \mathbf{V}_1'$? Clearly, $T(\mathbf{V}_0, \mathbf{V}_0')$ and $T(\mathbf{V}_1, \mathbf{V}_1')$ hold. More importantly, $T(\mathbf{V}_0, \mathbf{V}_1)$ must be true. This can be concluded from the definition of G and the choice of k and $\mathbf{V}_1$. We can continue the procedure by taking

$$k = \max_{1 \le j \le l} \{|P_j(\mathbf{V}_1')|, |Q_j(\mathbf{V}_1')|\},$$

picking $\mathbf{V}_2, \mathbf{V}_2'$ from (13) according to this k, and concluding $T(\mathbf{V}_1, \mathbf{V}_2)$, etc. Finally, we obtain an ω-chain $\mathbf{V}_0, \cdots, \mathbf{V}_k, \cdots$ of T. It is straightforward to verify that the chain is monotonic (of mode $\mathcal{M}$) and strong. □

In summary, for any $\mathcal{M}$, T has a monotonic and strong ω-chain iff

$$\exists \mathbf{C} \forall k \exists \mathbf{V} \exists \mathbf{V}' \; G(k, \mathbf{C}, \mathbf{V}, \mathbf{V}', \mathcal{M}). \tag{14}$$

Since G is Presburger, we have,

Lemma 6 *Assume that* T *is a conjunction of* l *discrete linear inequalities* $P_j(\mathbf{Y}) + Q_j(\mathbf{Y}') > c_j$*. Let* $\mathcal{M}$ *be a mode vector on* $\mathbf{Y}$*,* P_j *and* Q_j*,* $1 \le j \le l$*. It is decidable whether* T *has a monotonic and strong* ω*-chain.*

Now, we are ready to put Theorem 1, Lemma 1, Lemma 2, Lemma 4, Lemma 6 together and conclude the main theorem.

Theorem 2 *It is decidable whether a transitive mixed linear relation has an ω-chain.*

An upper bound for the time complexity of the decidable result in Theorem 2 can be obtained as follows. Let R be given in (1) whose length is L. One can show that the length of formula (11) as well as formula (14) is $O(L)$ (for any fixed choice of $\mathcal{M}$). Using the complexity result given in [24], the satisfiability of (11) and the satisfiability of (14) are decidable in time $2^{L^{(m+n)^{O(1)}}}$, for each fixed $\mathcal{M}$. But since there are only (at most) $3^m 3^n 3^{3pl} 3^{3pl}$ choices for $\mathcal{M}$, whether R has an ω-chain is still decidable in time $2^{L^{(m+n)^{O(1)}}}$.

Notice that the transitivity in Theorem 2 is critical. The existence of an ω-chain is undecidable for mixed linear relations. The undecidability remains even for Presburger relations. This is because a Presburger relation can be used to encode one-step transitions of a deterministic two-counter machine. The negation of the halting problem (which is undecidable) for the machine can be reduced to the existence of an ω-chain for the Presburger relation.

6. Applications

In this section, we will study various verification problems for restricted infinite state systems containing both dense counters and discrete counters. We start with a general model.

6.1. Mixed linear counter systems

Let M be a machine that is equipped with a number of dense counters $\mathbf{X}$ and discrete counters $\mathbf{Y}$ and whose transitions involve changing control states while changing counter values. A configuration of M is a tuple consisting of a control state and counter values. Formally, M is a tuple $\langle S, \mathbf{X}, \mathbf{Y}, t\rangle$ where t is the one-step transition such that for each $s, s' \in S$, $t(s, \mathbf{X}, \mathbf{Y}, s', \mathbf{X}', \mathbf{Y}')$ indicates that M transits from a configuration $(s, \mathbf{X}, \mathbf{Y})$ at s to another configuration $(s', \mathbf{X}', \mathbf{Y}')$ at s'. $(s', \mathbf{U}', \mathbf{V}')$ is *reachable* from $(s, \mathbf{U}, \mathbf{V})$, written $\mathcal{T}(s, \mathbf{U}, \mathbf{V}, s', \mathbf{U}', \mathbf{V}')$, if there are k (for some k) configurations $(s_0, \mathbf{U}^0, \mathbf{V}^0), \cdots, (s_k, \mathbf{U}^k, \mathbf{V}^k)$ such that $(s_0, \mathbf{U}^0, \mathbf{V}^0) = (s, \mathbf{U}, \mathbf{V})$, $(s_k, \mathbf{U}^k, \mathbf{V}^k) = (s', \mathbf{U}', \mathbf{V}')$, and $t(s_i, \mathbf{U}^i, \mathbf{V}^i, s_{i+1}, \mathbf{U}^{i+1}, \mathbf{V}^{i+1})$ for all $0 \leq i < k$. In this case, we say that $(s, \mathbf{U}, \mathbf{V})$ reaches $(s', \mathbf{U}', \mathbf{V}')$ through configurations $(s_i, \mathbf{U}^i, \mathbf{V}^i)$, $0 \leq i \leq k$. Notice that $\mathcal{T}$, called the *binary reachability* of M, is the transitive closure of t. M is a *mixed linear counter system* if, when s and s' are understood as bounded integer variables,

- $t(s, \mathbf{X}, \mathbf{Y}, s', \mathbf{X}', \mathbf{Y}')$ is a mixed linear relation,
- $\mathcal{T}(s, \mathbf{X}, \mathbf{Y}, s', \mathbf{X}', \mathbf{Y}')$ is an (obviously transitive) mixed linear relation.

Now, we assume that M is a mixed linear counter system. Let I and P be two subsets of configurations of M both of which are definable by mixed formulas. There are two kinds of verification problems we will consider. M is *P-safe from I* if no configuration in I reaches a configuration in P. The *mixed linear safety* *problem* for M is to decide whether M is P-safe from I. An infinite sequence of configurations

$$(s_0, \mathbf{U}^0, \mathbf{V}^0), \cdots, (s_k, \mathbf{U}^k, \mathbf{V}^k), \cdots$$

of M is *P-live from I* if the following items hold:

- $(s_0, \mathbf{U}^0, \mathbf{V}^0) \in I$,
- there are infinitely many k such that $(s_k, \mathbf{U}^k, \mathbf{V}^k) \in P$, and
- for all $k \geq 0$, $t(s_k, \mathbf{U}^k, \mathbf{V}^k, s_{k+1}, \mathbf{U}^{k+1}, \mathbf{V}^{k+1})$. That is, the sequence is an infinite execution of M.

M is *P-live from I* if there is an infinite sequence of configurations that is P-live from I. The *mixed linear liveness problem* for M is to decide whether M is P-live from I.

These two problems can be further generalized. Let I, $P_1, \cdots, P_k$ be subsets of configurations of M definable in mixed formulas. The k-mixed linear safety problem for M is to decide whether no configuration in I reaches a configuration in P_k through some configurations $c_1, ..., c_{k-1}$ in $P_1, ..., P_{k-1}$ respectively. The k-mixed linear liveness problem for M is to decide whether there is an infinite execution of M that is P_i-live from I for each $1 \leq i \leq k$. The k-mixed linear safety (resp. liveness) problem is exactly the mixed linear safety (resp. liveness) problem, when $k = 1$.

Theorem 3 *(1). The k-mixed linear safety problem for mixed linear counter systems is decidable for each k. (2). The k-mixed linear liveness problem for mixed linear counter systems is decidable for each k.*

Proof. Let M be a mixed linear counter system with states S and one-step transition t, I and $P_1, \cdots, P_k$ be sets (definable by mixed formulas) of configurations of M. The proof of (1) is straightforward, since one can show that the set of configurations c_0 satisfying:

- c_0 in I,
- there are configurations $c_1 \in P_1, ..., c_k \in P_k$ such that c_0 reaches c_k through c_1,...,c_{k-1}; i.e., $\mathcal{T}(c_0, c_1)$,...,$\mathcal{T}(c_{k-1}, c_k)$,

is definable in a mixed formula (its satisfiability is decidable). Now, we look at (2). Define a formula $\hat{\mathcal{T}}$ as follows. $\hat{\mathcal{T}}(s, \mathbf{X}, \mathbf{Y}, s', \mathbf{X}', \mathbf{Y}')$ is true iff there are configurations $(s_1, \mathbf{X}^1, \mathbf{Y}^1), \cdots, (s_k, \mathbf{X}^k, \mathbf{Y}^k)$ such that,

- $(s, \mathbf{X}, \mathbf{Y})$ is reachable from some configuration in I,
- $(s_i, \mathbf{X}^i, \mathbf{Y}^i)$ satisfies P_i, for each $1 \leq i \leq k$,
- $(s, \mathbf{X}, \mathbf{Y})$ reaches $(s_1, \mathbf{X}^1, \mathbf{Y}^1)$ (i.e., $\mathcal{T}(s, \mathbf{X}, \mathbf{Y}, s_1, \mathbf{X}^1, \mathbf{Y}^1)$),
- $(s_i, \mathbf{X}^i, \mathbf{Y}^i)$ reaches $(s_{i+1}, \mathbf{X}^{i+1}, \mathbf{Y}^{i+1})$, for each $1 \leq i < k$,
- $(s_k, \mathbf{X}^k, \mathbf{Y}^k)$ reaches $(s', \mathbf{X}', \mathbf{Y}')$.

Since M is a mixed linear counter system, it is not hard to see that $\hat{\mathcal{T}}$ is a transitive mixed linear relation. (2) follows from Theorem 2, noticing that $\hat{\mathcal{T}}$ has an ω-chain iff there is an infinite execution of M that is P_i-live from I for each $1 \leq i \leq k$. □

Consider the *eventuality* problem: is there an infinite execution of M that starts from some configuration in I such that P is satisfied somewhere on the execution? The problem is a special case of the mixed linear liveness problem. To see this, let I' be the set of configurations that are reachable from I and satisfy P. Obviously, the eventuality problem is equivalent to the problem whether M is *true*-live from I', which is decidable (*true* stands for the set of all configurations) from Theorem 3. We can modify the eventuality problem as follows: is there an infinite execution of M that starts from some configuration in I such that P is satisfied by each configuration on the execution? Unfortunately, this modified problem is undecidable for M, even when M is a discrete timed automaton (cf. [12] for a proof).

In practice, there are many counter models that have been found being mixed linear. Applying Theorem 3 on these systems gives a number of new decidability results concerning safety/liveness verification. We first recall some definitions.

A *timed automaton* $\mathcal{A}$ is a tuple

$$\langle S, \{x_1, \cdots, x_m\}, \mathcal{C}, Inv, R, C\rangle,$$

where

- S is a finite set of *(control) states*,
- $x_1, \cdots, x_m$ are (dense) clocks,
- $\mathcal{C}$ is the set of all clock constraints over clocks $x_1, \cdots, x_m$; i.e., boolean combinations of formulas in the form of $x_i - x_j \sim d$ or $x_i \sim d$ where d is an integer, $\sim$ stands for $<, >, \leq, \geq, =$.
- $Inv : S \to \mathcal{C}$ assigns a clock constraint over clocks $x_1, \cdots, x_m$, called an *invariant*, to each state,
- $R : S \times S \to 2^{\{x_1, \cdots, x_m\}}$ assigns a subset of clocks to a directed edge in $S \times S$,
- $C : S \times S \to \mathcal{C}$ assigns a clock constraint over clocks $x_1, \cdots, x_m$, called a *reset condition*, to a directed edge in $S \times S$.

The semantics of $\mathcal{A}$ is defined as follows. A configuration $(s, \mathbf{U})$ is a pair of a control state s and a tuple $\mathbf{U}$ of clock values. A transition is either a progress transition or a reset transition. A progress transition makes all the clocks synchronously progress by a positive amount, during which the invariant is consistently satisfied, while the automaton remains at the same control state. A reset transition, by moving from state s_1 to state s_2, resets every clock in $R(s_1, s_2)$ to 0 and keeps all the other clocks unchanged. In addition, clock values before the transition satisfy the invariant $Inv(s_1)$ and the reset condition $C(s_1, s_2)$; clock values after the transition satisfy the invariant $Inv(s_2)$. In particular, when the clocks are integer-valued (and hence clocks are incremented by some positive integral amount in a progress transition), $\mathcal{A}$ is called a *discrete timed automaton.* The following characterization has recently been established [8].

Theorem 4 *Timed automata, as well as discrete timed automata, are mixed linear counter systems.*

Hence, from Theorem 3, the following corollary is obtained.

Corollary 1 *(1). The k-mixed linear safety problem is decidable for timed automata as well as for discrete timed automata [8].*

(2). The k-mixed linear liveness problem is decidable for discrete timed automata [12].

(3). The k-mixed linear liveness problem is decidable for timed automata.

A (free) counter is an integer variable that can be tested against 0, incremented by 1, decremented by 1, and stay unchanged. A timed automaton can be augmented with counters by integrating a reset transition with a counter operation. A counter in a timed automaton is *reversal-bounded* if there is a number r such that, during any execution of the automaton, the counter changes mode between nondecreasing and nonincreasing for at most r times. Let $\mathcal{A}$ be a timed automaton augmented with a finite number of reversal-bounded counters and one free counter. Now, a configuration $(s, \mathbf{U}, \mathbf{V})$ of $\mathcal{A}$ is a tuple of a control state s, dense clock values $\mathbf{U}$ and counter values $\mathbf{V}$. When $\mathcal{A}$ does not contain any clocks, it is a finite automaton augmented with reversal-bounded counters and one free counter.

Theorem 5 *(1). Discrete timed automata augmented with reversal-bounded counters and one free counter are mixed linear counter systems [10].*

(2). Timed automata augmented with reversal-bounded counters and one free counter are mixed linear counter systems [9].

Hence, from Theorem 3, the following corollary is obtained.

Corollary 2 *(1). The k-mixed linear safety problem is decidable for discrete timed automata augmented with reversal-bounded counters and one free counter [10].*

(2). The k-mixed linear safety problem is decidable for timed automata augmented with reversal-bounded counters and one free counter [9].

(3). The k-mixed linear liveness problem is decidable for finite automata augmented with reversal-bounded counters and one free counter [11].

(4). The k-mixed linear liveness problem is decidable for timed automata (as well as discrete timed automata) augmented with reversal-bounded counters and one free counter.

Corollary 1 (3) and Corollary 2 (4) are new decidability results. One shall notice that the loop analysis techniques presented in [12, 11] to show Corollary 1 (2) and Corollary 2 (3) can not be easily used to prove our new results. The corollaries can be used to automatically verify a class of non-region safety and liveness properties that, previously, could not be done using the traditional region technique [2]. Below, we look at an example of liveness verification. Consider a system S of two concurrent processes S_1 and S_2. The two processes may use a counting semaphore to perform concurrency control. In some applications, we would like to ensure that the concurrency control makes S starvation-free; i.e., it is not possible that the composite system S, starting from some initial configuration, executes for some finite number of steps and then S_1 solely executes forever (in this case, S_2 starves). We use S' to denote the system that behaves like S then, nondeterministically, behaves like S_1 afterwards. It is observed that S_2 starves iff S' has an ω-chain (i.e., S' is *true*-live from the initial configuration). Now, we suppose that S_1 and S_2 are real-time processes modeled as discrete timed automata. A free counter is used for the counting semaphore. From Corollary 2 (4), whether S_2 starves can be automatically verified.

Besides mixed linear safety/liveness problems, one may also be interested in a class of boundedness problems as below. Let M be a mixed linear counter system with dense counters $\mathbf{X}$ and discrete counters $\mathbf{Y}$. Let I be a set of configurations definable in a mixed formula. We use l to denote a linear combination of $\mathbf{X}$ and $\mathbf{Y}$; i.e., $l = \Sigma a_i x_i + \Sigma b_j y_j + c$ with a_i, b_j, c integers. Let $l_1, ..., l_p$ be p such linear combinations. Are there numbers $B_1, ..., B_p$ such that, starting from a configuration in I, M can only reach a configuration satisfying $l_i \leq B_i$ for each $1 \leq i \leq p$? This boundedness problem can be easily shown decidable, since the question is equivalent to the satisfiability (for $B_1, ..., B_p$) of the following mixed formula: $\forall \alpha, \beta : \alpha \in I \wedge \mathcal{T}(\alpha, \beta) \rightarrow$ "β satisfies $l_i \leq B_i$ for each $1 \leq i \leq p$". One may also ask a slightly different question:

> (*) For each infinite execution starting from I, are there $p \geq 1$ numbers $B_1, ..., B_p$ such that every configuration on the execution satisfies $l_i \leq B_i$ for each $1 \leq i \leq p$?

We call this question as the *mixed linear boundedness problem*, whose decidability is not obvious.

Theorem 6 *The mixed linear boundedness problem is decidable for mixed linear counter systems.*

Proof. Let M be a mixed linear counter system. Without loss of generality, we assume $p = 1$ (the other cases for p are similar). That is, we are given one linear combination l. An infinite execution is *unbounded* for l if for any B there is some configuration on the execution satisfying $l > B$. It suffices for us to consider the negation of the question statement (*): whether there is an unbounded infinite execution starting from I. The proof uses the idea of Theorem 3. Define a formula $\hat{\mathcal{T}}$ as follows. $\hat{\mathcal{T}}(s, \mathbf{X}, \mathbf{Y}, s', \mathbf{X}', \mathbf{Y}')$ is true iff the following two items are true:

- $(s, \mathbf{X}, \mathbf{Y})$ is reachable from some configuration in I,
- $(s, \mathbf{X}, \mathbf{Y})$ reaches $(s', \mathbf{X}', \mathbf{Y}')$; i.e., $\mathcal{T}(s, \mathbf{X}, \mathbf{Y}, s', \mathbf{X}', \mathbf{Y}')$,
- $l(\mathbf{X}, \mathbf{Y}) + 1 \leq l(\mathbf{X}', \mathbf{Y}')$.

The result follows immediately, noticing that $\hat{\mathcal{T}}$ is a transitive mixed linear relation and $\hat{\mathcal{T}}$ has an ω-chain iff M has an unbounded infinite execution from I. □

From their proofs, Theorem 6 and Theorem 3 (2) can be combined. For instance, the following question is decidable: is there an infinite execution of M that is P-live from I and that is unbounded for l?

Notice that, in (*), the bounds $B_1, ..., B_p$ are not uniform over all the infinite executions. To make them uniform, one might ask another different question by switching the quantifications in (*):

> (**) Are there numbers $B_1, ..., B_p$ such that, for each infinite execution starting from I, every configuration on the execution satisfies $l_i \leq B_i$ for each $1 \leq i \leq p$?

Currently, we do not know whether (**) is decidable or not. We leave this as an open question. However, the following question (by making $B_1, ..., B_p$ in (**) fixed, e.g., 0)

> is it true that, for each infinite execution starting from I, every configuration on the execution satisfies $l_i \leq 0$ for each $1 \leq i \leq p$?

is decidable, since its negation is equivalent to an eventuality problem.

One can easily find applications for Theorem 6. For instance, consider a system with two concurrent real-time processes running on one CPU. The processes are modeled as two discrete timed automata using a lock semaphore to achieve concurrency and using clocks to enforce timing constraints. The system is designed to be non-terminating and some fairness constraints are expected. We use t_1 (resp. t_2) to denote the total time that process 1 (resp. process 2) takes the CPU so far. One such constraint could be as follows. There is no infinite execution of the system on which the difference $|t_1 - t_2|$ is unbounded. This constraint can be automatically verified due to Theorem 6 and the fact, from Theorem 5, that the system, a discrete timed automaton augmented with two monotonic (and hence reversal-bounded) counters t_1 and t_2, is a mixed linear counter system.

6.2. Timed pushdown systems

There has been much interesting work on various verification problems for pushdown systems [4, 5, 6, 9, 10, 11, 13, 14]. Studying pushdown systems is important, since they are directly related to recursive programs and processes. In this subsection, we will study pushdown systems with discrete clocks and reversal-bounded counters. Safety verification for these systems is discussed in [10]. Here, we investigate the mixed linear liveness problem (since now we have only discrete variables, we call the problem as the Presburger liveness problem).

As we mentioned before, a timed automaton can be augmented with reversal-bounded counters. Here we only consider discrete clocks that take integer values. The discrete timed automaton can be further augmented with a pushdown stack. The resulting machine $\mathcal{A}$ is called a discrete pushdown timed automaton with reversal-bounded counters. In addition to counter operations and clock operations, $\mathcal{A}$ can push a symbol on the top of the stack, pop the top symbol from the stack, and test whether the top symbol of the stack equals some symbol. A configuration of $\mathcal{A}$ is a tuple of a control state, discrete clock values, counter values, and a stack word. The binary reachability $\mathcal{T}$ is the set of configurations pairs such that one can reach the other in $\mathcal{A}$. Each stack word w corresponds to an integer tuple $\mathbf{n} = (\mathbf{n}_{a^1}, \cdots, \mathbf{n}_{a^l})$, where $\{a^1, \cdots, a^l\}$ is the stack alphabet and each count $\mathbf{n}_{a^i}$ stands for the number of symbol a^i in w. The tuple $\mathbf{n}$ is also called the *stack word counts* for w. In this way, a set C of configurations corresponds to a predicate on states, clock values, counter values, and stack word counts. C is Presburger if the predicate is definable by a Presburger formula. C is *commutative* if, for any configurations c and c' satisfying that c and c' are the same except that the stack word in c is a permutation of the stack word in c', $c \in C$ iff $c' \in C$. In this case, the predicate exactly characterizes the set C. Let I and P be two Presburger subsets of configurations. We say $\mathcal{A}$ is *P-live from I* if there is an infinite sequence $c^0, \cdots, c^k, \cdots$ such that (1). $c^0 \in I$, (2). for all $k \geq 0$, $\mathcal{T}(c^k, c^{k+1})$, and (3). $c^k \in P$ for infinitely many k. The Presburger liveness problem for $\mathcal{A}$ is whether $\mathcal{A}$ is *P-live from I*, given I and P two Presburger subsets of configurations.

Theorem 7 *The Presburger liveness problem for discrete pushdown timed automata with reversal-bounded counters is decidable.*

Proof. Let $\mathcal{A}$ be a discrete pushdown timed automaton with reversal-bounded counters. We use $\mathbf{Y}$ to denote the discrete clocks and counters in $\mathcal{A}$. We use $\mathbf{n}$ to denote an integer tuple of stack word counts. Let I and P be two Presburger subsets of configurations of $\mathcal{A}$. Define $\hat{\mathcal{T}}$ as follows. $\hat{\mathcal{T}}(s, \mathbf{Y}, \mathbf{n}, a, s', \mathbf{Y}', \mathbf{n}', a')$ is true iff there are two stack words w and w' (called *witnesses*) such that

- (Condition 1) w is a (not necessarily proper) prefix of w',
- (Condition 2.1) w ends with stack symbol a (i.e., a is the top symbol of the stack word w),
- (Condition 2.2) w' ends with stack symbol a',
- (Condition 3.1) $\mathbf{n}$ is the stack word counts for w,
- (Condition 3.2) $\mathbf{n}'$ is the stack word counts for w',
- (Condition 4) configuration $(s, \mathbf{Y}, w)$ is reachable from some configuration in I,
- (Condition 5) configuration $(s, \mathbf{Y}, w)$ reaches configuration $(s', \mathbf{Y}', w')$ through a sequence of moves in $\mathcal{A}$, during which the top symbol a of w is not popped out and during which there is a configuration in P.

Assume that w'' and w''' witness $\hat{\mathcal{T}}(s, \mathbf{Y}, \mathbf{n}, a, s', \mathbf{Y}', \mathbf{n}', a')$. Observe that, for any w satisfying (Condition 2.1), (Condition 3.1) and (Condition 4), w and $w' = w + (w''' - w'')$ (i.e., w concatenated with the result of deleting the prefix w'' from w''') also witness $\hat{\mathcal{T}}(s, \mathbf{Y}, \mathbf{n}, a, s', \mathbf{Y}', \mathbf{n}', a')$. The reason is as follows. According to (Condition 5), the top a of w'' will not be popped out. That is, the content (instead of counts) of w'' is insensitive to (Condition 5). Therefore, (Condition 5) still holds when w'' is replaced with w as long as the prefix w'' of w''' is also replaced with w; i.e., (Condition 5) still holds for w and w'. This observation will be used in proving the following claim.

(**Claim 1**) $\hat{\mathcal{T}}$ has an ω-chain iff $\mathcal{A}$ is P-live from I.

Proof of (**Claim 1**). ($\Rightarrow$). Assume $\hat{\mathcal{T}}$ has an ω-chain

$$(s_0, \mathbf{V}_0, \mathbf{n}_0, a_0), \cdots, (s_k, \mathbf{V}_k, \mathbf{n}_k, a_k), \cdots.$$

Therefore, for each k, we have a pair of stack words w_k and w'_k that witness the fact of $\hat{\mathcal{T}}(s_k, \mathbf{V}_k, \mathbf{n}_k, a_k, s_{k+1}, \mathbf{V}_{k+1}, \mathbf{n}_{k+1}, a_{k+1})$. Now, take $w''_0 = w_0$, and for all $k \geq 1$, $w''_k = w_0 + (w'_0 - w_0) + \cdots + (w'_{k-1} - w_{k-1})$. Using the above observation, it can be easily shown that, for any $k \geq 0$, w''_k and w''_{k+1} witness

$$\hat{\mathcal{T}}(s_k, \mathbf{V}_k, \mathbf{n}_k, a_k, s_{k+1}, \mathbf{V}_{k+1}, \mathbf{n}_{k+1}, a_{k+1}).$$

Applying (Condition 4) on configuration $(s_0, \mathbf{V}_0, w''_0)$ and (Condition 5) on configurations $(s_k, \mathbf{V}_k, w''_k)$ and $(s_{k+1}, \mathbf{V}_{k+1}, w''_{k+1})$ for all $k \geq 0$, we can show $\mathcal{A}$ is P-live from I.

($\Leftarrow$). Assume $\mathcal{A}$ is P-live from I. That is, there is an infinite sequence $c^0, \cdots, c^k, \cdots$ such that (1). $c^0 \in I$, (2). for all $k \geq 0$, $\mathcal{T}(c^k, c^{k+1})$, and (3). $c^k \in P$ for infinitely many k. Without loss of generality, we assume that $\mathcal{A}$ leads c_k to c_{k+1} by running exactly one move, for all $k \geq 0$. Therefore, the stack word w_k in c_k and the stack word w_{k+1} in c_{k+1} satisfy one of the following conditions: (1). $w_k = w_{k+1}a$; i.e., the move pops a for some symbol a, (2). $w_{k+1} = w_k a$; i.e., the move pushes a for some symbol a, (3). $w_{k+1} = w_k$; i.e., the move does not change the stack. Notice that the stack has a special bottom symbol Z_0; i.e., every w_k starts with Z_0. The following technique has been used in several places (e.g., [18, 5]). For the sequence of the stack words $w_0, \cdots, w_k, \cdots$, define a strictly increasing sequence $k_0, \cdots, k_i, \cdots$ as follows.

k_0 is picked such that w_{k_0} is a prefix of each w_k with $k \geq 0$;

$k_1 > k_0$ is picked such that w_{k_1} is a prefix of each w_k with $k > k_0$;

$k_2 > k_1$ is picked such that w_{k_2} is a prefix of each w_k with $k > k_1$; etc.

Such a sequence always exists. Clearly, each w_{k_i} is a prefix of $w_{k_{i+1}}$ and from configuration c_{k_i} to configuration $c_{k_{i+1}}$, the top symbol of w_{k_i} is not popped out. Since there are infinitely many k with $c_k \in P$, there is a strictly increasing sequence $i^0, \cdots, i^j, \cdots$ such that, for all j, there is a k satisfying $c_k \in P$ and $k_{i^j} < k < k_{i^{j+1}}$. For each $j \geq 0$, we use $(s_j, \mathbf{V}_j, \mathbf{n}_j, a_j)$ to denote the control state, clock and counter values, the count vector of the stack word, and the top symbol of the stack word, respectively in configuration $c_{k_{i^j}}$. It is left to the reader to check

$$(s_0, \mathbf{V}_0, \mathbf{n}_0, a_0), \cdots, (s_j, \mathbf{V}_j, \mathbf{n}_j, a_j), \cdots$$

is an ω-chain of $\hat{\mathcal{T}}$, where, for all $j \geq 0$, $\hat{\mathcal{T}}(s_j, \mathbf{V}_j, \mathbf{n}_j, a_j, s_{j+1}, \mathbf{V}_{j+1}, \mathbf{n}_{j+1}, a_{j+1})$ is witnessed by $w_{k_{i^j}}$ and $w_{k_{i^{j+1}}}$.

Therefore, (**Claim 1**) is proved. Next, we are going to show that,

(**Claim 2**). $\hat{\mathcal{T}}(s, \mathbf{Y}, \mathbf{n}, a, s', \mathbf{Y}', \mathbf{n}', a')$ is a Presburger formula (when s, s', a, a' are understood as bounded integer variables).

Proof of (**Claim 2**). We build a machine M that accepts the domain (which are integer tuples) of $\hat{\mathcal{T}}$. Then we argue that integer tuples accepted by M are definable by a Presburger formula. M is a machine with a one-way input tape and a pushdown stack. M is also equipped with a number of counters, among which each clock in $\mathcal{A}$ corresponds a *clock-counter* in M and each reversal-bounded counter in $\mathcal{A}$ corresponds to a *rv-counter* in M. In addition, M contains a count-counter for each stack symbol and contains a number of other auxiliary counters. Whenever M pushes a to (resp. pops a from) the stack, the count-counter for a is incremented (resp. decremented) by one. So, a count-counter is used to record the number of a stack symbol in a stack word. M works as follows. Given an input

$$(s, \mathbf{Y}, \mathbf{n}, a, s', \mathbf{Y}', \mathbf{n}', a')$$

on M's input tape, where each integer in the above tuple is encoded as a unary string and separated by a delimiter, $\mathcal{M}$ starts to simulate $\mathcal{A}$ as follows. M guesses a control state for $\mathcal{A}$, a value for each clock-counter and a value for each rv-counter. At this moment, M makes sure that the stack is empty and each count-counter is 0. Then M guesses a stack word (by nondeterministically pushing symbols) and updates the count-counters accordingly. At some moment, M decides that I is satisfied by checking that the guessed control state, the clock-counter values, the rv-counter values, and the count-counters satisfy I. Doing this needs some auxiliary counters and needs only a finite number of counter reversals, since I is Presburger [19]. When this is checked out, M starts to simulate $\mathcal{A}$ (from the guessed state) using its own stack for the stack in $\mathcal{A}$, its own clock-counters for the clocks in $\mathcal{A}$ and its own rv-counters for the reversal-bounded counters in $\mathcal{A}$. All the transitions of $\mathcal{A}$ are faithfully simulated by M. In addition, whenever $\mathcal{A}$ pushes a to (resp. pops a from) the stack, M increments (resp. decrements) the count-counter for a by one. Nondeterministically at some moment, M decides to read the input tape by suspending the simulation. Then, M makes sure that the first half of the input $(s, \mathbf{Y}, \mathbf{n}, a)$ are consistent with the current configuration of $\mathcal{A}$. That is, the control state of $\mathcal{A}$ (remembered in $\mathcal{M}$'s finite control) is s, clock-counters and rv-counters have the same values as in $\mathbf{Y}$ (doing this needs auxiliary reversal-bounded counters), the stack top symbol is a, and count-counters have the same values as in $\mathbf{n}$ (doing this also needs auxiliary reversal-bounded counters). When these are checked out, (Condition 2.1), (Condition 3.1) and (Condition 4) are satisfied for the current configuration $(s, \mathbf{Y}, \mathbf{n}, a)$ of $\mathcal{A}$.

Then, M replaces the stack top symbol a with a new symbol $\hat{a}$ and resumes the simulation of $\mathcal{A}$. M makes sure that the simulation afterwards will not pop the new symbol out of M's stack. Nondeterministically at some moment later, M decides that the current configuration of $\mathcal{A}$ satisfies P. M checks that this is indeed true using its own counters. Similar to the previous scenario for I, this checking needs only a finite number of counter reversals and needs other auxiliary reversal-bounded counters. When this is checked out, M resumes the simulation of $\mathcal{A}$. Again, nondeterministically at some moment later, M shuts down the simulation and compares the rest of the input tape $(s', \mathbf{Y}', \mathbf{n}', a')$ with the control state of $\mathcal{A}$ in M's finite control, the clock-counter and rv-counter values of M, the count-counter values, and the top symbol of the stack. The comparisons make sure that (Condition

1), (Condition 2.2), (Condition 3.2) and (Condition 5) are satisfied by the current configuration of $\mathcal{A}$. M accepts the input if the comparisons are successful. Clearly, M accepts exactly the domain of $\hat{\mathcal{T}}$.

What are the counters in M? they are clock-counters, rv-counters, count-counters, and a number of other auxiliary reversal-bounded counters. All of them are reversal-bounded except the clock-counters and the count-counters. Each count-counter n_a can be treated as the difference $n_a^+ - n_a^-$ of two reversal-bounded counters n_a^+ and n_a^-: n_a^+ (resp. n_a^-) is used to record the number of pushes (resp. pops) of a. So, each count-counter can be simulated by two reversal-bounded counters. How about clock-counters? In [10] (see also its full version), a technique is proposed such that, as far as binary reachability is concerned, discrete clocks can be simulated by reversal-bounded counters [a]. Therefore, clock-counters can be made reversal-bounded from the start of simulating $\mathcal{A}$ to the moment checking P, and, from the moment checking P to shutting down $\mathcal{A}$. Hence, M only has reversal-bounded counters as well as a pushdown stack. Therefore, M is a reversal-bounded multicounter machine with a pushdown stack and a one-way input tape (NPCM). It is known that NPCMs accepts semilinear languages [19]. In particular, since M accepts a language in the form of integer tuples, the language is definable by a Presburger formula [19]. Hence, $\hat{\mathcal{T}}$ is Presburger. Therefore, (**Claim 2**) is proved.

Since a Presburger formula is a special form of a mixed linear relation, Theorem 7 is followed from (**Claim 1**), (**Claim 2**), and Theorem 2. □

We are not able to extend the result of Theorem 7 to dense clocks. The pattern technique [9] that abstracts a dense clock into a discrete clock and a pattern does not apply here. This is because the abstraction maintains the exact binary reachability of dense clocks, but does not maintain the exact dense clock values between the binary reachability. Timed pushdown systems with reversal-bounded counters dealt in Theorem 7 also have a lot of applications. For instance, it can be used to model some real-time recursive concurrent programs. The reversal-bounded counters can also be used to count the number of external events – these counts can be later used to specify some fairness constraints on the environment.

7. Conclusions

In this paper, we showed that it is decidable whether a transitive mixed linear relation has an ω-chain. Using this main theorem, we were able to establish, within a unified framework, a number of liveness verification results on generalized timed automata. More precisely, we proved that (1) the mixed linear liveness problem for timed automata with dense clocks, reversal-bounded counters, and a free counter is decidable, and (2) the Presburger liveness problem for timed automata with discrete clocks, reversal-bounded counters, and a pushdown stack is decidable. The results can be used to analyze some fairness constraints (e.g., livelock-free and starvation-free) for infinite-state concurrent systems.

Our results are useful in formulating a decidable subset of linear temporal logic

[a] More precisely, discrete clocks in $\mathcal{A}$ can be replaced by reversal-bounded counters (the result is called $\mathcal{A}'$) such that, whenever c_1 can reach c_2 in $\mathcal{A}$, c_1 can reach c_2 in $\mathcal{A}'$ [10].

(LTL) for a class of timed automata augmented with counters. Let $\mathcal{A}$ be a timed automaton with dense clocks, reversal-bounded counters, and a free counter. The set of linear temporal logic formulas $\mathcal{L}_{\mathcal{A}}$ with respect to $\mathcal{A}$ is defined by the following grammar:

$$\phi := P|\neg\phi|\phi \wedge \phi| \bigcirc \phi|\phi U \phi$$

where P is a set of configurations of $\mathcal{A}$ definable by a mixed formula (on control states, dense clocks, reversal-bounded counters, and the free counter). $\bigcirc$ denotes "next", and U denotes "until". Formulas in $\mathcal{L}_{\mathcal{A}}$ are interpreted on infinite execution sequences p of configurations of $\mathcal{A}$ in the usual way. This logic is very similar to the Presburger LTL for timed automata with discrete clocks [12] except that P is a mixed formula instead of a Presburger formula.

The *satisfiability-checking problem* is to check, given $\mathcal{A}$ and $\phi \in \mathcal{L}_{\mathcal{A}}$, whether there exists an infinite execution p of $\mathcal{A}$ with $p \models \phi$. From Corollary 2, the satisfiability-checking problems are decidable for the following LTL formulas:

- $I \wedge \Box\Diamond P$.
- $I \wedge \Diamond P$.
- $I \wedge \Box\Diamond P \wedge \Box\Diamond Q$.

In our previous paper [12], the first two items as above were shown but only for timed automata with discrete clocks. In the same paper, the last item as above was left open.

Some work needs to be done in the future in formulating an exact decidable subset (broader than the subset in Comon and Cortier [7]) of $\mathcal{L}_{\mathcal{A}}$ for satisfiability-checking. Notice that the entire $\mathcal{L}_{\mathcal{A}}$ is undecidable for satisfiability-checking/model-checking, even when the next operator is dropped from the logic. This is because the satisfiability-checking problem for $\Box P$ is undecidable, when $\mathcal{A}$ is a discrete timed automaton, as shown in [12].

A similar decidable subset of LTL formulas $\mathcal{L}_{\mathcal{A}}$ could be formulated for discrete timed pushdown systems, by combining Theorem 7, the results in [10] and [20]. Another issue is on the complexity analysis of the decision procedures presented in Theorem 3 and Theorem 7. However, this issue is related to the complexity for the emptiness problem of NPCMs, which is still unknown, though it is believed that it can be derived along Gurari and Ibarra [15].

Acknowledgements

We would like to thank the following students at WSU for reading an earlier draft of this paper: K. Gjermundrod, H. He, A. Khodjanov, C. Li, J. Nelson, G. Xie, and L. Yang. The work by Oscar H. Ibarra has been supported in part by NSF Grant IIS-0101134. Thanks also go to the anonymous referees for many useful suggestions.

References

1. R. Alur, C. Courcoibetis and D. Dill, "Model-checking in dense real time," *Information and Computation* **104** (1993) 2–34.
2. R. Alur and D. Dill, "A theory of timed automata," *Theoretical Computer Science* **126** (1994) 183–236.
3. R. Alur and T. A. Henzinger, "A really temporal logic," *J. ACM* **41** (1994) 181–204.
4. T. Ball and S. K. Rajamani, "Bebop: a symbolic model-checker for Boolean programs," in *SPIN Model Checking and Software Verification, Proc. 7th Int. SPIN Workshop*, eds. K. Havelund, J. Penix and W. Visser (Springer, Berlin, 2000) pp. 113–130.
5. A. Bouajjani, J. Esparza and O. Maler, "Reachability analysis of pushdown automata: application to model-checking," in *CONCUR'97, Proc. 8th Int. Conf. on Concurrency Theory*, eds. A. W. Mazurkiewicz and J. Winkowski (Springer, Berlin, 1997) pp. 135–150.
6. A. Bouajjani, R. Echahed and R. Robbana, "On the automatic verification of systems with continuous variables and unbounded discrete data structures," in *Hybrid Systems II*, eds. P. J. Antsaklis, W. Kohn, A. Nerode and Shankar Sastry (Springer, Berlin, 1995) pp. 64–85.
7. H. Comon and V. Cortier, "Flatness is not a weakness," in *Computer Science Logic, Proc. 14th Annual Conf. of the EACSL*, eds. P. Clote and H. Schwichtenberg (Springer, Berlin, 2000) pp. 262–276.
8. H. Comon and Y. Jurski, "Timed automata and the theory of real numbers," in *CONCUR'99, Proc. 10th Intl. Conf. on Concurrency Theory*, eds. Jos C. M. Baeten and S. Mauw (Springer, Berlin, 1999) pp. 242–257.
9. Z. Dang, "Binary reachability analysis of pushdown timed automata with dense clocks," in *CAV'01, Proc. 13th Intl. Conf. on Computer Aided Verification,* eds. G. Berry, H. Comon and A. Finkel (Springer, Berlin, 2001) pp. 506–517.
10. Z. Dang, O. H. Ibarra, T. Bultan, R. A. Kemmerer and J. Su, "Binary reachability analysis of discrete pushdown timed automata," in *CAV'00, Proc. 12th Intl. Conf. on Computer Aided Verification,* eds. E. A. Emerson, A. P. Sistla (Springer, Berlin, 2000) pp. 69–84.
11. Z. Dang, O. H. Ibarra and P. San Pietro, "Liveness verification of reversal-bounded multicounter machines with a free counter," in *FSTTCS'01, Proc. 21st Intl. Conf. on Foundations of Software Technology and Theoretical Computer Science,* eds. R. Hariharan, M. Mukund and V. Vinay (Springer, Berlin, 2001) pp. 132–143.
12. Z. Dang, P. San Pietro and R. A. Kemmerer, "On Presburger liveness of discrete timed automata," in *STACS 2001, Proc. 18th Annual Symp. on Theoretical Aspects of Computer Science,* eds. A. Ferreira and H. Reichel (Springer, Berlin, 2001) pp. 132–143.
13. J. Esparza and S. Schwoon, "A BDD-based model-checker for recursive programs," in *CAV'01, Proc. 13th Intl. Conf. on Computer Aided Verification,* eds. G. Berry, H. Comon and A. Finkel (Springer, Berlin, 2001) pp. 324–336.
14. A. Finkel, B. Willems and P. Wolper, "A direct symbolic approach to model checking pushdown systems," *Electronic Notes in Theoretical Computer Science* **9** (2000).
15. E. Gurari and O. Ibarra, "The complexity of decision problems for finite-turn multicounter machines," *J. Comput. System Sci.* **22** (1981) 220–229.
16. T. A. Henzinger and Pei-Hsin Ho, "HyTech: the Cornell hybrid technology tool," in *Hybrid Systems II*, eds. P. J. Antsaklis, W. Kohn, A. Nerode and Shankar Sastry (Springer, Berlin, 1995) pp. 265–294.

17. T. A. Henzinger, X. Nicollin, J. Sifakis and S. Yovine, "Symbolic model checking for real-time systems," *Information and Computation* **111** (1994) 193–244.

18. J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation* (Addison-Wesley, Reading, 1979).

19. O. H. Ibarra, "Reversal-bounded multicounter machines and their decision problems," *J. ACM* **25** (1978) 116–133.

20. O. H. Ibarra and Z. Dang, "On removing the stack from reachability constructions with applications to Presburger LTL satisfiability-checking," in *ISAAC 2001, Proc. 12th Intl. Conf. on Algorithms and Computation* eds. P. Eades and T. Takaoka (Springer, Berlin, 2001) pp. 244–256.

21. K. G. Larsen, P. Pattersson and W. Yi, "UPPAAL in a nutshell," *Intl. J. on Software Tools for Technology Transfer* **1** (1997) 134–152.

22. F. Laroussinie, K. G. Larsen and C. Weise, "From timed automata to logic - and back," in *MFCS'95, Proc. 20th Intl. Symp. on Mathematical Foundations of Computer Science* eds. J. Wiedermann and P. Hajek (Springer, Berlin, 1995) pp. 529–539.

23. J. Raskin and P. Schobben, "State clock logic: a decidable real-time logic," in *HART'97, Proc. Intl. Workshop on Hybrid and Real-Time Systems* eds. O. Maler (Springer, Berlin, 1997) pp. 33–47.

24. V. Weispfenning, "Mixed real-integer linear quantifier elimination," *Proc. Intl. Symp. on Symbolic and Algebraic Computation*, Vancouver, B.C., Canada, July 29-31, 1999, pp. 129–136.

25. T. Wilke, "Specifying timed state sequences in powerful decidable logics and timed automata," in *FTRTFT'94, Proc. 3rd Intl. Conf. on Formal Techniques in Real-Time and Fault-Tolerant Systems* eds. H. Langmaack, W. P. de Roever and J. Vytopil (Springer, Berlin, 1994) pp. 694–715.