Constraint LTL with Remote Access

Ashwin Bhaskar
Chennai Mathematical Institute, India

M. Praveen
Chennai Mathematical Institute, India
CNRS IRL ReLaX, Chennai, India

Abstract

Constraint Linear Temporal Logic (CLTL) is an extension of LTL that is interpreted on sequences of valuations of variables over an infinite domain. The atomic formulas are interpreted as constraints on the valuations. The atomic formulas can constrain valuations at the current position and positions that are a fixed distance apart (e.g., the previous position or the second previous position and so on). The satisfiability problem for CLTL is known to be Pspace-complete. We generalize CLTL to let atomic formulas access positions that are unboundedly far away in the past. We annotate the sequence of valuations with letters from a finite alphabet and use regular expressions on the finite alphabet to control how atomic formulas access past positions. We prove that the satisfiability problem for this extension of the logic is decidable in cases where the domain is dense and open with respect to a linear order (e.g., rational numbers with the usual linear order). We prove that it is also decidable over integers with linear order and equality.

2012 ACM Subject Classification Theory of computation → Logic and verification; Theory of computation → Modal and temporal logics; Theory of computation → Verification by model checking; Theory of computation → Automata over infinite objects

Keywords and phrases Constraint LTL, Regular Expressions, MSO formulas, Satisfiability, Büchi automata

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2023.41

Funding *M. Praveen*: This author is partially supported by the Infosys foundation.

1 Introduction

Propositional Linear Temporal Logic (LTL) and related automata-theoretic models have been extended in various ways to make it more expressive. Constraint LTL (abbreviated as CLTL in [12]), LTL with freeze operators [9], temporal logic of repeating values [7, 19], finite memory automata [16], data automata [3] are all examples of this. These extensions are concerned with using variables that range over infinite domains in place of Boolean propositions used in propositional LTL. Variables ranging over infinite domains are a natural choice for writing specifications for systems that deal with infinite domains. For example, CLTL has been used for specifications of cloud-based elastic systems [1], where the domain of natural numbers is used to reason about the number of resources that are being used by cloud-based systems. In CLTL, atomic formulas can refer to valuations in previous/next positions using operators like X^{-1}, X^{-2}, X, X^2 etc.

We generalize CLTL by introducing operators of the form $\textcircled{e}^{-1}$, where e is a regular expression over some finite alphabet. To illustrate the intended meaning of these operators, suppose $e = a^*b$. The operator $\textcircled{$a^*b^{-1} refers to a position in the past such that the sequence of letters between the current and that past position is in the language of the expression a^*b. The operator X^{-2} is same as $\textcircled{$\Sigma \cdot \Sigma \cdot \Sigma^{-1}. We call this extension Constraint LTL with remote access (RCLTL). CLTL is a fragment of this, where the expressions e are constrained to be of the form Σ^n for some number n. Apart from being a natural extension of CLTL, RCLTL is inspired by the usage of automata to model smart contracts [18], where components interact

43rd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2023).
Editors: Patricia Bouyer and Srikanth Srinivasan; Article No. 41; pp. 41:1–41:21
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

with each other by passing messages. For example, bidders in an online bidding forum may pass messages to propose a bidding value, or to request the current highest bid. Suppose we want to verify that any bid is always strictly greater than the previous bid. However, the previous bid may not have been proposed in the previous step: several past steps may have been requests for the current highest bid. If we represent bidding messages by the letter b and request messages by the letter r, we can access the position where the last bid was made using the operator $\textcircled{$r^*b^{-1}. This operator can refer to past positions that are unboundedly far behind to access the last bid, which cannot be done in CLTL.

Contributions. The main contribution of this paper is to prove that RCLTL is decidable over the domain of integers with equality and linear order. At a high level, the techniques we use are similar to those used for proving decidability of satisfiability of CLTL [12], based on patterns that repeat in ultimately periodic words. However, the fact that RCLTL operators can access positions that are remotely in the past poses many challenges to be overcome. E.g., suppose a variable x has value 1 in the previous position and value 2 in the current position, so the constraint $X^{-1}x < x$ is satisfied. In CLTL, the values 1 and 2 are abstracted away and only the constraint $X^{-1}x < x$ is retained. This constraint spans two consecutive positions, referred to as a frame of length 2 in [12]. However in RCLTL, a constraint of the form $\textcircled{$r^*b$}^{-1}x < \textcircled{$\epsilon$}^{-1}x$ can span unboundedly many positions, so the concept of frames of fixed size cannot be used. We introduce memoirs to handle this. In turn, many properties of frames are not true for memoirs, so new techniques have to be introduced to handle them. Details of such differences with previous work will be highlighted at appropriate sections of this paper.

Related works. The satisfiability of the description logic $\mathcal{ALCF}^{\mathcal{P}}(\mathcal{Z}_c)$ has been studied in [17] and has been proven to be EXPTIME-complete. The logic $\mathcal{ALCF}^{\mathcal{P}}(\mathcal{Z}_c)$ subsumes CLTL $(\mathbb{Z}, <, =)$. In [11], the authors prove that the satisfiability of CTL* with constraints over $(\mathbb{Z}, <, =)$ is 2EXPTIME-complete. Again clearly, CTL* with constraints is an extension for CLTL $(\mathbb{Z}, <, =)$. In both these papers, the tree automata technique used is based on identifying repeating patterns over regular trees. The chapter “Functional Specification of Hardware via Temporal Logic” in [13] introduces an extension of LTL with regular expressions and a suffix implication operator.[1] This results in a temporal logic that extends LTL and is as expressive as ω-regular languages. Note however that, unlike in [13] where the authors extend the set of temporal operators, in our paper we extend the atomic formulas with regular expressions. Also note that the variables in our work range over infinite domains whereas [13] only deals with Boolean propositions. Atomic formulas are extended also in LTL with freeze operators [9] and temporal logic of repeating values [7, 19]. Like our extension, the atomic formulas there can also compare current values with values in other positions that are unboundedly far away. However, in our extension the other position is chosen by a pattern of letters from a finite alphabet, whereas the atomic formulas in LTL with freeze operators and temporal logic of repeating values can choose other positions based on the values from the infinite domain. For example, in LTL with freeze operators, an atomic formula can check that there is some position in the past where the value is equal to the value at the current position. The negation of this formula checks the current value with all the previous values. In the logic we propose here, only two positions can be compared. This feature of LTL with freeze operators is quite powerful and satisfiability is undecidable in general, while the logic we propose here is decidable.

[1] We thank an anonymous reviewer for pointing this work to us.

Temporal Stream Logic (TSL) [15, 14] extends LTL with updates and predicates over arbitrary function terms. TSL is designed to track how variable values evolve over time. The satisfiability problem for TSL is not decidable but approaches to be used in practice have been proposed. Constraint LTL over clocks [2] is a variant of CLTL and many other time-domain extensions of LTL (such as Metric Interval Temporal Logic and Quantified Temporal Logic) can be translated to CLTL over clocks. The satisfiability problem for this logic has been shown to be decidable, by reducing it to a decidable Satisfiability Modulo Theory problem. Presburger LTL is an extension of LTL where atomic formulas are quantifier-free Presburger formulas. The satisfiability problem for this logic is undecidable in general. Many decidable fragments have been proposed [8, 6, 4]. The survey [10] mentions many other logics that are extended at the atomic level to deal with concrete domains.

2 Preliminaries

Let $\mathbb{Z}$ be the set of integers and $\mathbb{N}$ be the set of non-negative integers. For integers n_1, n_2, we denote by $[n_1, n_2]$ the set $\{n \in \mathbb{Z} \mid n_1 \leq n \leq n_2\}$. We recall the definitions of constraint systems and Constraint Linear Temporal Logic (CLTL) from [12]. A constraint system $\mathcal{D}$ is of the form $(D, R_1, \ldots, R_n, \mathcal{I})$, where D is a non-empty set called the domain. Each R_i is a predicate symbol of arity a_i, with $\mathcal{I}(R_i) \subseteq D^{a_i}$ being its interpretation.

We introduce an extension of Constraint LTL (CLTL) to let atomic formulas access positions in the past that are unboundedly far away. Let V be a finite set of variables, Σ be a finite alphabet and let E be a finite set of regular expressions over Σ. A term t over V and E is of the form $\textcircled{e}^{-1}v$, where $v \in V$, $e \in E$[2]. Let T_E^V denote the set of all terms over V and E. A constraint c is of the form $R(t_1, \ldots, t_n)$, where R is a predicate symbol of arity n and $t_1, \ldots, t_n$ are terms. The syntax of Constraint LTL with Remote access (RCLTL) is given by the following grammar, where c is a constraint as defined above.

$$\phi ::= c \mid \neg\phi \mid \phi \vee \phi \mid X\phi \mid \phi U \phi$$

Let F_D denote the set of all mappings of the form $f \colon V \to D$. The semantics of RCLTL is defined with respect to infinite sequences over $\Sigma \times F_D$ (also called concrete models in the following). Given a concrete model σ, a regular expression $e \in E$ and positions $j, i \in \mathbb{N}$, we write $\sigma[j, i] \in \mathcal{L}(e)^R$ to denote that the infix of σ between the positions j and i (both j and i inclusive), projected to Σ is in the reverse of the language of e. We say that j is the minimal match for e at i if $\sigma[j, i] \in \mathcal{L}(e)^R$ and for every $j < j' \leq i$, $\sigma[j', i] \notin \mathcal{L}(e)^R$ (the word "minimal" refers to minimality with respect to the length of infixes ending at position i). In other words, $\sigma[j, i]$ belongs to $\mathcal{L}(e)^R$, but none of its suffixes belong to $\mathcal{L}(e)^R$. A regular expression $e \in E$ is said to be relevant at position i of a concrete model if there exists a position $j \geq 0$ that is the minimal match for e at i. A term $t = \textcircled{e}^{-1}v$ is said to be relevant at a position i if e is relevant at i. Given a position i and a term $\textcircled{e}^{-1}v$ relevant at i, we write $d(i, e, v)$ (resp. $\mathrm{mmp}(i, e)$) to denote the value $f_j(v) \in D$ (resp. j) such that j is the minimal match for e at i. Intuitively, if we start at position i, go backwards and j is the first position such that the infix $\sigma[j, i]$ belongs to $\mathcal{L}(e)^R$, then $d(i, e, v)$ is the value assigned to v at that position j and $\mathrm{mmp}(i, e) = j$, the minimal matching position.

[2] This choice of notation is inspired by a convention of LTL syntax, where $O\phi$ is intended to mean next ϕ. We write O^{-1} to mean past and insert e inside the O to indicate that we want the path between the current and past positions to match e, similar to the syntax used in dynamic logics.

Given $v_1, \ldots, v_n \in V$ and $e_1, \ldots, e_n \in E$, the i^{th} position of a concrete model σ satisfies the constraint $R(\enclose{circle}{e_1}^{-1}v_1, \ldots, \enclose{circle}{e_n}^{-1}v_n)$ (written as $\sigma, i \models R(\enclose{circle}{e_1}^{-1}v_1, \ldots, \enclose{circle}{e_n}^{-1}v_n)$) if $e_1, \ldots, e_n$ are all relevant at i and $(d(i, e_1, v_1), \ldots, d(i, e_n, v_n)) \in \mathcal{I}(R)$. Intuitively, the terms $\enclose{circle}{e_1}^{-1}v_1, \ldots, \enclose{circle}{e_n}^{-1}v_n$ refer to variables at previous positions and $R(\enclose{circle}{e_1}^{-1}v_1, \ldots, \enclose{circle}{e_n}^{-1}v_n)$ imposes a constraint on the values of those variables at those positions. Note that $\sigma, i \models \enclose{circle}{e}^{-1}v = \enclose{circle}{e}^{-1}v$ iff e is relevant at i. E.g., the expression b^+a is relevant at position i if "b" is the letter at i, $j < i$ is the last position before i that has "a" and all positions between j and i carry "b"; j is the minimal matching position.

The semantics is extended to rest of the RCLTL syntax similar to the usual propositional LTL. We use the standard abbreviations $F\phi$ (resp. $G\phi$) to mean that ϕ is true at some position (resp. all positions) in the future. The formula $XG(\enclose{circle}{b^*a}^{-1}v < \enclose{circle}{a^*b}^{-1}v)$ will be true if in all positions other than the first, the value of v at the last position with an "a" is less than the value at the last position with "b". In the rest of this paper we will only consider constraint systems of the form $(D, <, =)$ where D is an infinite set, $=$ is the equality relation and $<$ is a linear order. Some proofs and technical details in the subsequent sections are moved to the appendix due to space constraints.

3 Symbolic Models

CLTL (introduced in [12]) can be considered as a restriction of RCLTL where the expressions in E are restricted to be of the form Σ^n for some $n \in \mathbb{N}$. The maximum such n used in a given CLTL formula is called the X-length of the formula. The models of CLTL are infinite sequences over an infinite alphabet. The notion of frames was introduced in [12] in order to abstract the infinite alphabet to a finite one. A frame for a given CLTL formula, as introduced in [12], is the set of all constraints satisfied by valuations along some infix of some concrete model, the length of the infix being bound by the X-length of the formula. The spans of these frames are equal to the X-length of the CLTL formula. In a fundamental deviation from [12], we introduce here the notion of a *memoir*, which can span positions that are unboundedly far away from one another.

▶ **Definition 1** (Memoirs and symbolic models). *Let ϕ be an RCLTL formula with set of variables V and expressions E. A* memoir *is a total pre-order*[3] *$\leq$ on a subset of terms over V and E. A* symbolic model *ρ is an infinite sequence in $(\Sigma \times M)^\omega$, where M is the set of all memoirs.*

Given a concrete model σ, we associate with it a symbolic model $\mu(\sigma)$ as follows: $\mu(\sigma)$ is an infinite sequence such that for every $i \geq 0$, $e_1, e_2 \in E$ and $v_1, v_2 \in V$, the i^{th} memoir has the relation $\enclose{circle}{e_1}^{-1}v_1 \leq_i \enclose{circle}{e_2}^{-1}v_2$ iff e_1 and e_2 are relevant at position i and $d(i, e_1, v_1) \leq d(i, e_2, v_2)$. Intuitively, the memoir at position i of $\mu(\sigma)$ stores in its memory the relations between all variables in those past positions which can be accessed from position i through one of the given regular expressions.

For a symbolic model ρ, let $\leq_i$ be the memoir at position i. We denote by $<_i$ and $\equiv_i$ the strict order and equivalence relation induced by $\leq_i$: $t_1 <_i t_2$ iff $t_1 \leq_i t_2$ and $t_2 \not\leq_i t_1$, and $t_1 \equiv_i t_2$ iff $t_1 \leq_i t_2$ and $t_2 \leq_i t_1$. We define symbolic semantics $\models_s$, to interpret RCLTL formulas on symbolic models. Minimal matches in symbolic models are just like those in concrete models: we say that j is the minimal match for expression e at position i of a symbolic model ρ if $\rho[j, i] \in \mathcal{L}(e)^R$ and for every $j < j' \leq i$, $\rho[j', i] \notin \mathcal{L}(e)^R$. The definitions

[3] a reflexive and transitive relation such that for all a, b, either $a \leq b$ or $b \leq a$.

of the relevance of regular expressions and terms are the same for both symbolic and concrete models. The i^{th} position of ρ symbolically satisfies the constraint $t_1 < t_2$ (where t_1, t_2 are terms), if t_1, t_2 are relevant at i and $t_1 <_i t_2$ holds. Formally, we write it as $\rho, i \models_s t_1 < t_2$. The case of the equality constraint is similar. The symbolic satisfaction relation $\models_s$ is extended to all RCLTL formulas by induction on the structure of the formula as done for propositional LTL. Following is the RCLTL counterpart of a similar result [12, Lemma 3.1] for CLTL. The proof is by a routine induction on the structure of formulas.

▶ **Lemma 2.** *Let ϕ be an RCLTL($\mathcal{D}$) formula. Let σ be a concrete model and $\rho = \mu(\sigma)$. Then $\sigma, 0 \models \phi$ iff $\rho, 0 \models_s \phi$.*

For every concrete model, there is an associated symbolic model. However, there may be symbolic models that are not associated with any concrete models – symbolic models are intended to be abstractions of concrete models, but some combination of abstract symbols may not make sense. For example, the symbolic model ρ may have memoirs that are *irrelevant*: the memoir at some position i may have the relation $\textcircled{$e_1$}^{-1}v_1 \leq_i \textcircled{e_2}^{-1}v_2$, but there may not be any position that is a match for e_1 at i. We say that a symbolic model ρ admits a concrete model if there exists a concrete model σ such that $\rho = \mu(\sigma)$.

▶ **Definition 3.** ***Satisfiability Problem for RCLTL**: Given an RCLTL($\mathcal{D}$) formula ϕ, the satisfiability problem is to check if there exists a concrete model that satisfies ϕ*

From Lemma 2, we have the following corollary.

▶ **Corollary 4** ([12, Corollary 4.1]). *An RCLTL($\mathcal{D}$) formula ϕ is satisfiable iff there exists a symbolic model ρ such that $\rho, 0 \models_s \phi$ and ρ admits a concrete model.*

To check whether $\rho, i \models_s t_1 < t_2$ we only need to check $\rho(i)$, the i^{th} memoir in ρ, unlike the RCLTL semantics, where we may need to check other positions also. In this sense, the symbolic semantics lets us treat RCLTL formulas as if they are formulas in propositional LTL and employ techniques that have been developed for propositional LTL. To get decidability of RCLTL satisfiability, we additionally need to check if symbolic models admit concrete ones. For this, as mentioned before, we first need to check that memoirs in symbolic models are relevant.

Another reason why symbolic models may not admit concrete ones is that the symbolic model may be inconsistent: one memoir may say that the value of variable x at position i should be less than the value of y at position j, but another memoir may say the opposite (i.e., value of x at i should be greater than the value of y at j). To explain how we tackle this, we use a graph associated with symbolic models, as done in [12]. For a symbolic model ρ, we define a corresponding $\{<, =\}$-labelled, directed graph G_ρ over the set of vertices $V \times \mathbb{N}$. There is a $\sim$-labelled edge (for $\sim \in \{<, =\}$) from (x, i) to (y, j) (written as $(x, i) \xrightarrow{\sim} (y, j)$) if and only if there exists $k \geq i, j$ and there exist $e_1, e_2 \in E$ such that i is the minimal match for e_1 at k, j is the minimal match for e_2 at k and the constraint $\textcircled{$e_1$}^{-1}x \sim_k \textcircled{e_2}^{-1}y$ holds. Intuitively, the total pre-order at position k imposes the constraint $\sim$ between the variable x at position i and the variable y at position j. It follows that if there is an $=$-labelled edge from (x, i) to (y, j) then there must also be an $=$-labelled edge from (y, j) to (x, i). We define a strict edge to be an edge labelled with $<$ and a directed path is said to be strict if it contains at least one strict edge.

We say that G_ρ has a strict cycle if it has a directed cycle containing at least one edge labelled with $<$. Strict cycles indicate the presence of inconsistencies in symbolic models, as explained before. If G_ρ is the graph associated with a symbolic model ρ, then we need to check if between every pair of vertices in G_ρ there is at most one directed edge labelled with $<$, and that there are no strict cycles. Notice here that given a concrete model σ, $G_{\mu(\sigma)}$ does not have any strict cycle. Depending on the letters from Σ present along the positions of a symbolic model, the minimal match for some expression e at some position i can be unboundedly far behind i. So the presence/absence of an edge between a pair of vertices in G_ρ can potentially depend on memoirs that are unboundedly far in the future. This is a deviation from [12], whose frames can only span positions bounded by the X-length of the given formula (so consistency checks only need to be performed at bounded lengths).

Given a symbolic model ρ, it is easy to see that ρ admits a concrete model over a domain D iff there exists a labelling $l \colon V \times \mathbb{N} \to D$ of the vertices of G_ρ such that the labelling respects the edges in G_ρ. In other words, for all $x, y \in V, i, j \in \mathbb{N}$, $l((x,i)) = l((y,j))$ if there exists a directed path from (x,i) to (y,j) consisting of only $=$-labelled edges and $l((x,i)) < l((y,j))$ if there exists a strict directed path from (x,i) to (y,j). We call such a labelling of vertices in G_ρ, an edge-respecting labelling.

We say a domain D is dense (with respect to the ordering $<$) if for each $d, d' \in D$ with $d < d'$, there exists $d'' \in D$, such that $d < d'' < d'$. We say D is open if for each $d \in D$, there exist $d', d'' \in D$ such that $d' < d < d''$. $(\mathbb{R}, <, =)$ and $(\mathbb{Q}, <, =)$ are examples of constraint systems where the domains are dense and open. We say a symbolic model is consistent if all its memoirs are relevant and G_ρ does not have strict cycles. Using the proof ideas from Lemma 5.3 and Lemma 6.1 of [12], we get the following lemma:

▶ **Lemma 5** ([12, Lemma 5.3, Lemma 6.1]). *Let $\mathcal{D}$ be a constraint system of the form $(D, <, =)$ where D is infinite, $<$ is a linear order and D is dense and open with respect to $<$. Then all consistent symbolic models admit concrete models over D.*

In the next section we will give an MSO characterization of symbolic models with all memoirs relevant and whose associated graphs G_ρ have no strict cycles. This will help us in deciding the satisfiability problem for RCLTL($\mathcal{D}$) over dense and open domains.

4 A general decidability result

In this section, we give a decision procedure for the satisfiability problem for RCLTL over domains that are dense and open. Corollary 4 suggests a way to address the satisfiability problem for RCLTL. Given an RCLTL formula ϕ, we know that we can construct a Büchi automaton that recognizes the set of all symbolic models over $\Sigma \times M$ which symbolically satisfy ϕ [21]. If we could build two Büchi automata, one that recognizes the set of symbolic models with all memoirs relevant and with no strict cycle in the associated graph, and one that recognizes symbolic models that admit a concrete model, then we can intersect these three automata and check the resulting automaton for non-emptiness to decide the satisfiability for ϕ.

There are constraint systems for which the set of symbolic models that admit concrete models is not ω-regular. However, as we saw in Lemma 5, for constraint systems whose domains are dense and open, every consistent symbolic model admits a concrete model, hence constructing an automaton that accepts the set of consistent symbolic models suffices.

We now give a characterization of consistent symbolic models in the following lemma:

▶ **Lemma 6.** *Let $\mathcal{L}_{symb}$ denote the set of all consistent symbolic models ρ over $(\Sigma \times M)^\omega$. Then, $\mathcal{L}_{symb}$ is ω-regular.*

Proof. We prove that $\mathcal{L}_{symb}$ is ω-regular by constructing an MSO formula over the alphabet $\Sigma \times M$ defining $\mathcal{L}_{symb}$. We shall define two MSO formulas: ψ_{rel} whose language is the set of all symbolic models with all memoirs relevant and $\psi_{strcycle}$ whose language is the set of all symbolic models ρ whose associated graphs G_ρ have no strict cycles.

We first define the unary predicate symbols in the MSO vocabulary and their semantics. Let the set of predicate symbols be $P = \{P_a \mid a \in \Sigma\} \cup \{P_{\textcircled{e_1}^{-1}v_j \sim \textcircled{e_2}^{-1}v_k} \mid e_1, e_2 \in E, v_j, v_k \in V, \sim \in \{<, =\}\}$. Given a symbolic model ρ and a position i, $\rho, i \models i \in P_a$ iff $\rho(i)$ is of the form $(a, \leq_i)$ where $\leq_i$ is any memoir in M. We use $P_{\textcircled{e_1}^{-1}v_j \leq \textcircled{e_2}^{-1}v_k}$ as a shorthand for $P_{\textcircled{e_1}^{-1}v_j < \textcircled{e_2}^{-1}v_k} \vee P_{\textcircled{e_1}^{-1}v_j = \textcircled{e_2}^{-1}v_k}$. We also say that an edge of G_ρ is labelled with $\leq$ if it is labelled either with $<$ or $=$. We have $\rho, i \models i \in P_{\textcircled{e_1}^{-1}v_j \sim \textcircled{e_2}^{-1}v_k}$ iff the memoir at $\rho(i)$ induces the relation $\textcircled{e_1}^{-1}v_j \sim \textcircled{e_2}^{-1}v_k$.

We now define a few MSO formulas that will aid us in constructing the formula ψ_{rel}.

Let $\text{matches}_e(pos, match)$ be an MSO formula which evaluates to true iff the position $match$ is a match for e at pos, in other words, if $\rho[pos, match]^R \in \mathcal{L}(e)$. Such a formula can be easily constructed using standard techniques for translating automata to MSO formulas.

For a regular expression $e \in E$ and two positions pos and $match$, the following formula $\text{minmatch}_e(pos, match)$ is true iff the position $match$ is a minimal match for e at pos.

$$\text{minmatch}_e(pos, match) \equiv \text{matches}_e(pos, match) \wedge \\ \forall x((match < x \leq pos) \implies \neg(\text{matches}_e(pos, x)))$$

Given a regular expression $e \in E$, and given a position $pos \in \mathbb{N}$, we have the following formula $\text{rel}_e(pos)$ that is true iff e is relevant at pos.

$$\text{rel}_e(pos) \equiv \exists\, match(\text{minmatch}_e(pos, match))$$

We now define formula ψ_{rel}.

$$\psi_{rel} = \psi_1 \wedge \psi_2 \wedge \psi_3 \wedge \psi_4 \wedge \psi_5$$

where ψ_1 denotes that every position contains exactly one letter.

$$\psi_1 \equiv \forall x(\bigvee_{a \in \Sigma} (x \in P_a) \wedge \bigwedge_{a_i, a_j \in \Sigma, i \neq j} \neg(x \in P_{a_i} \wedge x \in P_{a_j}))$$

The formula ψ_2 says that given a pair of terms t_1, t_2 at a position x, if the predicate $P_{t_1 = t_2}$ holds at x, then the predicate $P_{t_2 = t_1}$ must also hold at x.

$$\psi_2 \equiv \forall x \bigwedge_{e_1, e_2 \in E} \bigwedge_{v_i, v_j \in V} ((x \in P_{\textcircled{e_1}^{-1}v_j = \textcircled{e_2}^{-1}v_k}) \Leftrightarrow (x \in P_{\textcircled{e_2}^{-1}v_k = \textcircled{e_1}^{-1}v_j}))$$

The formula ψ_3 says that given a pair of terms t_1, t_2 at a position x, at most one predicate among $P_{t_1 < t_2}$, $P_{t_1 = t_2}$ and $P_{t_2 < t_1}$ comparing this pair of terms holds at x.

$$\psi_3 \equiv \forall x \bigwedge_{e_1, e_2 \in E} \bigwedge_{v_i, v_j \in V} ((\neg(x \in P_{\textcircled{e_1}^{-1}v_j < \textcircled{e_2}^{-1}v_k} \wedge x \in P_{\textcircled{e_1}^{-1}v_j = \textcircled{e_2}^{-1}v_k})) \\ \wedge (\neg(x \in P_{\textcircled{e_2}^{-1}v_k < \textcircled{e_1}^{-1}v_j} \wedge x \in P_{\textcircled{e_1}^{-1}v_j < \textcircled{e_2}^{-1}v_k})))$$

The formula ψ_4 says that if e_1 and e_2 are relevant expressions at a position x, then the memoir at x must compare every term $\textcircled{e_1}^{-1}v_i$ with every term $\textcircled{e_2}^{-1}v_j$.

$$\psi_4 \equiv \forall x(\bigwedge_{e_1, e_2 \in E} (\ (\text{rel}_{e_1}(x) \wedge \text{rel}_{e_2}(x)) \implies \\ \bigwedge_{v_i, v_j \in V} ((x \in P_{\textcircled{e_1}^{-1}v_i \leq \textcircled{e_2}^{-1}v_j}) \vee (x \in P_{\textcircled{e_2}^{-1}v_j \leq \textcircled{e_1}^{-1}v_i}))))$$

The formula ψ_5 says that at every position x, the memoir at x compares only terms that are relevant at x.

$$\psi_5 \equiv \forall x(\bigwedge_{e_1,e_2 \in E} (\bigvee_{v_i,v_j \in V} ((x \in P_{\textcircled{e_1}^{-1}v_i \leq \textcircled{e_2}^{-1}v_j}) \vee (x \in P_{\textcircled{e_2}^{-1}v_j \leq \textcircled{e_1}^{-1}v_i})) \implies$$
$$(\mathrm{rel}_{e_1}(x) \wedge \mathrm{rel}_{e_2}(x))))$$

It is now easy to verify that the formula ψ_{rel} is satisfied by exactly those symbolic models, all of whose memoirs are relevant.

We now define a few MSO formulas that will aid us in constructing the formula $\psi_{strcycle}$.

Given $\sim \in \{\leq, =\}$, the formula $\sim_{i,j}$ (x,y) is true iff there is a $\sim$-labelled edge from (v_i, x) to (v_j, y) in G_ρ. It is defined as:

$$\sim_{i,j}(x,y) \equiv \exists z(\ (z \geq x) \wedge (z \geq y) \ \wedge$$
$$\bigvee_{e_1,e_2 \in E} z \in P_{\textcircled{e_1}^{-1}v_i \sim \textcircled{e_2}^{-1}v_j} \wedge \ \mathrm{minmatch}_{e_1}(z,x) \wedge \ \mathrm{minmatch}_{e_2}(z,y))$$

Let $l = |V|$. Given $\sim \in \{\leq, =\}$, we define the following formulas:

$$\mathrm{closed}_k^{\sim}(y, Y_1, \ldots, Y_l) \equiv \bigwedge_{p,p' \in \{1,\ldots,l\}} \forall z \forall z'((z \in Y_p \wedge \sim_{p,p'} (z,z')) \implies z' \in Y_{p'}) \ \wedge$$
$$\forall z(\bigwedge_{k' \in \{1,\ldots,l\}} (\sim_{k,k'} (y,z) \implies z \in Y_{k'}))$$

$$\mathrm{path}_{i,j}^{\sim}(x,y) \equiv \forall Y_1 \ldots \forall Y_l(\mathrm{closed}_i^{\sim}(x, Y_1, \ldots, Y_l) \implies y \in Y_j)$$

The intention of the formula $\mathrm{closed}_k^{\sim}(y, Y_1, \ldots, Y_l)$ is the following: if there is a path starting from variable v_k at position y to some variable v_j at some position z, then z must belong to Y_j. This can be proved by an induction on the length of the path in addition to Knaster-Tarski theorem over the lattice of l-tuples of subsets of $\mathbb{N}$.

We claim that the formula $\mathrm{path}_{i,j}^{\sim}(x,y)$ holds iff there is a directed path of non-zero length from (v_i, x) to (v_j, y) in G_ρ, all of whose edges are labelled with $\sim$. The details of the proof can be found in section B of the appendix.

We define a new formula $\mathrm{path}_{i,j}^{<}(x,y)$ which holds iff there is a strict directed path from (v_i, x) to (v_j, y).

$$\mathrm{path}_{i,j}^{<}(x,y) \equiv \mathrm{path}_{i,j}^{\leq}(x,y) \wedge \neg \mathrm{path}_{i,j}^{=}(x,y)$$

We now define the MSO formula $\psi_{strcycle}$.

$$\psi_{strcycle} \equiv \neg(\mathrm{path}_{i,i}^{<}(x,x))$$

Thus, we have a formula $\psi_{symb} \equiv \psi_{rel} \wedge \psi_{strcycle}$ whose language is the set of all consistent symbolic models. Hence, $\mathcal{L}_{symb} = \mathcal{L}(\psi_{symb})$. Now, by Büchi's theorem [5], we know that we can construct a Büchi automaton $\mathcal{A}_{symb}$ recognizing $\mathcal{L}_{symb}$, or in other words, $\mathcal{L}_{symb}$ is ω-regular. ◀

▶ **Lemma 7.** *Let $\mathcal{D} = (D, <, =)$ be a constraint system such that D is dense and open. Let ϕ be an RCLTL($\mathcal{D}$) formula. Then, the language $\mathcal{L} = \{\mu(\sigma) \mid \sigma \models \phi\}$ is ω-regular.*

Proof. Lemma 5 and Corollary 4 imply that $\mathcal{L} = \mathcal{L}_{symb} \cap \mathcal{L}_{sat}^{\phi}$ where $\mathcal{L}_{sat}^{\phi}$ is equal to the set of symbolic models that symbolically satisfy ϕ. A straightforward adaptation of standard automata-theoretic construction for propositional LTL [21] proves that $\mathcal{L}_{sat}^{\phi}$ is ω-regular. Also, we know that $\mathcal{L}_{symb}$ is ω-regular, as shown before. Since ω-regular languages are closed under intersection, $\mathcal{L}$ is also ω-regular. ◀

▶ **Theorem 8.** *The satisfiability problem for RCLTL($\mathcal{D}$) where $\mathcal{D} = (D, <, =)$ and D is a dense and open domain, is decidable*

Proof. Solving the satisfiability problem for RCLTL($\mathcal{D}$) amounts to checking if, given an RCLTL($\mathcal{D}$) formula ϕ, $\mathcal{L} = \{\mu(\sigma) \mid \sigma \models \phi\} \neq \emptyset$. By Lemma 7, we know $\mathcal{L}$ is ω-regular. Hence, it is possible to construct a Büchi automaton accepting $\mathcal{L}$ and check for its non-emptiness. ◀

5 Satisfiability over $(\mathbb{Z}, <, =)$

In this section we consider the constraint system $\mathcal{Z} = (\mathbb{Z}, <, =)$. Since $\mathbb{Z}$ is not dense, Lemma 5 does not apply here. We solve the satisfiability problem for RCLTL over $(\mathbb{Z}, <, =)$ by using an automata-theoretic technique, similar to the one used in [12] to solve the satisfiability problem for CLTL over $(\mathbb{Z}, <, =)$. The idea is to define a Büchi automaton which accepts a superset of the set of all symbolic models which admit a concrete model with the property that all ultimately periodic words recognized by this automaton admit concrete models.

We begin with a characterization of symbolic models that admit a concrete model over $\mathbb{Z}$. Let ρ be a consistent symbolic model. For a directed path p in G_ρ, let $slen(p)$ denote the strict length of p, i.e. the number of strict edges in p if this number is finite, and ω otherwise. For any two vertices $u, v \in G_\rho$, define $slen(u, v)$ to be the supremum of $slen(p)$ over directed paths p from u to v, if it exists, and ω otherwise. If there is no direct path from u to v, $slen(u, v)$ is 0.

▶ **Lemma 9** ([12, Lemma 6.1]). *A consistent symbolic model ρ admits a concrete model over $\mathbb{Z}$ iff for all $u, v \in G_\rho$, $slen(u, v) < \omega$.*

Next we would like to design finite state automata to detect whether there are vertices u, v with $slen(u, v) = \omega$. Finite state automata cannot remember paths of unbounded lengths, so we look for bounded paths which can be repeated arbitrarily many times to get unbounded *slen*. The central idea in [12] is to show that such repeatable bounded paths are guaranteed to exist in ultimately periodic sequences. We borrow that idea here; but to implement that idea here requires much work. We now introduce some notation, conditions and results to formalize this.

Given a symbolic model ρ, and given a vertex (x, i) of G_ρ, we define the "level" of vertex (x, i) to be i and the "var" of vertex (x, i) to be x. A forward $u - v$ (resp. backward $u - v$) witness in G_ρ is a sequence $\nu \colon \mathbb{N} \to V \times \mathbb{N}$ such that $\nu(0) = u$ (resp. v), for all $i \geq 0$ there is a strict path from $\nu(i)$ to $\nu(i+1)$ (resp. from $\nu(i+1)$ to $\nu(i)$), $\text{level}(\nu(i+1)) > \text{level}(\nu(i))$ and for infinitely many i, there is a directed path from $\nu(i)$ to v (resp. from u to $\nu(i)$). We now give a condition $C_{\mathbb{Z}}$ (similar to condition $C_{\mathbb{Z}}$ in [12]) on a symbolic model ρ.

$(C_{\mathbb{Z}})$: There do not exist vertices $u, v \in G_\rho$ with either a forward or backward $u - v$ witness.

If ρ does not satisfy $C_{\mathbb{Z}}$, there exist vertices u and v with either a forward or a backward $u - v$ witness implying that $slen(u, v) = \omega$. Figure 1 illustrates a forward $u - v$ witness. Hence, lemma 9 implies that ρ does not admit a concrete model over $\mathbb{Z}$. We shall now see

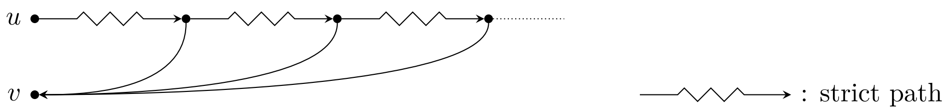

Figure 1 A forward $u - v$ witness.

that if ρ is an ultimately periodic word, then $C_{\mathbb{Z}}$ is both necessary and sufficient. We say a symbilic model is ultimately periodic if it is of the form $\tau \cdot \delta^{\omega}$ for some finite words τ and δ over $\Sigma \times M$.

We use the following terminologies with respect to an ultimately periodic consistent symbolic model $\rho = \tau \cdot \delta^{\omega}$. A vertex in G_ρ is said to be a τ-vertex if its level is at most $|\tau|$. The suffixes $\rho[i, \infty)$ and $\rho[i + m|\delta|, \infty)$ are equal for every $m \in \mathbb{N}$ and $i > |\tau|$. Accordingly, we say the pair of vertices (x, i) and $(x, i + m|\delta|)$ are isomorphic for all $i > |\tau|$ and all $m \in \mathbb{N}$ and denote it by $(x, i) \cong (x, i + m|\delta|)$. With slight abuse of notation, we also say that the positions i and $i + m|\delta|$ are isomorphic. The set of isomorphism classes $\cong_0, \cong_1, \cdots, \cong_{|\delta|-1}$ is a partition of $[|\tau| + 1, \infty)$ such that for all $m \in \mathbb{N}$ and for all $j \in [0, |\delta| - 1]$, any position of the form $|\tau| + m|\delta| + j$ is in the isomorphism class $\cong_j$. A non-τ-vertex v in G_ρ is said to be in the isomorphism class $\cong_j$ if $\text{level}(v) \in \cong_j$. An edge from or to a τ-vertex is said to be a long jump. An edge between non-τ-vertices is said to be a hop. It is a strict hop if it is labelled by $<$. A path is said to be a hopping path if it does not contain any long jumps. A forward hopping map of n traversals is a sequence $\mu_{fh} : [0, n] \to V \times \mathbb{N}$ such that for every $i \in [0, n - 1]$, there is a hopping path from $\mu_{fh}(i)$ to $\mu_{fh}(i + 1)$ and $\text{level}(\mu_{fh}(i + 1)) > \text{level}(\mu_{fh}(i))$. The rank of a forward hopping map μ_{fh} is the number of indices i such that there is a strict path from $\mu_{fh}(i)$ to $\mu_{fh}(i + 1)$. The existence of one strict path from $\mu_{fh}(i)$ to $\mu_{fh}(i + 1)$ ensures that all paths from $\mu_{fh}(i)$ to $\mu_{fh}(i + 1)$ are strict – if there was a strict path and a non-strict path from $\mu_{fh}(i)$ to $\mu_{fh}(i + 1)$, combining them would result in a strict cycle from $\mu_{fh}(i)$ to itself, contradicting the consistency of G_ρ. From a forward hopping map μ_{fh} of n traversals, we can obtain a path by concatenating a sequence of paths $\pi_1, \ldots \pi_n$, where every π_i is a hopping path from $\mu_{fh}(i - 1)$ to $\mu_{fh}(i)$. We call paths constructed this way forward hopping paths. The rank of a forward hopping path π_{fh} is the maximum over the ranks of all forward hopping maps from which π_{fh} can be constructed. Similarly we have the notions of backward hopping maps μ_{bh} and backward hopping paths π_{bh}. Given two vertices u, v in G_ρ and a path p from u to v, we define the furthermost vertex $\text{fmv}(p)$ to be the vertex w along p such that for all vertices x in p, $\text{level}(x) \leq \text{level}(w)$ (if there are multiple such occurrences of w, choose the last such occurrence along p). The portion of the path p from u to $\text{fmv}(p)$ is said to be the forward portion and the rest is said to be the backward portion.

Given any regular expression $e \in E$, let A_e denote the minimal DFA that recognizes $\mathcal{L}(e)$ and let n_e denote the number of states in A_e. Now, given a symbolic model ρ and a pair of positions i, j with $j \leq i$, checking if j is a minimal match for e at i is equivalent to checking if the run of A_e on $\rho[j, i]^R$ is accepting and that A_e does not accept any strict prefix of $\rho[j, i]^R$.

The following claim states that a hop cannot be too long. Intuitively, a hop is between two non-τ vertices. The sequence between these two vertices consists of copies of δ and is a minimal match for some regular expression. If there are too many copies of δ, some copies can be removed to get a shorter match for the regular expression.

▷ Claim 10. Suppose $\rho = \tau \cdot \delta^{\omega}$ is an ultimately periodic consistent symbolic model and $e \in E$ is a regular expression. For all positions $i > |\tau|$, if there exists a match j for e at i such that $j > |\tau|$, then $i - \text{mmp}(i, e) \leq 2n_e|\delta|$.

Non-strict directed cycles in a path do not contribute to its *slen*. So, henceforth when we refer to a path, we mean path without non-strict cycles.

▷ Claim 11. Suppose $\rho = \tau \cdot \delta^\omega$ is an ultimately periodic consistent symbolic model and there are vertices u, v with $slen(u, v) = \omega$. For any $n \in \mathbb{N}$, there is a path from u to v containing either a forward hopping path in its forward portion starting at a vertex whose level is greater than $\text{level}(u)$ and $\text{level}(v)$, or a backward hopping path in its backward portion starting at a vertex whose level is greater than $\text{level}(u)$ and $\text{level}(v)$, with the rank of the forward/backward hopping path being at least n.

Proof. The number of long jumps along any $u - v$ path is $\leq 2|V||\tau|$, because every τ-vertex can be visited at most once along a path (since G_ρ has no strict cycles). If the condition in the claim is not satisfied, then there is a position beyond which strict paths cannot go, contradicting the hypothesis that $slen(u, v) = \omega$. ◁

▶ **Definition 12.** *For an ultimately periodic consistent symbolic model $\rho = \tau \cdot \delta^\omega$, let arl (intended to be short for automata run repeat length) be the number $n!|\delta|$, where $n = \max_{e \in E} n_e$.*

Claim 10 proves that hops are short. The next claim proves that long jumps to non-τ vertices can be unboundedly long. Intuitively, such a long jump includes many copies of δ. Some copy may be repeated many times to make the jump longer.

▷ Claim 13. Suppose $\rho = \tau \cdot \delta^\omega$ is an ultimately periodic consistent symbolic model and G_ρ has vertices w_1, w_2 such that $w_1 \cong w_2$, $\text{level}(w_1) \geq |\tau| + arl$ and $\text{level}(w_2) - \text{level}(w_1)$ is a multiple of arl. If there is a long jump from w_1 to a τ-vertex y, then there is also a long jump from w_2 to y. Similarly, if there is a long jump from some τ-vertex z to w_2 then there is also a long jump from z to w_1.

Proof. We prove the first case; the proof of the second case is symmetric. Also, we prove for the case where $\text{level}(w_2) - \text{level}(w_1) = arl$. If $\text{level}(w_2) - \text{level}(w_1)$ is any multiple of arl, we can choose intermediary vertices between w_1 and w_2 with the difference of levels between every pair of consecutive vertices as arl and applying the result to every pair of intermediary vertices gives us the required result for w_1 and w_2.

Since there is an edge from w_1 to y, there is a position $k_1 > \max(\text{level}(w_1), \text{level}(y))$ and regular expressions $e_1, e_2 \in E$ such that $\text{mmp}(k_1, e_1) = \text{level}(w_1)$ and $\text{mmp}(k_1, e_2) = \text{level}(y)$.

Consider the position $k_2 = k_1 + arl$. Positions k_1 and k_2 are isomorphic. Therefore, $\text{mmp}(k_2, e_1) = \text{level}(w_2)$. We know $\textcircled{$e_1$}^{-1}\text{var}(w_1) \sim_{k_1} \textcircled{e_2}^{-1}\text{var}(y)$, where $\sim \in \{=, <\}$. Also $\text{var}(w_1) = \text{var}(w_2)$ and $\rho(k_1) = \rho(k_2)$. This gives us $\textcircled{$e_1$}^{-1}\text{var}(w_2) \sim_{k_2} \textcircled{e_2}^{-1}\text{var}(y)$. So, in order to prove that there is an edge from w_2 to y, it is sufficient to prove that $\text{mmp}(k_2, e_2) = \text{level}(y)$.

Now consider a run of the DFA A_{e_2} on the word $\rho[0, k_2]^R$. There must exist a state s which is visited at at least 2 isomorphic positions along the run. Consider the first such state s visited along the run, let j_1 and j_2 be the first two isomorphic positions at which s is visited. In between positions j_1 and j_2, every state of the automaton could have been visited at most $|\delta|$ times. Therefore, $j_1 - j_2 = m|\delta|$ where $m \leq n_{e_2}$.

By Claim 10, either $\text{mmp}(k_2, e_2) \leq |\tau|$ or $\text{mmp}(k_2, e_2) \geq k_2 - arl$. If $\text{mmp}(k_2, e_2) \geq k_2 - arl$, then due to isomorphism, $\text{mmp}(k_1, e_2) \geq k_1 - arl$, which means that $\text{mmp}(k_1, e_2) > |\tau|$. But this contradicts our assumption that $\text{mmp}(k_1, e_2) = \text{level}(y)$ as y is a τ-vertex. Therefore, $\text{mmp}(k_2, e_2) \leq |\tau|$.

Let $j_1' = j_1 - arl$. Consider the run of A_{e_2} on the word $\rho[0, k_1]^R$. This run visits state s at position j_1' (this is because, we start from the initial state at k_1 and visit the same sequence of states that the run of the word $\rho[0, k_2]^R$ visited between k_2 and j_1).

Along the run of $\rho[0, k_2]^R$, the sequence of states visited between j_1 and j_2 keeps repeating itself after position j_2 (till the first non-τ-vertex) and thus must eventually visit the state s at position j_1'. This is because, $j_1 - j_1' = n!|\delta|$ which is a multiple of $j_1 - j_2 = m|\delta|$ (since $m \leq n_{e_2} \leq n$, m divides $n!$).

We now claim that the first final state visited by the run of $\rho[0, k_2]^R$ must be at level(y). This is because the sequence of states visited beyond position j_1' along the runs of the words $\rho[0, k_1]^R$ and $\rho[0, k_2]^R$ is the same and therefore, we visit a final state at level(y) and a visit to a final state before position level(y) would indicate a shorter minimal match for e_2 at position k_1 contradicting our assumption. Therefore, there is a long jump from w_2 to y. ◁

Next we prove that satisfying $C_{\mathbb{Z}}$ is necessary and sufficient for ultimately periodic consistent symbolic models to admit concrete models. Compared to [12, Lemma 6.2], this proof is more involved due to the presence of long jumps, which are not there in [12].

▶ **Lemma 14** ([12, Lemma 6.2]). *Let $\rho = \tau \cdot \delta^\omega$ be an ultimately periodic consistent symbolic model. Then ρ admits a concrete model over $\mathbb{Z}$ iff ρ satisfies the condition $C_{\mathbb{Z}}$.*

Proof. ($\Rightarrow$) If ρ admits a concrete model over $\mathbb{Z}$ then for all $u, v \in G_\rho$, $slen(u, v) < \omega$. Therefore, condition $C_{\mathbb{Z}}$ is clearly satisfied.

($\Leftarrow$) Let there exist vertices $u, v \in G_\rho$ such that $slen(u, v) = \omega$. Let $l = |V|$. By Claim 11, we can find a path p from u to v such that there is a forward hopping path π_{fh} of rank at least $(2l^2|\delta|^2 + 3) \cdot arl$ starting at a vertex whose level is greater than level(u) and level(v) in the forward portion of p, or a similar backward hopping path. We consider the first case and prove that Condition $C_{\mathbb{Z}}$ is violated, leading to a contradiction. The argument for the second case is symmetric.

Let μ_{fh} be a forward hopping map of rank $(2l^2|\delta|^2 + 3) \cdot arl$ from which π_{fh} can be constructed. Let strict vertices of μ_{fh} be those vertices $\mu_{fh}(i)$ such that the hopping path from $\mu_{fh}(i)$ to $\mu_{fh}(i+1)$ (along π_{fh}) is a strict hopping path. One level may contain at most one strict vertex of μ_{fh}. So, μ_{fh} must contain strict vertices in at least $(2l^2|\delta|^2 + 3) \cdot arl$ distinct levels. Suppose i_{min} and i_{max} are the minimum and maximum among these levels. Take the subgraph of G_ρ induced by vertices in levels between i_{min} and i_{max} and divide it into $\lfloor \frac{i_{max} - i_{min}}{arl} \rfloor$ portions containing arl levels each and a last portion containing the remaining levels ($< arl$). At least $(2l^2|\delta|^2 + 3)$ portions contain strict vertices of μ_{fh}.

We look at the forward hopping path after the first portion. Now we drop alternate portions and from each of the remaining $(l^2|\delta|^2 + 1)$ portions, take one strict vertex so that the difference in levels between picked vertices is at least arl. Among these vertices, we can find a set of at least $l|\delta| + 1$ vertices, all belonging to the same isomorphism class (since there are $l|\delta|$ isomorphism classes). Let the vertices in this set be $w_1, w_2, \cdots w_{l|\delta|+1}$ (in increasing order of their levels). Note that there is a strict path from w_i to w_{i+1} for all $i \in [1, l|\delta|]$. Let us now look at the portion of the path from $w_{l|\delta|+1}$ to v. There can be two possible cases:

Case a. The portion of the path from $w_{l|\delta|+1}$ to the first vertex occurring after $w_{l|\delta|+1}$ along the path whose level is less than level(w_1), is a hopping path.

We know by Claim 10 that the difference in levels between the end-points of a hop must be less than arl. Hence, along this hopping path, a set of vertices $w_1', w_2' \cdots w_{l|\delta|+1}'$ must be visited such that $|\text{level}(w_i) - \text{level}(w_i')| < arl$ for all $i \in [1, l|\delta| + 1]$. Among these $l|\delta| + 1$ vertices, at least two must be isomorphic. Let us say (w_i', w_j') is such a pair of isomorphic vertices along this hopping path with $i < j$ (see the picture on the left side of Fig. 2).

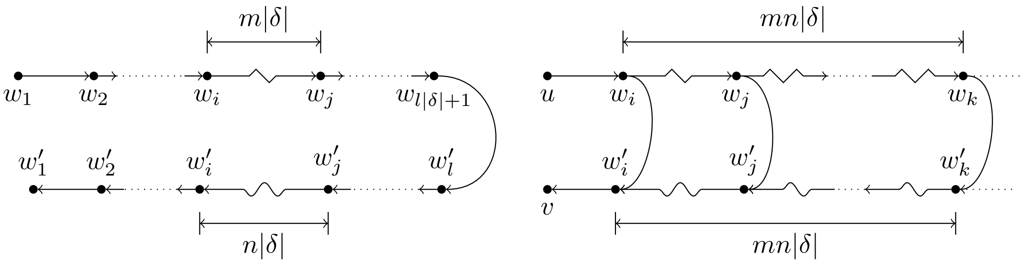

Figure 2 Illustration for **case a** of the proof of Lemma 14.

Now, let $|\text{level}(w_j)-\text{ level}(w_i)| = m|\delta|$ and $|\text{level}(w'_j)-\text{ level}(w'_i)| = n|\delta|$ for some $m, n \in \mathbb{N}$. Note that $|\text{level}(w_i)-\text{ level}(w'_i)|$ and $|\text{level}(w_j)-\text{ level}(w'_j)|$ need not be equal. But since $w'_i \cong w'_j$ and $w_i \cong w_j$, it is possible to pump the portion of the path from w'_i to w'_j m times and the portion of the path from w_i to w_j n times, resulting in a forward path from u to some vertex w_k passing through w_i and in a backward path from v to some vertex w'_k passing through w'_i such that $w'_i \cong w'_k$, $w_i \cong w_k$ and $|\text{level}(w_i)-\text{ level}(w'_i)| = |\text{level}(w_k)-\text{ level}(w'_k)|$. This is illustrated in the picture on the right side of Fig. 2. Since there is a directed path from w_i to w'_i that does not visit any τ-vertex, there must also be a directed path from w_k to w'_k (a path isomorphic to that from w_i to w'_i). Note that the portion of the path from w_i to w_k along p is strict. It is therefore possible to pump the portion of the forward and backward paths from w_i to w_k and respectively from w'_i to w'_k infinitely often, to obtain a forward $u - v$ witness. This results in a violation of condition $C_{\mathbb{Z}}$, giving rise to a contradiction.

Case b. In the portion of the path from $w_{l|\delta|+1}$ to v, before a vertex with level less than level(w_1) is visited, there is a long jump.

Let the first long jump along the portion of the path from $w_{l|\delta|+1}$ to v be from a vertex z to a τ-vertex y. Figure 3 illustrates this. Consider a vertex $w_{l|\delta|+2}$ isomorphic to $w_{l|\delta|+1}$ such that $(\text{level}(w_{l|\delta|+2})-\text{ level}(w_{l|\delta|+1})) = (\text{level}(w_{l|\delta|+1})-\text{ level}(w_{l|\delta|})) \cdot \mathit{arl}$. Since $(\text{level}(w_{l|\delta|+2})-\text{ level}(w_{l|\delta|+1}))$ is a multiple of $(\text{level}(w_{l|\delta|+1})-\text{ level}(w_{l|\delta|}))$ and since $w_{l|\delta|+1} \cong w_{l|\delta|+2}$, there must be a strict path from $w_{l|\delta|+1}$ to $w_{l|\delta|+2}$ as well. We now "mimic" the path from $w_{l|\delta|+1}$ to z by taking an isomorphic copy of that path from $w_{l|\delta|+2}$. We have $(\text{level}(w_{l|\delta|+2})-\text{ level}(w_{l|\delta|+1})) = (\text{level}(w_{l|\delta|+1})-\text{ level}(w_{l|\delta|})) \cdot \mathit{arl}$ and the portion of the path from $w_{l|\delta|+1}$ to z consists only of hops. So we basically construct a path of hops from $w_{l|\delta|+2}$ such that the level of every vertex along that path is exactly $(\text{level}(w_{l|\delta|+1})-\text{ level}(w_{l|\delta|})) \cdot \mathit{arl}$ more than the level of the corresponding vertex in the path from $w_{l|\delta|+1}$ to z. This means that corresponding to position z is the position z' in this new path such that $(\text{level}(z')-\text{ level}(z)) = (\text{level}(w_{l|\delta|+1})-\text{ level}(w_{l|\delta|})) \cdot \mathit{arl}$. By construction, we know that level$(z) > |\tau| + \mathit{arl}$. Therefore

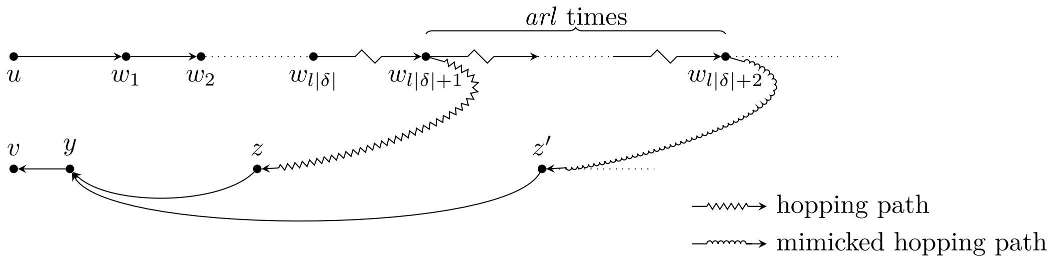

Figure 3 Illustration for **case b** of the proof of Lemma 14.

by Claim 13, there exists a long jump from z' to y, hence there exists a path from $w_{l|\delta|+2}$ to v. We can now repeat this process by considering a vertex $w_{l|\delta|+3}$ which is isomorphic to $w_{l|\delta|+2}$ and whose level is exactly (level$(w_{l|\delta|+1})-$ level$(w_{l|\delta|})) \cdot arl$ more than that of $w_{l|\delta|+2}$. Now, we mimic the path from $w_{l|\delta|+2}$ to z' as before, to get a path from $w_{l|\delta|+3}$ to v. The paths from $w_{l|\delta|+i}$ to $w_{l|\delta|+i+1}$ is strict for all $i \in \mathbb{N}$ by construction. Thus, we repeat this process infinitely many times to get a forward $u - v$ witness with the sequence of vertices $w_{l|\delta|+1}, w_{l|\delta|+2}, w_{l|\delta|+3} \cdots$ being the infinitely many vertices along the witness such that there is a directed path from each of them to v. This again leads to a violation of condition $C_{\mathbb{Z}}$, contradicting our assumption.

In the case where there is a backward hopping path in the backward portion of p, a symmetric argument gives us a backward $u - v$ witness, violating condition $C_{\mathbb{Z}}$. ◀

We shall now define an MSO formula $\psi_{C_{\mathbb{Z}}}$ whose language is the set of all symbolic models that satisfy condition $C_{\mathbb{Z}}$. The formula $succ(x, y, Z)$ holds iff the position y is a successor of position x when restricted to the set Z.

$$\text{succ}(x, y, Z) \equiv ((x < y) \wedge (x \in Z) \wedge (y \in Z) \wedge \forall z((x < z < y) \implies \neg(z \in Z)))$$

The formula $\inf(Z)$ holds iff there are infinitely many positions in Z.

$$\inf(Z) \equiv \forall y \exists z((z > y) \wedge (z \in Z))$$

The formula $\text{sfp}(Z, Z_1, \ldots, Z_l)$ (strict forward path) holds iff $Z = \bigcup_{i=1}^{l} Z_i$ and between every pair of consecutive positions x, y in Z, if $x \in Z_p$ and $y \in Z_{p'}$, then there is a strict directed path from the vertex (v_p, x) to $(v_{p'}, y)$ in G_ρ. The formula $\text{sbp}(Z, Z_1, \ldots, Z_l)$ (strict backward path) is symmetric.

$$\begin{aligned}\text{sfp}(Z, Z_1, \ldots, Z_l) \equiv & \forall z(\bigvee_{i=1}^{l}(z \in Z_i) \Leftrightarrow z \in Z) \wedge \\ & \forall x \forall y(\bigwedge_{p,p' \in [1,l]} ((x \in X_p \wedge y \in X_{p'} \wedge succ(x, y, Z)) \implies \text{path}^{<}_{p,p'}(x, y))\end{aligned}$$

$$\begin{aligned}\text{sbp}(Z, Z_1, \ldots, Z_l) \equiv & \forall z(\bigvee_{i=1}^{l}(z \in Z_i) \Leftrightarrow z \in Z) \wedge \\ & \forall x \forall y(\bigwedge_{p,p' \in [1,l]} ((x \in X_p \wedge y \in X_{p'} \wedge succ(x, y, Z)) \implies \text{path}^{<}_{p',p}(y, x))\end{aligned}$$

The formula $\psi^a_{C_{\mathbb{Z}}}$ (resp. $\psi^b_{C_{\mathbb{Z}}}$) denotes that there do not exist vertices (v_i, u) and (v_j, v) in G_ρ such that there is a forward (resp. backward) $(v_i, u) - (v_j, v)$ witness.

$$\begin{aligned}\psi^a_{C_{\mathbb{Z}}} \equiv \neg(\exists u \exists v \bigvee_{i,j \in [1,l]} & (\exists X \exists X_1 \ldots \exists X_l \wedge ((\text{sfp}(X, X_1, \ldots, X_l) \wedge \inf(\text{X}) \wedge (u \in X_i) \\ & \wedge \forall y(\bigwedge_{k \in [1,l]} ((y \in X_k) \implies \text{path}^{\leq}_{k,j}(y, v))))\end{aligned}$$

$$\begin{aligned}\psi^b_{C_{\mathbb{Z}}} \equiv \neg(\exists u \exists v \bigvee_{i,j \in [1,l]} & (\exists X \exists X_1 \ldots \exists X_l((\text{sbp}(X, X_1, \ldots, X_l) \wedge \inf(\text{X}) \wedge (v \in X_j) \\ & \wedge \forall y(\bigwedge_{k \in [1,l]} ((y \in X_k) \implies \text{path}^{\leq}_{i,k}(u, y))))\end{aligned}$$

Now we define the formula $\psi_{C_\mathbb{Z}}$ as $\psi_{C_\mathbb{Z}} \equiv \psi^a_{C_\mathbb{Z}} \vee \psi^b_{C_\mathbb{Z}}$. It is easy to verify that $\psi_{C_\mathbb{Z}}$ is true iff the word satisfies condition $C_\mathbb{Z}$.

As discussed earlier, the language $\mathcal{L}^{sat}_\phi$ (described in Section 4) is ω-regular, hence by Büchi's theorem, we can define an MSO formula ψ^{sat}_ϕ whose language is $\mathcal{L}^{sat}_\phi$. ψ_{symb}, which we have defined in the previous section, is the MSO formula whose language is $\mathcal{L}_{symb}$. Now let ϕ be the given RCLTL($\mathcal{Z}$) formula. We define the MSO formula $\psi^{\mathcal{Z}}_\phi = \psi^{sat}_\phi \wedge \psi_{symb} \wedge \psi_{C_\mathbb{Z}}$. We now have the following lemma:

▶ **Lemma 15.** *An RCLTL($\mathcal{Z}$) formula ϕ is satisfiable iff $\mathcal{L}(\psi^{\mathcal{Z}}_\phi)$ is non-empty.*

Proof. ($\Rightarrow$) Suppose ϕ is satisfiable. Let σ be a concrete model such that $\sigma \models \phi$. Let $\rho = \mu(\sigma)$. By Lemma 2, $\rho \in \mathcal{L}(\psi^{sat}_\phi)$. By definition, ρ is a consistent symbolic model, hence $\rho \in \mathcal{L}(\psi_{symb})$. Also since ρ admits a concrete model over $\mathbb{Z}$, by Lemma 9, ρ satisfies condition $C_\mathbb{Z}$, hence $\rho \in \mathcal{L}(\psi_{C_\mathbb{Z}})$. Thus $\rho \in \mathcal{L}(\psi^{sat}_\phi) \cap \mathcal{L}(\psi_{symb}) \cap \mathcal{L}(\psi_{C_\mathbb{Z}})$. Hence $\rho \in \mathcal{L}(\psi^{\mathcal{Z}}_\phi)$.

($\Leftarrow$) Suppose a word ρ satisfies the formula $\psi^{\mathcal{Z}}_\phi$. Then, there must exist an ultimately periodic word ρ' such that ρ' satisfies $\psi^{\mathcal{Z}}_\phi$. Since $\rho' \in \mathcal{L}(\psi^{\mathcal{Z}}_\phi)$, ρ' is a symbolic model that symbolically satisfies ϕ and also satisfies condition $C_\mathbb{Z}$. By Lemma 14, ρ' admits a concrete model σ over $\mathbb{Z}$. Also, since $\rho' \models_s \phi$, by Lemma 2, $\sigma \models \phi$. Thus, ϕ is RCLTL($\mathcal{Z}$)-satisfiable. ◀

▶ **Theorem 16.** *The satisfiability problem for RCLTL($\mathcal{Z}$) is decidable.*

Proof. Given an RCLTL($\mathcal{Z}$) formula ϕ, we can effectively convert the MSO formula to a Büchi automaton $\mathcal{A}^{\mathcal{Z}}_\phi$ using the construction of Büchi's theorem [5] and check it for non-emptiness. ◀

6 Discussion and Future Work

In this paper, we consider an extension RCLTL of Constraint LTL to let atomic formulas access positions that are unboundedly far away in the past. We prove that the satisfiability problem for RCLTL is decidable over dense and open domains, and also over the integers with linear order and equality. With some changes in the technical details of the proofs, we think that the satisfiability problem can be proven to be decidable if we allow the atomic formulas to access positions that are unboundedly far away in the future as well, and also over the domain of natural numbers.

Based on the quantifier alternations in the MSO formulas that we have used, a rough complexity upper bound of six exponentials may be worked out. Figuring out the exact complexity of the satisfiability problem is left for future work.

References

1 Marcello M Bersani, Domenico Bianculli, Schahram Dustdar, Alessio Gambi, Carlo Ghezzi, and Srđan Krstić. Towards the formalization of properties of cloud-based elastic systems. In *proceedings of the 6th international workshop on principles of engineering service-oriented and cloud systems*, pages 38–47, 2014.

2 Marcello M Bersani, Matteo Rossi, and Pierluigi San Pietro. A tool for deciding the satisfiability of continuous-time metric temporal logic. *Acta Informatica*, 53(2):171–206, 2016.

3 Mikołaj Bojańczyk, Claire David, Anca Muscholl, Thomas Schwentick, and Luc Segoufin. Two-variable logic on data words. *ACM Transactions on Computational Logic (TOCL)*, 12(4):1–26, 2011.

4 Ahmed Bouajjani and Peter Habermehl. Constrained properties, semilinear systems, and petri nets. In *International Conference on Concurrency Theory*, pages 481–497. Springer, 1996.

5 J Richard Buchi. On a decision method in restricted second order arithmetic. *Proc. Internat. Congr. on Logic, Methodology and Philosophy of Science, 1960*, 1960.

6 Hubert Comon and Véronique Cortier. Flatness is not a weakness. In *International workshop on computer science logic*, pages 262–276. Springer, 2000.

7 Stephane Demri, Deepak D'Souza, and Régis Gascon. Temporal logics of repeating values. *Journal of Logic and Computation*, 22, October 2012. `doi:10.1093/logcom/exr013`.

8 Stéphane Demri and Régis Gascon. The effects of bounding syntactic resources on presburger ltl. *Journal of Logic and Computation*, 19(6):1541–1575, 2009.

9 Stéphane Demri and Ranko Lazić. Ltl with the freeze quantifier and register automata. *ACM Trans. Comput. Logic*, 10(3), April 2009. `doi:10.1145/1507244.1507246`.

10 Stéphane Demri and Karin Quaas. Concrete domains in logics: a survey. *ACM SIGLOG News*, 8(3):6–29, 2021.

11 Stéphane Demri and Karin Quaas. Constraint automata on infinite data trees: from ctl(z)/ctl*(z) to decision procedures. In *34th International Conference on Concurrency Theory (CONCUR 2023)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. `doi:10.4230/LIPIcs.CONCUR.2023.29`.

12 Stéphane Demri and Deepak D'Souza. An automata-theoretic approach to constraint ltl. *Information and Computation*, 205(3):380–415, 2007. `doi:10.1016/j.ic.2006.09.006`.

13 Cindy Eisner and Dana Fisman. Functional specification of hardware via temporal logic. *Handbook of Model Checking*, pages 795–829, 2018.

14 Bernd Finkbeiner, Philippe Heim, and Noemi Passing. Temporal stream logic modulo theories. In *International Conference on Foundations of Software Science and Computation Structures*, pages 325–346. Springer International Publishing Cham, 2022.

15 Bernd Finkbeiner, Felix Klein, Ruzica Piskac, and Mark Santolucito. Temporal stream logic: Synthesis beyond the bools. In *International Conference on Computer Aided Verification*, pages 609–629. Springer, 2019.

16 Michael Kaminski and Nissim Francez. Finite-memory automata. *Theoretical Computer Science*, 134(2):329–363, 1994.

17 Nadia Labai, Magdalena Ortiz, and Mantas Šimkus. An exptime upper bound for alc with integers. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, volume 17, pages 614–623, 2020.

18 Gabor Madl, Luis Bathen, German Flores, and Divyesh Jadav. Formal verification of smart contracts using interface automata. In *2019 IEEE International Conference on Blockchain (Blockchain)*, pages 556–563, 2019. `doi:10.1109/Blockchain.2019.00081`.

19 M Praveen, Diego Figueira, and Stephane Demri. Reasoning about data repetitions with counter systems. *Logical Methods in Computer Science*, 12, 2016.

20 Alfred Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5(2):285–309, 1955.

21 Moshe Y Vardi and Pierre Wolper. An automata-theoretic approach to automatic program verification. In *Proceedings of the First Symposium on Logic in Computer Science*, pages 322–331. IEEE Computer Society, 1986.

A Details of Section 3

Proof of Lemma 5. We shall prove the lemma by showing that the graph G_ρ admits an edge-respecting labelling $l\colon V \times \mathbb{N} \to D$.

We assume an ordering $\prec$ on V and use it to define an ordering on the vertices of G_ρ given by $(x, i) \prec (y, j)$ iff $i < j$, or $i = j$ and $x \prec y$. We now give a procedure to label the vertices.

1. Label the vertices in order of $\prec$. Begin by labelling the first, say $(x, 0)$ by any element $d \in D$.
2. In general, let X be the portion of the graph already labelled, and u be the next vertex to be labelled. Let $X_{=u}$ denote the set of vertices in X to which there is an $=$-labelled path from u; $X_{<u}$ denote the set of vertices in X from which there is a strict directed path to u; and $X_{u<}$ denote the set of vertices in X to which there is a strict directed path from u. Let $p = \max\{l(v) \mid v \in X_{<u}\}$ and $q = \min\{l(w) \mid w \in X_{u<}\}$ (we will use the values of p and q only when the sets $X_{<u}$ and $X_{u<}$ are, respectively, non-empty). The values p and q are in D, which is dense and open. So there exist $p', q', r \in D$ such that $p < p'$, $q' < q$ and $p < r < q$. Now we can label u as follows:
   - **(a)** if $X_{=u}$ is non-empty, label u by $l(v)$ for any $v \in X_{=u}$
   - **(b)** if $X_{<u}$ is non-empty and both $X_{=u}$ and $X_{u<}$ are empty, label u by p'.
   - **(c)** if $X_{u<}$ is non-empty and both $X_{=u}$ and $X_{<u}$ are empty, label u by q'.
   - **(d)** if $X_{<u}$ and $X_{u<}$ are non-empty and $X_{=u}$ is empty, label u by r.
   - **(e)** if $X_{<u}$, $X_{u<}$ and $X_{=u}$, all three are empty, label u by any arbitrary element d of D.

We now aim to prove that the labelling l that we obtain for the vertices of the graph G_ρ is edge-respecting. In order to prove this, we state the following invariant and we shall prove inductively that the invariant holds at the end of every iteration of the procedure.

Invariant: The labelling when restricted to the subgraph X is edge-respecting. Base case: X is the graph with just one vertex $(x, 0)$. In this case, any labelling for $(x, 0)$ is trivially edge-respecting.

Inductive Step: Let X be the portion of the graph labelled till now. By Induction hypothesis, the labelling of all vertices in X is edge-respecting. Now, let u be the vertex that is next labelled by the procedure. We wish to prove that the label $l(u)$ is such that the labelling l is an edge-respecting labelling of the graph $X \cup \{u\}$.

To argue that $l(u)$ is a valid edge-respecting label for u, suppose to the contrary, that it was not. Then there must exist a vertex v in $X \cup \{u\}$ and a directed path from u to v or from v to u which is inconsistent with the labels of u and v. First note that such a v cannot be equal to u. This is because, we know that there is no strict cycle in the graph and any path from u to itself must contain only $=$-labelled edges. The labelling of u is therefore, trivially consistent with every path from u to itself. Thus, the vertex v must be from X.

Observe now that if u was labelled by clause 2b, 2c, 2d, or 2e, then $l(u)$ must clearly be a valid label. If, for instance, u was labelled by clause 2b, then it must be the case that there is no directed path from u to any vertex in X. u could have been labelled inconsistently only if there is a vertex v in X, such that there is a strict directed path from v to u but $l(v) \geq l(u)$. But since u was labelled using clause 2b, $l(u) = p' > \max\{l(v) \mid v \in X_{<u}\}$. Therefore $l(u)$ must be $> l(v)$. Using a similar argument, it is possible to see that even if u was labelled by clauses 2c, 2d, or 2e, $l(u)$ must be a valid label.

So it must be the case that u was labelled by clause 2a. u could have been labelled inconsistently in the following three possible cases: there exists a strict directed path from u to v but $l(u) \geq l(v)$ or there exists a strict directed path from v to u but $l(u) \leq l(v)$ or there exists a directed path from u to v with all $=$-labelled edges but $l(u) \neq l(v)$. The first two cases are symmetric (and hence arguing for the first and third cases would suffice).

The first case can arise only if there is another vertex w in X such that there is an $=$-labelled path from w to u, and $l(u) = l(w)$. This gives us a strict directed path from w to v. This also means that $l(w) \geq l(v)$. But since l is an edge-respecting labelling for X, $l(w)$ must be $< l(v)$ giving rise to a contradiction. Hence, this case is not possible.

The third case, again, could only have occurred if u was labelled by 2a. Thus, there is a vertex w such that there is an $=$-labelled path from u to w and $l(u) = l(w)$. This now means that there is a directed path with all $=$-labelled edges from w to v (hence, also from v to w) but with $l(w) \neq l(v)$. Again, since l is an edge-respecting labelling for X, $l(w)$ must be equal to $l(v)$, giving rise to a contradiction.

This proves that the label l is an edge-respecting labelling for $X \cup \{u\}$. Thus, the invariant holds at the end of every iteration of the procedure. Hence the labelling l obtained is indeed an edge-respecting labelling for the graph G_ρ. Thus, every symbolic model ρ with all memoirs relevant and whose associated graph G_ρ has no strict cycle, admits a concrete model over D. ◀

B Details of Section 4

▷ Claim 17. Consider the function $f^i_x \colon (2^{\mathbb{N}})^l \to (2^{\mathbb{N}})^l$ given by: $f^i_x(X_1, X_2, \ldots, X_l) = (X'_1, X'_2, \ldots, X'_l)$ where $X'_k = \{y \mid \sim_{i,k} (x, y)\} \cup \{y \mid \exists k', \exists z \in X_{k'} \text{ such that } \sim_{k',k} (z, y)\} \cup X_k$. We claim that the function f^i_x has a least fixed-point.

Proof. Consider the partial ordering $\leq$ on $(2^{\mathbb{N}})^l$, given by $(X_1, \ldots, X_l) \leq (Y_1, \ldots, Y_l)$ if for all $i \in [1, l]$, $X_i \subseteq Y_i$. Notice that all subsets of $(2^{\mathbb{N}})^l$ have both a greatest lower bound and a least upper bound and thus, $((2^{\mathbb{N}})^l, \leq)$ forms a complete lattice. Also note that the function f^i_x as defined above is non-decreasing, hence a monotonic function. Thus, by Knaster-Tarski theorem [20], the set of fixed-points of $(2^{\mathbb{N}})^l$ forms a complete lattice under $\leq$. This guarantees the existence of a least fixed-point of f^i_x. ◁

▷ Claim 18. $\text{closed}^{\sim}_i(x, Y_1, \ldots, Y_l)$ holds iff $(Y_1, \ldots, Y_l)$ is a fixed-point of f^i_x.

Proof. $(Y_1, \ldots, Y_l)$ is a fixed-point of f^i_x

$\Leftrightarrow f^i_x((Y_1, \ldots Y_l)) = (Y_1, \ldots, Y_l)$.

$\Leftrightarrow \forall k \in [1, l], Y_k \cup \{y \mid \sim_{i,k} (x, y)\} \cup \{y \mid \exists k', \exists z \in X_{k'} \text{ such that } \sim_{k',k} (z, y)\} = Y_k$.

$\Leftrightarrow \forall k \in [1, l], \forall z$ if $\exists k', \exists x' \in Y_{k'}$ such that if one of $\sim_{k,k'} (x', z)$ or $\sim_{i,k} (x, z)$ holds, then, $z \in Y_{k'}$.

$\Leftrightarrow \forall x' \forall z (x' \in Y_{k'} \wedge \sim_{k',k} (x', z) \implies z \in Y_k) \wedge \forall z \bigwedge_{k \in [1,l]} (\sim_{i,k} (x, z) \implies z \in Y_k)$

$\Leftrightarrow \text{closed}^{\sim}_i(x, Y_1, \ldots, Y_l)$ holds ◁

▷ Claim 19. If $X^n_1, \ldots X^n_l$ is obtained upon n iterations of f^i_x to $(\emptyset, \ldots, \emptyset)$, and if $z \in X^n_k$, then there is a path of length $\leq n$ (but non-zero) from (v_i, x) to (v_k, z) in G_ρ.

Proof. Let us prove this by induction on n. Take $n = 1$ as the base case. By definition, $f^i_x((\emptyset, \ldots, \emptyset))$ gives us the tuple $(y_1, \ldots, y_l)$ such that there is an edge from (v_i, x) to (v_j, y_j) for all $j \in [1, l]$. Clearly, the hypothesis holds in the base case.

Inductive step: Suppose the inductive hypothesis is true for all $m < n$. Let $X_1^n, \dots X_l^n$ be obtained upon n iterations of f_x^i to $(\emptyset, \dots, \emptyset)$. Now by induction hypothesis, there is a path of length $\leq (n-1)$ from (v_i, x) to (v_k, z) for all $k \in [1, l]$, for all $z \in X_k^{n-1}$. Now consider $z \in X_k^n \setminus X_k^{n-1}$. By definition, it must be the case that there exists $k' \in [1, l]$ and $z' \in X_{k'}^{n-1}$ such that $\sim_{k',k} (z', z)$ holds. Now there is a path of length $\leq (n-1)$ from (v_i, x) to $(v_{k'}, z')$, hence there is a path of length $\leq n$ from (v_i, x) to (v_k, z). ◁

▷ Claim 20. $\mathrm{path}^{\sim}_{i,j}(x, y)$ holds iff there is a path of non-zero length from (v_i, x) to (v_j, y).

Proof. ($\Leftarrow$) We prove this by an induction on the length of the path. For the base case, assume there is a path of length 1, that is, $\sim_{i,j} (x, y)$ holds. Now by definition of the formula $\mathrm{closed}^{\sim}_i(x, Y_1, \dots, Y_l)$, and since $\sim_{i,j} (x, y)$, for all $Y_1, \dots, Y_l$, if $\mathrm{closed}^{\sim}_i(x, Y_1, \dots, Y_l)$ holds then $y \in Y_j$ must hold. Or in other words, the formula $\mathrm{path}^{\sim}_{i,j}(x, y)$ holds.

Inductive step: Let us assume that there is a path of length n from (v_i, x) to (v_j, y). This means that there exists (v_k, z) such that there is a path of length $(n-1)$ from (v_i, x) to (v_k, z) and such that $\sim_{k,j} (z, y)$ holds. By induction hypothesis, this means that, both, $\mathrm{path}^{\sim}_{i,k}(x, z)$ and $\sim_{k,j} (z, y)$ hold. By the definitions of $\mathrm{path}^{\sim}_{i,k}(x, z)$ and $\mathrm{closed}^{\sim}_i(x, Y_1, \dots, Y_l)$, we get that $\forall Y_1 \dots \forall Y_l (\mathrm{closed}^{\sim}_i(x, Y_1, \dots, Y_l) \implies y \in Y_j)$ holds, or in other words, $\mathrm{path}^{\sim}_{i,j}(x, y)$ holds.

($\Rightarrow$) We assume that $\mathrm{path}^{\sim}_{i,j}(x, y)$ holds. This means, using Claim 18, that if $(Y_1, \dots, Y_l)$ is a fixed-point, then $y \in Y_j$. In particular, if $(Y_1, \dots, Y_l)$ is the least fixed-point, then $y \in Y_j$. We first prove that the least fixed-point of f_x^i can be obtained by iteratively applying f_x^i on $(\emptyset, \dots, \emptyset) \leq \omega$ times. In order to prove this, it is sufficient to prove that $(f_x^i)^{\omega+1}((\emptyset, \dots, \emptyset)) = (f_x^i)^{\omega}((\emptyset, \dots, \emptyset))$. By definition, $(f_x^i)^{\omega+1}((\emptyset, \dots, \emptyset)) = f_x^i((f_x^i)^{\omega}((\emptyset, \dots, \emptyset)))$ and $(f_x^i)^{\omega}((\emptyset, \dots, \emptyset)) = \bigcup_{n=0}^{\infty} (f_x^i)^n((\emptyset, \dots, \emptyset))$. Let us assume $(f_x^i)^{\omega}((\emptyset, \dots, \emptyset)) = (X_1, \dots X_l)$ and let $f_x^i((X_1, \dots X_l)) = (X_1', \dots X_l')$. Suppose for some $k \in [1, l]$, $X_k \subset X_k'$. This means that there exists $y \in X_k' \setminus X_k$ such that for some $k' \in [1, l]$ and for some $z \in X_{k'}$, $\sim_{k',k} (z, y)$ holds. But since, $X_{k'}$ contains z, the position z must have been added to the k'^{th} component of the l-tuple, in some n^{th} iteration of the function f_x^i for some $n \in \mathbb{N}$. This means that y should have been added to the k^{th} component of the l-tuple, in the $(n+1)^{\text{th}}$ iteration of f_x^i contradicting our assumption that $y \in X_k' \setminus X_k$. This means that $X_k' = X_k$ for all $k \in [1, l]$ implying that $(f_x^i)^{\omega+1}((\emptyset, \dots, \emptyset)) = (f_x^i)^{\omega}((\emptyset, \dots, \emptyset))$.

Now, if the least fixed-point of f_x^i can be obtained by applying f_x^i n times on $(\emptyset, \dots, \emptyset)$ for some $n \in \mathbb{N}$, then by Claim 19, there exists a path of length $\leq n$ from (v_i, x) to (v_j, y). Now, if the least fixed-point of f_x^i is obtained by applying f_x^i ω times on $(\emptyset, \dots, \emptyset)$, we need to perform a transfinite induction in order to prove that there is still a path from (v_i, x) to (v_j, y) in this case. Let $(f_x^i)^{\omega}((\emptyset, \dots, \emptyset)) = (Y_1, \dots, Y_l)$ be the least fixed-point. We know $y \in Y_j$. Now by definition of $(f_x^i)^{\omega}$, the element y must have been added to the k^{th} component of the l-tuple in the n^{th} iteration of f_x^i for some $n \in \mathbb{N}$. But then, this means that there is a path of length $\leq n$ from (v_i, x) to (v_j, y). Thus, there is a path of non-zero length from (v_i, x) to (v_j, y) even in this case, as required. ◁

C Details of Section 5

Proof of Lemma 9. ($\Rightarrow$) ρ admits some concrete model σ. Let l be the corresponding edge-respecting labelling of the graph G_ρ. Then, clearly, $slen(u, v) \leq |l(v) - l(u)|$, for all vertices $u, v \in G_\rho$. Thus, there cannot exist vertices u and v with $slen(u, v) = \omega$.

($\Leftarrow$) Suppose G_ρ satisfies the given condition. We outline a procedure that gives an edge-respecting $\mathbb{Z}$-labelling $l \colon V \times \mathbb{N} \to \mathbb{Z}$ of G_ρ.

We assume an ordering $\prec$ on V and use it to define an ordering on the vertices of G_ρ given by $(x, i) \prec (y, j)$ iff $i < j$, or $i = j$ and $x \prec y$. We now give a procedure to label the vertices.

1. Label the vertices in order of $\prec$. Begin by labelling the first, say $(x, 0)$ by 0.
2. In general, let X be the portion of the graph already labelled, and u be the next vertex to be labelled. Let $X_{\leq u}$ denote the set of vertices in X from which there is a directed path to u and $X_{u\leq}$ denote the set of vertices in X to which there is a directed path from u. Now we can label u as follows:
   (a) if $X_{u\leq}$ is non-empty then set $l(u) = \min\{(l(v) - \mathit{slen}(u, v)) \mid v \in X, \exists$ a path from u to $v\}$, else,
   (b) if $X_{\leq u}$ is non-empty, set $l(u) = \max\{(l(v) + \mathit{slen}(v, u)) \mid v \in X, \exists$ a path from v to $u\}$, else
   (c) if both $X_{\leq u}$ and $X_{u\leq}$ are empty, label u by 0.

We now aim to prove that the labelling l that we obtain for the vertices of the graph G_ρ is edge-respecting. In order to prove this, we state the following invariant and we shall prove inductively that the invariant holds at the end of every iteration of the procedure.

Invariant: The labelling when restricted to the subgraph X is edge-respecting. Base case: X is the graph with just one vertex $(x, 0)$. In this case, any labelling for $(x, 0)$ is trivially edge-respecting.

Inductive Step: Let X be the portion of the graph labelled till now. By Induction hypothesis, the labelling of all vertices in X is edge-respecting. Now, let u be the vertex that is next labelled by the procedure. We wish to prove that the label $l(u)$ is such that the labelling l is an edge-respecting labelling of the graph $X \cup \{u\}$.

To argue that $l(u)$ is an valid edge-respecting label for u, suppose to the contrary, that it was not. Then there must exist a vertex v in $X \cup \{u\}$ and a directed path from u to v or from v to u which is inconsistent with the labels of u and v. First note that such a v cannot be equal to u. This is because, we know that there is no strict cycle in the graph and any path from u to itself must contain only $=$-labelled edges. The labelling of u is therefore trivially consistent with every path from u to itself. Thus the vertex v must be from X.

Therefore, u could have been labelled inconsistently in the following two possible cases: there exists a directed path p from u to v but $\mathit{slen}(p) > l(v) - l(u)$ or there exists a directed path p from v to u but $\mathit{slen}(p) > l(u) - l(v)$.

For the first case to arise it must be the case that u was labelled by clause 2a. But then the label $l(u)$ must be such that $l(u) \leq l(v) - \mathit{slen}(p)$. Thus, this case is not possible.

There are two possibilities for the second case. In the first possibility, u was labelled by clause 2a. So there must have been a vertex w in X with a path q from u to w, and $l(u) = l(w) - \mathit{slen}(q)$. But since v and w were labelled without any discrepancy, it must be the case that $l(w) - l(v) \geq \mathit{slen}(p) + \mathit{slen}(q)$. Thus, $l(u) = l(w) - \mathit{slen}(q) \geq l(v) + \mathit{slen}(p)$. This contradicts our assumption that $\mathit{slen}(p) > l(u) - l(v)$, and hence this possibility is ruled out. The second possibility was that u was labelled by an application of clause 2b of the procedure. But then it must be the case that $l(u) \geq l(v) + \mathit{slen}(p)$. Thus, this case is also ruled out.

This proves that the label l is an edge-respecting labelling for $X \cup \{u\}$. Thus, the invariant holds at the end of every iteration of the procedure and hence, the labelling l obtained is indeed an edge-respecting labelling for the graph G_ρ. Hence, every consistent symbolic model ρ satisfying $\mathit{slen}(u, v) < \omega$ for all $u, v \in G_\rho$, admits a concrete model over $\mathbb{Z}$. ◀

Proof of Claim 10. There can be two cases.
Case 1: $i - j \leq n_e(|\delta| + 1)$.

In this case, $(i - mmp(i, e)) \leq (i - j) < 2n_e|\delta|$.

Case 2: $i - j > n_e(|\delta| + 1)$.

Consider a run of A_e on the word $\rho[j, i]^R$. Now since, $i - j > n_e(|\delta| + 1)$, at least one state s of A_e must be visited at a pair of isomorphic positions along this run. Consider the first state s along the run, that is visited at a pair of isomorphic positions. Let k_1 and k_2 be the first pair of isomorphic positions at which s is visited. Consider the portion of the run from i to k_1. Each state of A_e can be visited at most $|\delta|$ times in this portion, hence $i - k_1 \leq n_e|\delta|$. Now since k_1 and k_2 are isomorphic positions, $k_1 - k_2 = m|\delta|$ for some $m \in \mathbb{N}$.

Also, since between positions k_1 and k_2, every state of A_e can be visited at most $|\delta|$ times, $m \leq n_e$.

Now the sequence of states visited between positions k_1 and k_2 repeats itself along the run until the last time the state s is visited at a position isomorphic to k_1 along the run. Let k_l be the last position isomorphic to k_1, at which s is visited. We know a final state F is visited at position j. Now, since k_1 and k_l are isomorphic, the final state F must have been visited at position $k_1 - (k_l - j) = k_1 - k_l + j$. This means that $\mathrm{mmp}(i, e) = k_1 - k_l + j$. Therefore, $i - \mathrm{mmp}(i, e) = (i - k_1) + (k_l - j) \leq 2n_e|\delta|$. Hence, proved. $\triangleleft$