

Metric Propositional Neighborhood Logic with an Equivalence Relation

Angelo Montanari · Marco Pazzaglia · Pietro Sala

Received: date / Accepted: date

Abstract The propositional interval logic of temporal neighborhood (PNL for short) features two modalities that make it possible to access intervals adjacent to the right (modality $\langle A \rangle$) and to the left (modality $\langle \bar{A} \rangle$) of the current interval. PNL stands at a central position in the realm of interval temporal logics, as it is expressive enough to encode meaningful temporal conditions and decidable (undecidability rules over interval temporal logics, while PNL is NEXPTIME-complete). Moreover, it is expressively complete with respect to the two-variable fragment of first-order logic extended with a linear order $\text{FO}^2[\langle \cdot \rangle]$. Various extensions of PNL have been studied in the literature, including metric, hybrid, and first-order ones. Here, we study the effects of the addition of an equivalence relation \sim to Metric PNL (MPNL \sim). We first show that the finite satisfiability problem for PNL extended with \sim is still NEXPTIME-complete. Then, we prove that the same problem for MPNL \sim can be reduced to the decidable 0-0 reachability problem for vector addition systems and vice versa (EXPSpace-hardness immediately follows).

1 Introduction

In this paper, we study the effects of adding an equivalence relation to the metric interval logic of temporal neighborhood (MPNL), interpreted over the class of

Angelo Montanari
Department of Mathematics and Computer Science,
University of Udine, Italy
E-mail: angelo.montanari@uniud.it

Marco Pazzaglia
Bending Spoons
Milano, Italy
E-mail: marco@pazzaglia.me

Pietro Sala
Department of Computer Science,
University of Verona, Italy
E-mail: pietro.sala@univr.it

finite linear orders. Benefits that arise from extending a logic with one or more equivalence relations have been already pointed out in the literature. As an example, the addition of an equivalence relation to the two-variable fragment of first-order logic FO^2 on (finite) data words (words where each position consists of a pair whose first element belongs to a finite alphabet and the second one, the data value, to an infinite set) makes it possible to check whether two word positions carry the same data value [2]. In most cases, such an increase in expressiveness comes with a blow-up in complexity (see, for instance, the fields of timed automata [22], temporal logics [8], and semistructured data [2]).

Overview. In the context of first-order logic, the finite satisfiability problem for the two-variable fragment of first-order logic FO^2 extended with one, two, or more equivalence relations has been systematically explored in [13–15], while in [2] Bojańczyk et al. investigated the extension of FO^2 over finite and infinite data words with an equivalence relation. In the temporal setting, similar analyses have been done by Demri and Lazic [8], that studied the extension of linear temporal logic over data words by means of freeze quantifiers, which allow one to store elements at the current word position into a register and then to use them in equality comparisons deeper in the formula, and by Ouaknine and Worrell [22], who showed that both satisfiability and model checking for metric temporal logic over finite timed words are decidable with a non-primitive recursive complexity.

The addition of an equivalence relation to an interval temporal logic has been first investigated by Montanari and Sala in [19]. They focused on the interval logic of Allen’s relations *meets*, *begun by*, and *begins* extended with an equivalence relation $\sim (AB\bar{B}\sim)$, interpreted over finite linear orders and \mathbb{N} . On the one hand, they showed that the resulting increase in expressive power makes it possible to establish an original connection between interval temporal logics and extended regular languages of finite and infinite words [1]. On the other hand, they proved that the addition of \sim to $AB\bar{B}$, whose satisfiability problem is known to be EXPSPACE-complete [18], drastically increases its computational complexity: $AB\bar{B}\sim$ turns out to be decidable, with a non-primitive recursive complexity, over finite linear orders and undecidable over \mathbb{N} . As a preliminary step, they gave a geometrical interpretation to $AB\bar{B}\sim$ models in terms of compass structures [24], where each interval $[a, b]$, with $a \leq b$, is interpreted as a point (a, b) in the plane and interval relations are replaced by corresponding relations between points. Decidability of $AB\bar{B}\sim$ over finite linear orders rests on a suitable small model property, based on a contraction method that, under suitable conditions, makes it possible to collapse two rows, and all rows in between, of the compass structure. Then, they provided an encoding of the 0-0 reachability problem for lossy Minsky counter machines, which is known not to be a recursive primitive problem, in the finite satisfiability problem for $AB\bar{B}\sim$, and an encoding of the 0- N reachability problem for the same class of machines, which is known to be undecidable, into the satisfiability problem for $AB\bar{B}\sim$ over \mathbb{N} .

In [4], Bresolin et al. established a strong connection between FO^2 and PNL, proving that the latter is expressively complete with respect to the former extended with a linear order $<$, denoted $\text{FO}^2[<]$. Moreover, there are polynomial reductions in both directions (the polynomial PNL-to- $\text{FO}^2[<]$ reduction is straightforward, while the exponential $\text{FO}^2[<]$ -to-PNL reduction given in [4] can be turned into a polynomial one by first applying Scott’s normal form [23] to $\text{FO}^2[<]$ formulas,

which can be computed in polynomial time and reduces the quantifier depth of the formulas to two¹). Such a bridge allows useful transfers in both directions.

Decidable and undecidable extensions of FO². Various extensions of FO² with one or more equivalence relations, the ordering relation $<$, and/or the successor relation $+1$ have been already studied in the literature. In the following, we will denote by FO²[rel_1, \dots, rel_k] the extension of FO² over arbitrary signatures, with rel_1, \dots, rel_k as the only pre-interpreted relations, and by FO²(rel_1, \dots, rel_k) the extension of FO² where, besides the pre-interpreted relations rel_1, \dots, rel_k , only unary symbols can be used.

In [20], Mortimer showed that FO² enjoys the finite model property and thus its satisfiability problem is decidable. From his proof, a doubly exponential bound on the size of models can be extracted. Later on [10], Grädel et al. proved that the finite satisfiability problem for FO² is in fact NEXPTIME-complete.

A systematic analysis of the extensions of FO² with one or more equivalence relations was started by Kieronski and Otto in [14], where they proved that the finite satisfiability problem for FO² extended with one or two equivalence relations is decidable, while it becomes undecidable when three or more equivalence relations are added. As a preliminary step, they showed that FO² has a small model substructure property, that is, one can substitute a part of the domain of any size (even infinite) with an equivalent part whose size is bounded in the size of the vocabulary (such a result generalizes the small model property for FO² given in [10]). Then, they exploited this property both to prove that FO²[\sim] itself has a small model property (they provided an exponential upper bound to the size of the model) and to reduce the satisfiability problem for FO²[\sim_1, \sim_2] to an equivalent combinatorial problem for an infinite chessboard. The undecidability of the satisfiability problem for FO²[\sim_1, \sim_2, \sim_3] is proved by means of a reduction from a suitable tiling problem. As for the complexity of the decidable extensions, they showed that finite satisfiability for FO²[\sim] is NEXPTIME-complete, as in the case of FO². Later [15], Kieronski and Tendera proved that finite satisfiability for FO²[\sim_1, \sim_2] is in 3-NEXPTIME, and then, in [13], Kieronski et al. further refined such a result showing that it is in fact 2-NEXPTIME-complete.

By building on techniques from [10] and performing an in-depth analysis of the basic 1-types and 2-types in FO²[$<$]-models, Otto proved that FO²[$<$] is decidable in NEXPTIME over the class of all linear orders, as well as over some natural subclasses of it, including the class of finite linear orders [21].

In Section 3, we generalize Kieronski and Otto's result about NEXPTIME-completeness of finite satisfiability for FO²[\sim] [14], as well as Otto's result about NEXPTIME-completeness of finite satisfiability for FO²[$<$] [21], by showing that finite satisfiability for PNL extended with an equivalence relation \sim , which is equivalent to FO²[$<, \sim$], is NEXPTIME-complete as well. The proof rests on a model contraction technique that extends the point elimination technique used to demonstrate the decidability of finite satisfiability for PNL in [5]. Unfortunately, the proof cannot be lifted to FO²[$<, \sim_1, \sim_2$] in any natural way. As a matter of fact, the extension of FO² with an equivalence relation and a linear order was already considered, albeit in a restricted form, by Bojańczyk et al. in [2], where

¹ The polynomial reduction from FO²[$<$] to PNL is outlined in the appendix.

the authors proved that $\text{FO}^2(<, \sim)$ on data words, when interpreted over the class of finite linear orders or \mathbb{N} , is NEXPTIME-complete.

In [2], Bojańczyk et al. also showed that the extension of FO^2 with three binary relations, interpreted as a linear order, a successor relation, and an equivalence relation, and an unlimited number of unary relations, denoted by $\text{FO}^2(\sim, <, +1)$, is decidable over finite and infinite data words (that is, finite linear orders and \mathbb{N}). Their proof consists of three steps: (i) an $\text{FO}^2(\sim, <, +1)$ -formula is rewritten in a suitable normal form; (ii) the resulting formula in normal form is transformed into an equivalent data automaton; (iii) the emptiness problem for data automata is reduced (by a 2-EXPTIME-reduction) to the emptiness problem for vector addition systems (VAS), which is known to be decidable [16] (even though no elementary upper bound is known). They also provided an inverse reduction, from the emptiness problem for VAS to the satisfiability problem for $\text{FO}^2(\sim, <, +1)$, which is computable in PTIME.

In Section 4, we improve such a result, as far as finite linear orders are concerned, by substituting binary relations for unary ones. We prove the decidability of finite satisfiability for $\text{MPNL}\sim$, which is equivalent to $\text{FO}^2[<, \sim, +1]$, by reducing such a problem to the decidable 0-0 reachability problem for vector addition systems. EXPSPACE-hardness immediately follows from the polynomial-time reduction from the emptiness problem for VAS to the finite satisfiability problem for $\text{FO}^2(\sim, <, +1)$ given in [2].

Contributions and organization of the paper. To summarize, in this paper we investigate extensions of PNL (equivalently, $\text{FO}^2[<]$) with an equivalence and a successor relation. The outcomes of the paper can be summarized as follows. First, we prove that the NEXPTIME-completeness of the finite satisfiability problem for PNL is preserved under addition of an equivalence relation \sim . Then, we focus on MPNL . In [7], Bresolin et al. proved that the satisfiability problem for MPNL over finite linear orders is EXPSPACE-complete. Here, we show that the finite satisfiability problem for $\text{MPNL}\sim$ is decidable with a non-primitive recursive complexity and we prove its EXPSPACE-hardness. It is worth pointing out that, as in [2], the successor is assumed to be in relational form (it can be easily checked that the relational and functional characterizations of the successor are not equivalent in $\text{FO}^2[<]$). As a matter of fact, the addition of a successor function to PNL immediately yields undecidability, as shown in [3].

The rest of the paper is organized as follows. In Section 2, we provide some background knowledge. In Section 3, we prove that the addition of an equivalence relation to PNL does not lead to any increase in computational complexity. In Section 4, we study the extension of MPNL with an equivalence relation. Conclusions provide an assessment of the work done and outline future research directions. An appendix with an outline of the polynomial reduction from $\text{FO}^2[<]$ to PNL and all the missing proofs concludes the paper.

2 Preliminaries

In this section, we give syntax and semantics of Metric PNL and we show how to extend it with an equivalence relation \sim . Moreover, we introduce basic notions and

terminology that will be used in the following sections. Finally, we briefly recall the notion of vector addition system.

2.1 (Metric) Propositional Neighborhood Logic and extensions

The language of PNL consists of a set \mathcal{AP} of proposition letters, the propositional connectives \neg and \vee , modalities $\langle A \rangle$ and $\langle \bar{A} \rangle$ for Allen's relations *meets* and *met by*, respectively, and a modal constant π (also denoted by len_0) [9]. MPNL is obtained from PNL by adding an infinite set of (pre-interpreted) proposition letters $\text{len}_1, \dots, \text{len}_k, \dots$ for length constraints, that allow one to constrain the length of the current interval to be equal to $1, 2, \dots$ [3]. Formulae of MPNL, denoted by φ, ψ, \dots , are generated by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \psi \mid \langle A \rangle\varphi \mid \langle \bar{A} \rangle\varphi \mid \text{len}_k,$$

where $p \in \mathcal{AP}$ and $k \in \mathbb{N}$.

The other propositional connectives, the logical constants \top (*true*) and \perp (*false*), and the dual modalities $[A]$ and $[\bar{A}]$ are defined as usual (see below). Hereafter, we denote by $|\varphi|$ the size of φ , which equals the sum of the number of uninterpreted proposition letters in φ , the number of logical connectives and temporal modalities in φ , and, for each pre-interpreted proposition letter len_k in φ , the summation of the number of digits of the binary encoding of k .

Given a linearly-ordered domain $\mathbb{D} = \langle D, < \rangle$, a (*non-strict*) *interval* over \mathbb{D} is an ordered pair $[x, y]$, with $x = y$ or $x < y$ ($x \leq y$ for short). We denote by $\mathbb{I}(\mathbb{D})$ the set of all intervals over \mathbb{D} . From now on, we assume \mathbb{D} to be a finite prefix of \mathbb{N} (a finite linear order). The semantics of MPNL is given in terms of *models* of the form $M = \langle \mathbb{I}(\mathbb{D}), V \rangle$, where $V : \mathcal{AP} \rightarrow 2^{\mathbb{I}(\mathbb{D})}$ is a valuation function assigning a set of intervals to every proposition letter. We recursively define the truth relation \Vdash as follows:

- $M, [x, y] \Vdash p$ iff $[x, y] \in V(p)$, for any $p \in \mathcal{AP}$;
- $M, [x, y] \Vdash \neg\varphi$ iff it is not the case that $M, [x, y] \Vdash \varphi$;
- $M, [x, y] \Vdash \varphi \vee \psi$ iff $M, [x, y] \Vdash \varphi$ or $M, [x, y] \Vdash \psi$;
- $M, [x, y] \Vdash \langle A \rangle\varphi$ iff $M, [y, z] \Vdash \varphi$ for some $z \geq y$;
- $M, [x, y] \Vdash \langle \bar{A} \rangle\varphi$ iff $M, [z, x] \Vdash \varphi$ for some $z \leq x$;
- $M, [x, y] \Vdash \text{len}_k$ iff $y - x = k$, for all $k \geq 0$.

An MPNL formula φ is said to be *satisfiable* if there exist a model $M = \langle \mathbb{I}(\mathbb{D}), V \rangle$ and an interval $[x, y] \in \mathbb{I}(\mathbb{D})$ such that $M, [x, y] \Vdash \varphi$. For technical reasons (namely, to ensure the correctness of the contraction technique we will use in the decidability proof for PNL \sim), we assume φ to be of the form $\langle A \rangle\psi$. It can be easily shown that ψ is satisfiable if and only if $\varphi (= \langle A \rangle\psi)$ is satisfiable.

The extensions of PNL and MPNL with an equivalence relation, respectively denoted by PNL \sim and MPNL \sim , are obtained by adding a special symbol \sim to their language and by interpreting it as an equivalence relation on the domain. Formally, each model is augmented with an equivalence relation \sim and the symbol \sim is interpreted according to the following rule:

- $M, [x, y] \Vdash \sim$ iff $x \sim y$.

As usual, we denote by $[x]_{\sim}$ the \sim -class x belongs to. For the sake of simplicity, hereafter we restrict our attention to (a fragment of) MPNL \sim with π (that is, len_0) and len_1 only. The generalization to len_k , for all $k \geq 0$, is straightforward.

2.2 Atoms, interval tuples, labeled interval structures, and temporal requests

Let φ be an MPNL \sim formula. W.l.o.g., we assume that both len_1 and \sim occur in φ , as if len_1 (resp., \sim) is missing, φ belongs to PNL \sim (resp., to MPNL). We define the *closure* of φ , denoted by $Cl(\varphi)$, as the set of all sub-formulas of φ and their negations (we replace all formulas of the form $\neg\neg\psi$ in $Cl(\varphi)$ by ψ), plus the sub-formulas $\langle A \rangle \text{len}_1$, $[A] \neg \text{len}_1$, $\langle \bar{A} \rangle \text{len}_1$, and $[\bar{A}] \neg \text{len}_1$, $\langle A \rangle (\text{len}_1 \wedge \sim)$, and $[A] \neg (\text{len}_1 \wedge \sim)$.

A φ -atom (atom for short) F is a maximal, syntactically-consistent subset of $Cl(\varphi)$, that is, a set F such that (i) $\psi \in F$ if and only if $\neg\psi \notin F$, for all $\psi \in Cl(\varphi)$, (ii) $\psi \in F$ if and only if $\psi_1 \in F$ or $\psi_2 \in F$, for all $\psi = \psi_1 \vee \psi_2 \in Cl(\varphi)$. We denote the set of all φ -atoms by $Atoms_{\varphi}$.

Let Tr_{φ} be the set of all temporal formulas in $Cl(\varphi)$, that is, the subset of $Cl(\varphi)$ consisting of all and only the formulas of the forms $\langle A \rangle \psi$, $\langle \bar{A} \rangle \psi$, $[A] \psi$, and $[\bar{A}] \psi$ (we write $[A] \neg \psi$ for $\neg \langle A \rangle \psi$, $[\bar{A}] \neg \psi$ for $\neg \langle \bar{A} \rangle \psi$, $\langle A \rangle \neg \psi$ for $\neg [A] \psi$, and $\langle \bar{A} \rangle \neg \psi$ for $\neg [\bar{A}] \psi$), and let \mathcal{R}_{φ} be the set of all maximal, syntactically-consistent subsets R of Tr_{φ} (that is, $\langle A \rangle \psi \in R$ if and only if $[A] \neg \psi \notin R$, and so on). It can be easily shown that $|\mathcal{R}_{\varphi}|$ is bounded by $2^{(|\varphi|+2)}$. For each $R \in \mathcal{R}_{\varphi}$, we define the restrictions $R|_A = \{\langle A \rangle \psi \in R\}$ and $R|_{\bar{A}} = \{\langle \bar{A} \rangle \psi \in R\}$. We put $\mathcal{R}_{\varphi|_A} = \{R|_A : R \in \mathcal{R}_{\varphi}\}$ and $\mathcal{R}_{\varphi|_{\bar{A}}} = \{R|_{\bar{A}} : R \in \mathcal{R}_{\varphi}\}$.

Given $F \in Atoms_{\varphi}$ and $R_1, R_2 \in \mathcal{R}_{\varphi}$, the triple (R_1, F, R_2) is an *interval-tuple* if and only if it holds that: (i) for each $\langle A \rangle \psi \in Tr_{\varphi}$, if $\langle A \rangle \psi \notin R_1$, then $\neg\psi \in F$, and for each $\langle \bar{A} \rangle \psi \in Tr_{\varphi}$, if $\langle \bar{A} \rangle \psi \notin R_2$, then $\neg\psi \in F$; (ii) for each $\langle A \rangle \psi \in Tr_{\varphi}$, $\langle A \rangle \psi \in F$ if and only if $\langle A \rangle \psi \in R_2$; (iii) for each $\langle \bar{A} \rangle \psi \in Tr_{\varphi}$, $\langle \bar{A} \rangle \psi \in F$ if and only if $\langle \bar{A} \rangle \psi \in R_1$; (iv) for each $\psi \in F$, if $\langle A \rangle \psi \in Tr_{\varphi}$, then $\langle A \rangle \psi \in R_1$, and if $\langle \bar{A} \rangle \psi \in Tr_{\varphi}$, then $\langle \bar{A} \rangle \psi \in R_2$.

W.l.o.g., hereafter we restrict ourselves to sets of requests $R \in \mathcal{R}_{\varphi}$ such that there exists an interval tuple (R, F, R) for some atom $F \in Atoms_{\varphi}$ including π .

Let $LR_{\varphi} \subseteq Atoms_{\varphi} \times Atoms_{\varphi}$ be such that for each pair of atoms $A_1, A_2 \in Atoms_{\varphi}$, $A_1 LR_{\varphi} A_2$ if and only if (i) for each $[A] \psi \in Cl(\varphi)$, if $[A] \psi \in A_1$, then $\psi \in A_2$, and (ii) for each $[\bar{A}] \psi \in Cl(\varphi)$, if $[\bar{A}] \psi \in A_2$, then $\psi \in A_1$. A (φ) -labeled interval structure (LIS for short) is a pair $\mathbf{L} = \langle \mathbb{D}, \mathcal{L} \rangle$, where \mathbb{D} is a finite linear order and $\mathcal{L} : \mathbb{I}(\mathbb{D}) \rightarrow Atoms_{\varphi}$ is a *labeling function* such that:

1. for each pair of neighboring intervals $[x, y], [y, z] \in \mathbb{I}(\mathbb{D})$, it holds that $\mathcal{L}([x, y]) LR_{\varphi} \mathcal{L}([y, z])$;
2. for all $x, y, z \in D$, with $x \leq y \leq z$, if $\sim \in \mathcal{L}([x, y]) \cap \mathcal{L}([y, z])$, then $\sim \in \mathcal{L}([x, z])$, if $\sim \in \mathcal{L}([x, z]) \cap \mathcal{L}([x, y])$, then $\sim \in \mathcal{L}([y, z])$, and if $\sim \in \mathcal{L}([x, z]) \cap \mathcal{L}([y, z])$, then $\sim \in \mathcal{L}([x, y])$;
3. for all $x, y \in D$, with $x < y$, $\text{len}_1 \in \mathcal{L}([x, y])$ if and only if for all $z \in D$, with $x \leq z \leq y$, either $z = x$ or $z = y$.

We say that a point $x \in D$ is fulfilled if and only if (i) for all $\langle A \rangle \psi \in Tr_{\varphi}$ and $y (\leq x) \in D$, if $\langle A \rangle \psi \in \mathcal{L}([y, x])$, then there exists $z \geq x$ such that $\psi \in \mathcal{L}([x, z])$ and (ii) for all $\langle \bar{A} \rangle \psi \in Tr_{\varphi}$ and $(x \leq) y \in D$, if $\langle \bar{A} \rangle \psi \in \mathcal{L}([x, y])$, then there exists $z \leq x$

such that $\psi \in \mathcal{L}([z, x])$. A LIS $\mathbf{L} = \langle \mathbb{D}, \mathcal{L} \rangle$ is *fulfilling* if and only if all points in D are fulfilled. A fulfilling LIS $\mathbf{L} = \langle \mathbb{D}, \mathcal{L} \rangle$ satisfies φ if and only if there exists an interval $[x, y] \in \mathbb{I}(\mathbb{D})$ such that $\varphi \in \mathcal{L}([x, y])$. The following theorem establishes a link between LISs and models of an MPNL \sim formula.

Theorem 1 *An MPNL \sim formula φ is satisfiable over a finite linear order \mathbb{D} if and only if there exists a fulfilling LIS $\mathbf{L} = \langle \mathbb{I}(\mathbb{D}), \mathcal{L} \rangle$ that satisfies φ .*

The proof is straightforward and thus omitted. In the following, we will often write ‘a fulfilling LIS for φ ’ for ‘a fulfilling LIS that satisfies φ ’.

Finally, we associate with each point of the temporal domain of a LIS a set of future temporal requests and a set of past temporal requests. Let $\mathbf{L} = \langle \mathbb{D}, \mathcal{L} \rangle$ be a LIS and $x \in D$. The set of *future temporal requests* of x is the set $\mathbf{req}|_A(x) = \{ \langle A \rangle \psi \in Tr_\varphi : \exists y \in D \langle A \rangle \psi \in \mathcal{L}([y, x]) \} \cup \{ [A] \psi \in Tr_\varphi : \exists y \in D [A] \psi \in \mathcal{L}([y, x]) \}$, while the set of *past temporal requests* of x is the set $\mathbf{req}|_{\bar{A}}(x) = \{ \langle \bar{A} \rangle \psi \in Tr_\varphi : \exists y \in D \langle \bar{A} \rangle \psi \in \mathcal{L}([x, y]) \} \cup \{ [\bar{A}] \psi \in Tr_\varphi : \exists y \in D [\bar{A}] \psi \in \mathcal{L}([x, y]) \}$. For each $x \in D$, let $\mathbf{req}(x) = \mathbf{req}|_A(x) \cup \mathbf{req}|_{\bar{A}}(x)$. It trivially holds that $|\mathbf{req}(x)| (= |\mathbf{req}|_A(x)| + |\mathbf{req}|_{\bar{A}}(x)|)$ is bounded by $|\varphi| + 2$.

2.3 Vector addition systems

We conclude the section with a short account of vector addition systems (VAS). In Section 4, we will provide a reduction from the satisfiability problem for MPNL \sim to the 0-0 reachability problem for vector addition systems with states (VASS for short). A non-primitive recursive decision procedure for the latter problem can be found in [16], while a proof of its EXPSpace-hardness is given in [17].

Alternative, equally-expressive characterizations of VAS / VASS have been proposed in the literature, e.g., [11, 12]. In the following, we will provide a slightly different, but equivalent, formulation of VASS that better suits our needs. A VASS \mathcal{A} is a quintuple (Q, Q_i, Q_f, C, Δ) , where Q is a finite set of states, Q_i and Q_f are the subsets of Q representing initial and final states, respectively, C is a finite set of counters $\{c_1, \dots, c_n\}$, and Δ is the transition relation.

We define a set U of updates on C as a *finite* set of operations of the form $c_i + k$, with $c_i \in C$ and $k \in \mathbb{Z}$, to be executed as a single (atomic) operation. W.l.o.g., we can assume that U includes at most one operation per counter. This allows us to provide each set of updates U with a vector interpretation $v_U \subseteq \mathbb{Z}^{|C|}$ as follows: for each $c_i \in C$, $v_U[c_i] = k$, if $c_i + k \in U$; $v_U[c_i] = 0$ otherwise (no operation involving c_i belongs to U).

Let us denote by \mathcal{U} the set of all possible sets of updates. The transition relation Δ is a subset of $Q \times \mathcal{U} \times Q$. A configuration of the VASS \mathcal{A} is a pair $(q, v) \in Q \times \mathbb{N}^{|C|}$. We say that a transition $(q', U, q'') \in \Delta$ is *active* in a configuration (q, v) if (and only if) $q' = q$ and $v + v_U \in \mathbb{N}^{|C|}$. Let $Conf$ be the set of all possible configurations. We define a binary relation $\rightarrow_{\mathcal{A}} \subseteq Conf \times Conf$ as follows: for all $(q, v), (q', v') \in Conf$, $(q, v) \rightarrow_{\mathcal{A}} (q', v')$ if and only if there is a transition $(q, U, q') \in \Delta$ such that (q, U, q') is active in (q, v) and $v' = v + v_U$.

Let $\rightarrow_{\mathcal{A}}^*$ be the reflexive and transitive closure of $\rightarrow_{\mathcal{A}}$. The 0-0 *reachability problem* for \mathcal{A} consists of deciding whether there exist $q_i \in Q_i$ and $q_f \in Q_f$ such that $(q_i, 0) \rightarrow_{\mathcal{A}}^* (q_f, 0)$.

3 Adding an equivalence relation to PNL: PNL \sim

In this section, we prove that the addition of an equivalence relation \sim to PNL preserves decidability (in fact, it does not cause any increase in complexity). However, proving decidability of PNL \sim turns out to be more involved than in the case of PNL (it is worth pointing out that this is not the case with RPNL, the future fragment of PNL: if we restrict ourselves to RPNL the original decidability proof can be easily adapted to the extended logic RPNL \sim).

The proof consists of three lemmas. The first one provides an (exponential) upper bound on the size of an (equivalence) class in a minimal model. The second lemma gives a sufficient condition for the complete removal of a class. The third lemma uses the second one to obtain an (exponential) upper bound on the maximal number of classes in a minimal model. Therefore, combined together, these lemmas limit the cardinality of a minimal model to a size exponential in the length of the formula. Moreover, the polynomial reductions from/to PNL to/from $\text{FO}^2[<]$ can be easily extended to PNL \sim and $\text{FO}^2[<, \sim]$, thus proving decidability and NEXPTIME-completeness of $\text{FO}^2[<, \sim]$.

To start with, we introduce the concept of *close friends* of a point x in a LIS \mathbf{L} , which consists of a minimal set of points (in \mathbf{L}) that satisfy all requests of x . For the sake of readability, we distinguish between future and past close friends of a point x .

Let φ be a PNL \sim formula and $\mathbf{L} = (\mathbb{D}, \mathcal{L})$ be a fulfilling LIS that satisfies φ . For all $x \in D$, the set of *future close friends* of x , denoted by $\mathcal{C}f|_A(x)$, is a minimal subset of D such that, for each $\langle A \rangle \psi \in \mathbf{req}|_A(x)$, there exists $x' \in \mathcal{C}f|_A(x)$ with $\psi \in \mathcal{L}([x, x'])$. Similarly, the set of *past close friends* of x , denoted by $\mathcal{C}f|_{\bar{A}}(x)$, is a minimal subset of D such that, for each $\langle \bar{A} \rangle \psi \in \mathbf{req}|_{\bar{A}}(x)$, there exists $x' \in \mathcal{C}f|_{\bar{A}}(x)$ with $\psi \in \mathcal{L}([x', x])$. We define the set of *close friends* of x to be the set $\mathcal{C}f(x) = \mathcal{C}f|_A(x) \cup \mathcal{C}f|_{\bar{A}}(x)$. It clearly holds that $|\mathcal{C}f(x)| \leq |\varphi| - 1$.

The following contraction lemma provides an upper bound to the size of equivalence classes.

Lemma 1 *Let φ be a PNL \sim formula and $\mathbf{L} = (\mathbb{D}, \mathcal{L})$ be a fulfilling LIS that satisfies it. If there exists a point x such that $|[x]_{\sim}| > 2 \cdot (|\varphi|^2 - 1) \cdot 2^{|\varphi|-1}$, then there exists a fulfilling LIS $\mathbf{L}' = (\mathbb{D}', \mathcal{L}')$ that satisfies φ with $|D'| = |D| - 1$.*

Proof Let C be $2 \cdot (|\varphi|^2 - 1)$ and let $x \in D$ be such that $|[x]_{\sim}| > C \cdot 2^{|\varphi|-1}$. Since there are at most $2^{|\varphi|-1}$ different sets of requests², then there exist $C + 1$ points $\hat{x}_0 < \dots < \hat{x}_C$ in $[x]_{\sim}$ with the same set of requests, that is, $\mathbf{req}(\hat{x}_i) = \mathbf{req}(\hat{x}_{i+1})$ for each $0 \leq i < C$. Now, let $\hat{x} = \hat{x}_{C/2}$ and let $B_{\sim}^{\mathcal{L}}(\hat{x})$ and $B_{\sim}^{\mathcal{L}'}(\hat{x})$ be respectively the sets $\{\hat{x}_0, \dots, \hat{x}_{C/2-1}\}$ and $\{\hat{x}_{C/2+1}, \dots, \hat{x}_C\}$. We define the sets $Blocked(\hat{x}) = \mathcal{C}f(\hat{x}) \cup \bigcup_{x' \in \mathcal{C}f(\hat{x})} \mathcal{C}f(x')$, $Free_{\sim}^{\mathcal{L}}(\hat{x}) = B_{\sim}^{\mathcal{L}}(\hat{x}) \setminus Blocked(\hat{x})$, and $Free_{\sim}^{\mathcal{L}'}(\hat{x}) = B_{\sim}^{\mathcal{L}'}(\hat{x}) \setminus Blocked(\hat{x})$. By a cardinality argument, both $|Free_{\sim}^{\mathcal{L}}(\hat{x})|$ and $|Free_{\sim}^{\mathcal{L}'}(\hat{x})|$ are greater than or equal to $C/2 - ((|\varphi| - 1)^2 + |\varphi| - 1) = (|\varphi|^2 - 1) - (|\varphi|^2 - |\varphi|) = |\varphi| - 1$.

We now show how to build a fulfilling LIS $\mathbf{L}' = (\mathbb{D}', \mathcal{L}')$, whose domain D' is equal to $D \setminus \{\hat{x}\}$, that still satisfies φ . First, we observe that there may exist $y < \hat{x}$ and $\langle A \rangle \psi \in \mathbf{req}|_A(y)$ such that $\psi \in \mathcal{L}([y, \hat{x}])$ and $\psi \notin \mathcal{L}([y, \bar{x}])$ for all $\bar{x} \neq \hat{x}$, and,

² Unlike the case of MPNL \sim , the closure of a PNL \sim formula φ only consists of the set of all sub-formulas of φ and their negations.

similarly, there may exist $y > \hat{x}$ and $\langle \bar{A} \rangle \psi \in \mathbf{req}|_{\bar{A}}(y)$ such that $\psi \in \mathcal{L}([\hat{x}, y])$ and $\psi \notin \mathcal{L}([\bar{x}, y])$ for all $\bar{x} \neq \hat{x}$. In either case, the removal of \hat{x} causes the defect of an unsatisfied request. To fix this, in the following we show how to suitably exploit the points in $Free^{\mathcal{L}}(\hat{x})$ (resp., $Free^{\mathcal{R}}(\hat{x})$) to solve the problems possibly caused by the removal of \hat{x} . The idea is to properly define the labeling \mathcal{L}' , that we initially set equal to \mathcal{L} , in such a way that defects are eliminated and no new defect is introduced.

First, we force all points in the future of \hat{x} , belonging to the same equivalence class as \hat{x} , which have not been blocked, to satisfy their past requests in the same way as \hat{x} does (that is, with the same past points). Similarly, we force all points in the past of \hat{x} , belonging to the same equivalence class as \hat{x} , which have not been blocked, to satisfy their future requests in the same way as \hat{x} does. Formally, for all $x \in Free^{\mathcal{L}}(\hat{x})$ and $x' \in \mathcal{C}f|_{\bar{A}}(\hat{x})$, we put $\mathcal{L}'([x', x]) = \mathcal{L}([x', \hat{x}])$; similarly, for all $x \in Free^{\mathcal{R}}(\hat{x})$ and $x' \in \mathcal{C}f|_A(\hat{x})$, we put $\mathcal{L}'([x, x']) = \mathcal{L}([\hat{x}, x'])$. Such a relabeling guarantees that all points in the set $\mathcal{C}f(\hat{x}) \cup Free^{\mathcal{L}}(\hat{x}) \cup Free^{\mathcal{R}}(\hat{x})$ are fulfilling.

Next, we show how to use the elements of $Free^{\mathcal{L}}(\hat{x}) \cup Free^{\mathcal{R}}(\hat{x})$ to fix the defects of the remaining points, if any (recall that both $Free^{\mathcal{L}}(\hat{x})$ and $Free^{\mathcal{R}}(\hat{x})$ contains at least $|\varphi| - 1$ elements). For all $x \in D \setminus (\mathcal{C}f(\hat{x}) \cup Free^{\mathcal{L}}(\hat{x}) \cup Free^{\mathcal{R}}(\hat{x}))$, we must distinguish three cases. Let $\hat{x} \in \mathcal{C}f|_A(x)$. Since $\hat{x} \notin Free^{\mathcal{L}}(\hat{x})$, $|Free^{\mathcal{L}}(\hat{x}) \setminus \mathcal{C}f(x)| \geq 1$ and thus there exists at least one free element $x' \in Free^{\mathcal{L}}(\hat{x})$ such that the relabeling of the interval (x, x') does not introduce any new defect (x' has the same close friends as \hat{x} and x is not one of them). Hence, we put $\mathcal{L}'([x, x']) = \mathcal{L}'([\hat{x}, x'])$. The case $\hat{x} \in \mathcal{C}f|_{\bar{A}}(x)$ can be dealt with in a completely symmetric way. In case $\hat{x} \notin \mathcal{C}f(x)$, we leave the labeling unchanged. \square

The next lemma gives a sufficient condition for the removal of an equivalence class. To prove it, we introduce a finer classification of the requests of a point that makes it possible to distinguish between those which are satisfied by intervals whose endpoints belong to the same equivalence class and those which are satisfied by intervals whose endpoints belong to different classes.

We denote by $\mathbf{req}|_{\bar{A}}^{\sim}(x)$ the subset of future requests of a point x which are satisfied over an interval $[x, y]$, with $x \sim y$, and by $\mathbf{req}|_{\bar{A}}^{\not\sim}(x)$ the subset of future requests of x which are satisfied over an interval $[x, y]$, with $x \not\sim y$. The two sets are not necessarily disjoint as they may share requests of the form $\langle A \rangle \psi$. We define in a similar way the subsets $\mathbf{req}|_{\bar{A}}^{\sim}(x)$ and $\mathbf{req}|_{\bar{A}}^{\not\sim}(x)$. Moreover, we define the sets $\mathbf{req}|^{\sim}(x) = \mathbf{req}|_{\bar{A}}^{\sim}(x) \cup \mathbf{req}|_A^{\sim}(x)$ and $\mathbf{req}|^{\not\sim}(x) = \mathbf{req}|_{\bar{A}}^{\not\sim}(x) \cup \mathbf{req}|_A^{\not\sim}(x)$. We denote the (structured) set of requests of a point x by $\mathbf{req}^S(x) = (\mathbf{req}|^{\sim}(x), \mathbf{req}|^{\not\sim}(x))$. With a little abuse of notation, in the following we will use $\mathbf{req}(x)$ for $\mathbf{req}^S(x)$.

Let \mathcal{R} be the set of all possible (structured) sets of requests. It can be easily checked that $|\mathcal{R}|$ is now bounded by $4^{|\varphi|-1}$. In addition, we denote by $\mathcal{C}f_{\sim}(x)$ (resp., $\mathcal{C}f_{\not\sim}(x)$) the subset of $\mathcal{C}f(x)$ consisting of all and only those elements in $\mathcal{C}f(x)$ that fulfill some request in $\mathbf{req}|^{\sim}(x)$ (resp., $\mathbf{req}|^{\not\sim}(x)$). It trivially holds that, for any x , the two sets $\mathcal{C}f_{\sim}(x)$ and $\mathcal{C}f_{\not\sim}(x)$ are disjoint.

Finally, we say that an equivalence class $[x]_{\sim}$, for some $x \in D$, is *covered* if for each $x' \in [x]_{\sim}$, there exist $2M$ points $x_1 < \dots < x_M < x' < x_{M+1} < \dots < x_{2M}$ such that (i) $x_i \not\sim x'$ and $\mathbf{req}(x_i) = \mathbf{req}(x')$, for all $1 \leq i \leq 2M$, and (ii) $x_i \not\sim x_j$, for all $1 \leq i < j \leq 2M$, where $M = |\varphi|^2 + |\varphi| - 1$ (we concisely say that points x_1, \dots, x_{2M} cover x').

Lemma 2 *Let φ be a PNL \sim formula and $\mathbf{L} = (\mathbb{D}, \mathcal{L})$ be a fulfilling LIS that satisfies it. If there is $x \in D$ such that $[x]_{\sim}$ is covered, then there exists a fulfilling LIS $\mathbf{L}' = (\mathbb{D}', \mathcal{L}')$, that satisfies φ , with $D' = D \setminus [x]_{\sim}$.*

Proof We show how to obtain \mathbf{L}' as the output of an iterative procedure that takes $\mathbf{L}_0 = \mathbf{L}$ as input and, at the i -th iteration, replaces $\mathbf{L}_{i-1} = (\mathbb{D}_{i-1}, \mathcal{L}_{i-1})$ by $\mathbf{L}_i = (\mathbb{D}_i, \mathcal{L}_i)$, with $D_i = D_{i-1} \setminus \{x'\}$ for some $x' \in [x]_{\sim} \cap D_{i-1}$. Notice that, while the initial LIS $\mathbf{L}_0 = \mathbf{L}$ is fulfilling, this is not necessarily the case with the subsequent ones. For any LIS \mathbf{L}_i , we only require the following conditions (invariant of the iterative procedure) to be satisfied: (1) each $x' \in D_i \setminus [x]_{\sim}$ is fulfilled in \mathbf{L}_i , (2) $[x]_{\sim} \cap D_i$ is covered, and (3) for each $x' \in [x]_{\sim} \cap D_i$ and each $\langle A \rangle \psi \in \mathbf{req}_A^{\neq}(x')$ (resp., $\overline{A}\psi \in \mathbf{req}_A^{\neq}(x')$), there exists $x'' \in D_i$ such that $\psi \in \mathcal{L}_i([x', x''])$ (resp., $\psi \in \mathcal{L}_i([x'', x'])$).

The invariant trivially holds for \mathbf{L}_0 . We now prove that if it holds for the LIS \mathbf{L}_i , then we can build a new LIS \mathbf{L}_{i+1} , which is obtained from \mathbf{L}_i by removing a point $\hat{x} \in [x]_{\sim} \cap D_i$, that satisfies it as well.

By item (2) of the invariant, there exist $2M$ points x_1, \dots, x_{2M} , with $x_1 < \dots < x_M < \hat{x} < x_{M+1} < \dots < x_{2M}$, that cover \hat{x} (recall that $M = |\varphi|^2 + |\varphi| - 1$). We denote by $B_{\neq}^p(\hat{x})$, $B_{\neq}^f(\hat{x})$, and $B_{\neq}(\hat{x})$ the sets $\{x_1, \dots, x_M\}$, $\{x_{M+1}, \dots, x_{2M}\}$, and $B_{\neq}^p(\hat{x}) \cup B_{\neq}^f(\hat{x})$, respectively.

Let $Blocked(\hat{x})$ be the set $\mathcal{C}f(\hat{x}) \cup \bigcup_{x' \in \mathcal{C}f(\hat{x})} \mathcal{C}f(x') \cup \{x \in B_{\neq}(\hat{x}) : \exists x' \in \mathcal{C}f(\hat{x})(x' \sim x)\}$. This set of blocked points differs from that of Lemma 1 for the inclusion of those points in $B_{\neq}(\hat{x})$ that belong to the same equivalence class of some point in $\mathcal{C}f(\hat{x})$. Since all points in $B_{\neq}(\hat{x})$ belong to different equivalence classes, the last argument of the set union contributes at most $|\mathcal{C}f(\hat{x})| = |\varphi| - 1$ additional points to the set of blocked points. Hence, $Blocked(\hat{x})$ consists of at most $|\varphi|^2 - 1$ points. Moreover, in analogy with Lemma 1, we define the sets $Free_{\neq}^p(\hat{x}) = B_{\neq}^p(\hat{x}) \setminus Blocked(\hat{x})$ and $Free_{\neq}^f(\hat{x}) = B_{\neq}^f(\hat{x}) \setminus Blocked(\hat{x})$. Since $|B_{\neq}^p(\hat{x})| = |B_{\neq}^f(\hat{x})| = |\varphi|^2 + |\varphi| - 1$, both $|Free_{\neq}^p(\hat{x})|$ and $|Free_{\neq}^f(\hat{x})|$ are greater than or equal to $|\varphi|$.

Let $Free_{\neq}(\hat{x}) = Free_{\neq}^p(\hat{x}) \cup Free_{\neq}^f(\hat{x})$. By construction, each element belonging to $Free_{\neq}(\hat{x})$ is not \sim -related to any element in $\mathcal{C}f(\hat{x})$. We define the labeling \mathcal{L}_{i+1} as follows.

First, for each $x' \in Free_{\neq}^p(\hat{x})$ and each $x'' \in \mathcal{C}f|_A(\hat{x})$, we have that $x' < \hat{x} < x''$ and we put $\mathcal{L}_{i+1}([x', x'']) = \mathcal{L}_i([\hat{x}, x''])$ ($= \mathcal{L}([\hat{x}, x''])$). Similarly, for each $x' \in Free_{\neq}^f(\hat{x})$ and each $x'' \in \mathcal{C}f|_{\overline{A}}(\hat{x})$, we have that $x'' < \hat{x} < x'$ and we put $\mathcal{L}_{i+1}([x'', x']) = \mathcal{L}_i([x'', \hat{x}])$ ($= \mathcal{L}([x'', \hat{x}])$). Notice that in this way we fixed all problems possibly introduced by the removal of \hat{x} for the points in $Free_{\neq}(\hat{x})$ and $\mathcal{C}f(\hat{x})$.

Then, for all $x' \in D_i \setminus (Free_{\neq}(\hat{x}) \cup \mathcal{C}f_{\neq}(\hat{x}) \cup [x]_{\sim})$, we distinguish three cases³. Case (i): $\hat{x} \in \mathcal{C}f|_A(x')$. Since $|\mathcal{C}f(x')| \leq |\varphi| - 1$, $|Free_{\neq}^f(\hat{x})| \geq |\varphi|$, and $\hat{x} \notin Free_{\neq}^f(\hat{x})$, $|Free_{\neq}^f(\hat{x}) \setminus \mathcal{C}f(x')| \geq 2$ and thus there are (at least) two distinct points $y, z \in Free_{\neq}^f(\hat{x})$ that do not belong to $\mathcal{C}f(x')$. By definition of $Free_{\neq}^f(\hat{x})$, $y \not\sim z$; it immediately follows that at least one of them, say y , does not belong to $[x']_{\sim}$. We put $\mathcal{L}_{i+1}([x', y]) = \mathcal{L}_i([x', \hat{x}])$. Case (ii): $\hat{x} \in \mathcal{C}f|_{\overline{A}}(x')$. This case is completely symmetric to the previous one, and thus its analysis is omitted. Case (iii): $\hat{x} \notin \mathcal{C}f(x')$. The labeling of all intervals featuring x' as one of their endpoints remains unchanged.

³ It is worth remarking that $\mathcal{C}f_{\sim}(\hat{x}) \subseteq [x]_{\sim}$.

To complete the proof, we need to show that the invariant is preserved in L_{i+1} . As for item (1), we first observe that the labeling of intervals with \sim -related endpoints does not change. Moreover, for all $x' \in Free_{\neq}^f(\hat{x})$, we fulfill all requests in $\mathbf{req}_A^{\neq}(x')$ by forcing x' to satisfy them as, by item (3), \hat{x} does (in L_i). The same for $x' \in Free_{\neq}^p(\hat{x})$. Finally, for all $x' \notin Free_{\neq}(\hat{x}) \cup Cf_{\neq}(\hat{x}) \cup [x]_{\sim}$ such that $\hat{x} \in Cf_{\neq}(x')$, we make use of points in $Free_{\neq}(\hat{x})$ to fix the defects introduced by the removal of \hat{x} . Item (2) trivially holds in L_{i+1} as the procedure does not change the \sim -relation between pairs of points in D_{i+1} . Similarly, the procedure does not change the labeling of the intervals with at least one endpoint in $(D_i \cap [x]_{\sim}) \setminus \{\hat{x}\}$, and thus item (3) holds.

The procedure terminates after $\lfloor [x]_{\sim} \rfloor$ steps when all points in $[x]_{\sim}$ have been removed. From the invariant, it follows that the resulting LIS is fulfilling (and it satisfies φ). \square

The next lemma provides an upper bound to the number of equivalence classes. Its proof exploits Lemma 2.

Lemma 3 *Let φ be a PNL \sim formula and $\mathbf{L} = (\mathbb{D}, \mathcal{L})$ be a fulfilling LIS for it. If there are more than $T = 2 \cdot M \cdot |\mathcal{R}_{\varphi}|$ equivalence classes in \mathbf{L} , with $M = |\varphi|^2 + |\varphi| - 1$, then there is a fulfilling LIS $\mathbf{L}' = (\mathbb{D}', \mathcal{L}')$ for φ with a strictly smaller number of classes.*

Proof We show how to select a non-empty subset of \sim -classes that can be safely removed. We take into consideration one $R \in \mathcal{R}_{\varphi}$ at a time. More precisely, we start by letting $S_0 = D$, and, at the i -th step, we select a new $R \in \mathcal{R}_{\varphi}$ and we use it to replace S_{i-1} by S_i , where S_i is obtained from S_{i-1} by removing the elements of (at most) $2 \cdot M$ selected \sim -classes.

Let $A_R = \bigcup_{j=1}^M [x_j^R]_{\sim}$ be the union of the \sim -classes of the M leftmost elements in S_{i-1} such that $x_j^R \not\sim x_k^R$ and $\mathbf{req}(x_j^R) = R$ for all $1 \leq j \neq k \leq M$. Formally, each x_j^R is recursively defined as follows (for $1 < j \leq M$):

$$\begin{aligned} x_1^R &:= \min\{x \in S_{j-1} : \mathbf{req}(x) = R\}, \\ x_j^R &:= \min\{x \in S_{j-1} : \mathbf{req}(x) = R \wedge \forall k < j (x \not\sim x_k^R)\} \end{aligned}$$

Analogously, let $\bar{A}_R = \bigcup_{j=1}^M [\bar{x}_j^R]_{\sim}$, where $\bar{x}_1^R, \dots, \bar{x}_M^R$ are the M rightmost elements in $S_{i-1} \setminus A_R$ such that $\bar{x}_j^R \not\sim \bar{x}_k^R$ and $\mathbf{req}(\bar{x}_j^R) = R$ for all $1 \leq j \neq k \leq M$, that is (for $1 < j \leq M$):

$$\begin{aligned} \bar{x}_1^R &:= \max\{x \in S_{i-1} \setminus A_R : \mathbf{req}(x) = R\}, \\ \bar{x}_j^R &:= \max\{x \in S_{i-1} \setminus A_R : \mathbf{req}(x) = R \wedge \forall k < j (x \not\sim \bar{x}_k^R)\}. \end{aligned}$$

The condition $\bar{x}_j^R \notin A_R$ is introduced to avoid class collisions. It is worth noticing that some x_j^R or \bar{x}_j^R may be empty (this is the case, for instance, if there are fewer than $2 \cdot M$ classes with elements with requests R). We define S_i as $S_{i-1} \setminus (A_R \cup \bar{A}_R)$.

To conclude the proof, it suffices to show that all \sim -classes included in $S_{|\mathcal{R}_{\varphi}|}$ (the set obtained from the above iterative process) satisfy the conditions of Lemma 2 and thus they can be safely removed. Let C be one such class. By construction, each element $x \in C$ has M elements $x_1^{\mathbf{req}(x)}, \dots, x_M^{\mathbf{req}(x)} \in D \setminus S_{|\mathcal{R}_{\varphi}|}$ before it and M elements $\bar{x}_1^{\mathbf{req}(x)}, \dots, \bar{x}_k^{\mathbf{req}(x)} \in D \setminus S_{|\mathcal{R}_{\varphi}|}$ after it such that (i) they belong to different \sim -classes (all different from $[x]_{\sim}$) and (ii) they feature the same requests as x . \square

Theorem 2 *The satisfiability problem for $\text{PNL}\sim$ over the class of finite linear orders is NEXPTIME-complete.*

Proof Decidability of finite satisfiability for $\text{PNL}\sim$ immediately follows from the above lemmas, that provide an exponential upper bound to both the number of \sim -classes and the number of their elements. NEXPTIME-hardness follows from that of PNL [6]. \square

We conclude the section by pointing out that decidability (and NEXPTIME-completeness) of the satisfiability problem for $\text{FO}^2[<, \sim]$ over finite linear orders directly follows from Theorem 2, as the polynomial reductions from PNL to $\text{FO}^2[<]$ and from $\text{FO}^2[<]$ to PNL can be easily extended to $\text{PNL}\sim$ and $\text{FO}^2[<, \sim]$ maintaining the same (polynomial) complexity.

4 Adding an equivalence relation to MPNL: $\text{MPNL}\sim$

In this section, we show how to reduce the satisfiability problem for $\text{MPNL}\sim$ over the class of finite linear orders to the decidable 0-0 reachability problem for vector addition systems (VASS). The opposite reduction comes for free from the reduction of the 0-0 reachability problem for VASS to the satisfiability problem for $\text{FO}^2(\sim, <, +1)$ over finite data words [2], since the latter is embeddable in $\text{MPNL}\sim$.

Let φ be an $\text{MPNL}\sim$ formula to be checked for satisfiability. In the following, we build a VASS A_φ such that the 0-0 reachability problem for A_φ can be positively solved if and only if φ is satisfiable over the class of finite linear orders. As we shall see, the possibility of constraining the length of intervals to equal 1, provided by $\text{MPNL}\sim$ by means of the metric constraint len_1 , makes it quite difficult to give a bound on the size of each equivalence classes as well as on their overall number. In Section 3, we solved these two problems for $\text{PNL}\sim$ in Lemma 1 and Lemma 3, respectively. We shall show that the latter problem, namely, the problem of constraining the number of equivalence classes, can be dealt with by exploiting the counting feature of VASS. However, as a preliminary to that, we provide a characterization of each equivalence class in terms of the behaviour of a limited number of its elements (the exact number depends on the size of the formula). Such a characterization will be obtained via a re-labelling of intervals in a fulfilling LIS (that satisfies φ) that allows us to replace an arbitrary fulfilling LIS by a more regular one. The following definitions and results go in that direction.

Let $\mathbf{L} = (\mathbb{D}, \mathcal{L})$ be a LIS and let x be a point in D . We define the *class-position* of x , denoted $\text{pos}_\sim(x)$, as the number of distinct points $x' \in [x]_\sim$ such that $x' < x$ and $\text{req}(x') = \text{req}(x)$. Moreover, given an equivalence class $[x]_\sim$ and a set of requests R , we denote by $\max(R, [x]_\sim)$ the point $y \in [x]_\sim$ with $\text{req}(y) = R$ (if any) such that, for all points $z \in [x]_\sim$, if $\text{req}(z) = \text{req}(y)$, then $z \leq y$. Now, let $\mathbf{L} = (\mathbb{D}, \mathcal{L})$ be a fulfilling LIS that satisfies an $\text{MPNL}\sim$ formula φ . We define the set $\text{req}_{>1, A}(x)$ as the set of all and only those formulas $\langle A \rangle \psi$ for which there exists y such that $y > x$, $\psi \in \mathcal{L}([x, y])$, and $\text{len}_1 \notin \mathcal{L}([x, y])$. The set $\text{req}_{>1, \bar{A}}(x)$ is defined analogously. A *far friend* mapping $\text{ff}_{\mathbf{L}} : D \rightarrow 2^D$ is a function that maps each element $x \in D$ to a minimal set $D' \subseteq D$ such that (i) for each $\langle A \rangle \psi \in \text{req}_{>1, A}(x)$, there exists $y \in \text{ff}_{\mathbf{L}}(x)$ such that $y > x$, $\psi \in \mathcal{L}([x, y])$, and $\text{len}_1 \notin \mathcal{L}([x, y])$ and (ii) for each $\langle \bar{A} \rangle \psi \in \text{req}_{>1, \bar{A}}(x)$, there exists $y \in \text{ff}_{\mathbf{L}}(x)$ such that $y < x$, $\psi \in \mathcal{L}([y, x])$, and

$len_1 \notin \mathcal{L}([y, x])$. For all $x \in D$, we define the natural restrictions $ff_{\mathbf{L}}^A(x)$ and $ff_{\mathbf{L}}^{\bar{A}}(x)$ of $ff_{\mathbf{L}}(x)$ to the points in the future and in the past of x , respectively, that is, $ff_{\mathbf{L}}^A(x) = \{y \in ff_{\mathbf{L}}(x) : y > x\}$ and $ff_{\mathbf{L}}^{\bar{A}}(x) = \{y \in ff_{\mathbf{L}}(x) : y < x\}$.

Lemma 4 *Let φ be a satisfiable MPNL \sim formula. Then, there is a fulfilling LIS $\mathbf{L} = (\mathbb{D}, \mathcal{L})$ for φ and a far friend function $ff_{\mathbf{L}}$ such that, for all $x \in D$, it holds that (i) for each $y \in ff_{\mathbf{L}}^{\bar{A}}(x)$, $pos_{\sim}(y) \leq |\varphi|^2 - |\varphi|$ and (ii) if $|\varphi|^2 - |\varphi| < pos_{\sim}(x)$, then $ff_{\mathbf{L}}^A(x) = ff_{\mathbf{L}}^A(\max(\mathbf{req}(x), [x]_{\sim}))$.*

Proof The proof is done in two steps. First, given a fulfilling LIS $\mathbf{L} = (\mathbb{D}, \mathcal{L})$ for φ and a far friend function $ff_{\mathbf{L}}$, we build a fulfilling LIS \mathbf{L}' (for φ) and a far friend function $ff_{\mathbf{L}'}$ satisfying (i); then, we turn \mathbf{L}' and $ff_{\mathbf{L}'}$ into a fulfilling LIS \mathbf{L}'' (for φ) and a far friend function $ff_{\mathbf{L}''}$ that satisfy both (i) and (ii).

Let $M = |\varphi|^2 - |\varphi|$. The number k of ‘defects’ to fix in \mathbf{L} in order to satisfy condition (i) is equal to $|\{(x, x') : x \in ff_{\mathbf{L}}^{\bar{A}}(x') \wedge pos_{\sim}(x) > M\}|$. We build \mathbf{L}' by a finite number of iterations of a basic procedure, which starts from $\mathbf{L}_0 = (\mathbb{D}, \mathcal{L}_0) = (\mathbb{D}, \mathcal{L})$, $ff_{\mathbf{L}_0} = ff_{\mathbf{L}}$, and $k_0 = k$ (if $k_0 = 0$, we simply put $\mathbf{L} = \mathbf{L}'$ and $ff_{\mathbf{L}'} = ff_{\mathbf{L}}$, and we move to the next step).

Let $\mathbf{L}_i = (\mathbb{D}, \mathcal{L}_i)$ be the output of i -th iteration of the procedure and let $k_i = |\{(x, x') : x \in ff_{\mathbf{L}_i}^{\bar{A}}(x') \wedge pos_{\sim}(x) > M\}|$. We show how to obtain a fulfilling LIS $\mathbf{L}_{i+1} = (\mathbb{D}, \mathcal{L}_{i+1})$, that satisfies φ , and a far friend function $ff_{\mathbf{L}_{i+1}}$ with $k_{i+1} < k_i$. Let $x \in ff_{\mathbf{L}_i}^{\bar{A}}(x')$, for some $x' \in D$, with $pos_{\sim}(x) > M$, and let $\bar{D} = \{\bar{x}_1, \dots, \bar{x}_M\}$ be the set of the first M points in $[x]_{\sim}$ with $\mathbf{req}(\bar{x}_j) = \mathbf{req}(x)$ (clearly, $\bar{x}_j < x$ for all $1 \leq j \leq M$). As the cardinality of the set $\bar{D}' = \bigcup_{x'' \in ff_{\mathbf{L}_i}^A(x)} ff_{\mathbf{L}_i}^{\bar{A}}(x'')$ is bounded by $(|\varphi| - 1)^2 = M - (|\varphi| - 1)$, there are at least $|\varphi| - 1$ distinct points $\bar{x}_{j_1}, \dots, \bar{x}_{j_{|\varphi|-1}}$ in $\bar{D} \setminus \bar{D}'$. Since $x \in ff_{\mathbf{L}_i}(x')$ (and $x \notin \bar{D}$), there is at least one point \bar{x}_{j_k} in $(\bar{D} \setminus \bar{D}') \setminus ff_{\mathbf{L}_i}(x')$. We put $\mathcal{L}_{i+1}([\bar{x}_{j_k}, x']) = \mathcal{L}_i([x, x'])$, $\mathcal{L}_{i+1}([\bar{x}_{j_k}, x'']) = \mathcal{L}_i([x, x''])$ for each $x'' \in ff_{\mathbf{L}_i}^A(x)$, and $ff_{\mathbf{L}_{i+1}}(x') = (ff_{\mathbf{L}_i}(x') \setminus \{x\}) \cup \{\bar{x}_{j_k}\}$. The labeling function \mathcal{L}_{i+1} and the far friend function $ff_{\mathbf{L}_{i+1}}$ over the other intervals and points, respectively, are the same as those given at the i -th step. After $k' \leq k$ steps (at some iteration, it may happen that we solve more than one defect), we obtain a pair $(\mathbf{L}_{k'}, ff_{\mathbf{L}_{k'}})$ that satisfies condition (i).

The construction of a fulfilling LIS \mathbf{L}'' and a far friend function $ff_{\mathbf{L}''}$ that satisfy both (i) and (ii) is pretty much the same, apart from the fact that it requires to start from a fulfilling LIS for φ that satisfies (i). We begin by letting $\mathbf{L}_0 = \mathbf{L}'$, which satisfies condition (i). Next, we show that, for each $n > 0$, the fulfilling LIS \mathbf{L}_n generated by the n -th iteration of the procedure still satisfies (i). Let \mathbf{L}_i and $ff_{\mathbf{L}_i}$ be respectively the LIS and the far friend function generated at the i -th iteration of the procedure. By the inductive hypothesis, they satisfy condition (i). If they satisfy also condition (ii), then we put $\mathbf{L}'' = \mathbf{L}_i$ and $ff_{\mathbf{L}''} = ff_{\mathbf{L}_i}$, and we stop. Otherwise, there exists $x \in D$ such that $pos_{\sim}(x) > M$ and $ff_{\mathbf{L}_i}(x) \neq ff_{\mathbf{L}_i}(\max(\mathbf{req}(x), [x]_{\sim}))$, which implies that $x < \max(\mathbf{req}(x), [x]_{\sim})$. Let $x' = \max(\mathbf{req}(x), [x]_{\sim})$. By the inductive hypothesis (\mathbf{L}_i and $ff_{\mathbf{L}_i}$ satisfy (i)), $x \notin \bigcup_{x'' \in ff_{\mathbf{L}_i}^A(x')} ff_{\mathbf{L}_i}^{\bar{A}}(x'')$. We put $\mathcal{L}_{i+1}([x, x'']) = \mathcal{L}_i([x', x''])$ for each $x'' \in ff_{\mathbf{L}_i}^A(x')$, $ff_{\mathbf{L}_{i+1}}(x) = ff_{\mathbf{L}_i}^{\bar{A}}(x) \cup ff_{\mathbf{L}_i}^A(x')$. The labeling function \mathcal{L}_{i+1} and the far friend function $ff_{\mathbf{L}_{i+1}}$ over the other intervals and points, respectively, are the same as those given at the i -th step. It can be easily checked that \mathbf{L}_{i+1} is a fulfilling LIS for φ and \mathbf{L}_{i+1} and $ff_{\mathbf{L}_{i+1}}$ still satisfy

condition (i). Moreover, the number of points in D that do not satisfy condition (ii) in \mathbf{L}_{i+1} is strictly less than the number of points that do not satisfy it in \mathbf{L}_i . The procedure terminates after a number $h' \leq h$ of steps, where $h = |\{x : \text{pos}_{\sim}(x) > M \wedge \text{ff}_{\mathbf{L}_0}^A(x) \neq \text{ff}_{\mathbf{L}_0}^A(\max(\mathbf{req}(x), [x]_{\sim}))\}|$. We put $\mathbf{L}'' = \mathbf{L}_{h'}$ and $\text{ff}_{\mathbf{L}''} = \text{ff}_{\mathbf{L}_{h'}}$. \square

Let \mathbf{L} and $\text{ff}_{\mathbf{L}}$ be respectively the fulfilling LIS for φ and the far friend function whose existence is guaranteed by Lemma 4. We call such a pair $(\mathbf{L}, \text{ff}_{\mathbf{L}})$ a *well-match*. We shall use the far friend function to keep track of the requests of any point $x \in D$ which are satisfied by points y at a distance greater than 1. Requests of x which are satisfied by points y at a distance less than or equal to 1 from x , that is, at the immediate predecessor or successor of x , indeed, turn out to be easy to check by means of a VASS. Lemma 4 states that for any fulfilling LIS \mathbf{L} , we can rearrange the intervals of \mathbf{L} in such a way that, for each point $x \in D$, each $\psi \in \mathbf{req}_{\overline{A}, > 1}(x)$ is satisfied at a point y with $\text{pos}_{\sim}(y) \leq |\varphi|^2 - |\varphi|$. Moreover, it states that, for each point $x \in D$, $\text{pos}_{\sim}(x) \leq |\varphi|^2 - |\varphi|$ or its future requests $\mathbf{req}_{A, > 1}(x)$ are dealt with as those of the maximum witness of $\mathbf{req}(x)$ in its class, i.e., by making use of the same points. It is worth pointing out that there is an asymmetry in the way we deal with past and future requests, which stems from the fact that the VASS will process a (candidate) model from the first point to the last one. Indeed, for each past request $\langle \overline{A} \rangle \psi$ of a point x , the VASS guarantees that it can be satisfied by the points already introduced, which amounts to saying that the whole set $\mathbf{req}_{\overline{A}}(x)$ can be readily checked for fulfillment when x is added to the model. On the other hand, fulfillment of $\langle A \rangle \psi$ requests of x is postponed as it can be checked only when all the points of the (candidate) model have been introduced.

We are now ready to provide a finite characterization of \sim -classes based on the notion of class word. A *class word* \tilde{w} is a finite word on the alphabet \mathcal{R}_{φ} such that, for each $R \in \mathcal{R}_{\varphi}$, $|\{i : \tilde{w}[i] = R\}| \leq |\varphi|^2 - |\varphi| + 1$. Let $\tilde{\mathcal{W}}_{\varphi}$ be the set of all class words on \mathcal{R}_{φ} . Given a fulfilling LIS $\mathbf{L} = (\mathbb{D}, \mathcal{L})$ for φ , we say that a point $y \in D$ is *meaningful* if $\text{pos}_{\sim}(y) \leq |\varphi|^2 - |\varphi|$ or $y = \max(\mathbf{req}(y), [y]_{\sim})$. For any class $[x]_{\sim}$, we define the class word $\tilde{w}_{[x]_{\sim}}$ to be the word $\mathbf{req}(\bar{x}_1), \dots, \mathbf{req}(\bar{x}_m)$, where $\bar{x}_1, \dots, \bar{x}_m$ is the sequence (according to the ordering relation $<$) of all and only the meaningful points in $[x]_{\sim}$. As it happens with atoms and sets of requests, class words are syntactic objects that may or may not occur in a fulfilling LIS \mathbf{L} for φ , that is, the set of class words witnessed by some class $[x]_{\sim}$ in some fulfilling LIS \mathbf{L} for φ is contained in $\tilde{\mathcal{W}}_{\varphi}$, but not vice versa.

As already observed, in any computation of a VASS, $\langle \overline{A} \rangle$ requests are checked immediately for fulfillment, while $\langle A \rangle$ requests can be postponed. In order to keep track of the $\langle A \rangle$ requests which have been already fulfilled, class words are paired with an additional word that stores information about these requests. Given a class word \tilde{w} , a *temporary fulfillment* of \tilde{w} is a word $t_{\tilde{w}} \in \mathcal{R}_{\varphi}^*$ such that (i) $|t_{\tilde{w}}| \leq |\tilde{w}|$, (ii) for each $1 \leq i \leq |t_{\tilde{w}}|$, $t_{\tilde{w}}[i] \subseteq \tilde{w}[i]_A$, and (iii) $\langle A \rangle \text{len}_1 \in t_{\tilde{w}}[i]$ for each $1 \leq i < |t_{\tilde{w}}|$. A temporary fulfillment of length m collects the $\langle A \rangle$ -requests of a class word up to position m that have been already fulfilled.

In the following, we will make use of the temporary fulfillment of a given class word up to a given point, which is defined as follows. Let $(\mathbf{L}, \text{ff}_{\mathbf{L}})$, where $\mathbf{L} = (\mathbb{D}, \mathcal{L})$, be a well-match, $[x]_{\sim}$ be an equivalence class, and y be a point in D . The temporary fulfillment of $\tilde{w}_{[x]_{\sim}}$ up to point y , denoted by $t_{\tilde{w}_{[x]_{\sim}}, y}$ is a sequence $R_1 (\subseteq \mathbf{req}_A(\bar{x}_1)) \dots R_{m'} (\subseteq \mathbf{req}_A(\bar{x}_{m'}))$, where $\bar{x}_1 \dots \bar{x}_{m'}$ is the sequence of meaningful

points in $[x]_{\sim}$ less than or equal to y and, for each $1 \leq i \leq m'$, $R_i = \{\psi \in \mathcal{L}([\bar{x}_i, z]) \cap \mathbf{req}_A(\bar{x}_i) : z < y \wedge (z \in \mathbf{ff}_{\mathbf{L}}(\bar{x}_i) \vee \forall z'(\bar{x}_i < z' \leq z \rightarrow z' = z))\}$. Intuitively, a temporary fulfilling $t_{\tilde{w}[x]_{\sim}, y}$ represents the snapshot of the fulfilments of the $\langle A \rangle$ -requests of the meaningful points in $[x]_{\sim}$ when some point y (not necessarily in $[x]_{\sim}$) is introduced.

Given a class word \tilde{w} , let $\mathcal{T}_{\tilde{w}}$ be the set of all possible temporary fulfilments of it. Moreover, let \mathcal{T} be the set union of $\mathcal{T}_{\tilde{w}}$ for all $\tilde{w} \in \widetilde{\mathcal{W}}_{\varphi}$. We say that a temporary fulfilment $t_{\tilde{w}}$ is *final* if and only if $|t_{\tilde{w}}| = |\tilde{w}|$ and for all $1 \leq i \leq |\tilde{w}|$, $t_{\tilde{w}}[i] = \{\langle A \rangle \psi \in \tilde{w}[i]\}$. Moreover, we say that the pair $(\tilde{w}, t_{\tilde{w}})$ is *final* if $t_{\tilde{w}}$ is final. We define a partial order \leq over $\mathcal{T}_{\tilde{w}}$ such that, for each $t_{\tilde{w}}, t'_{\tilde{w}}$ in $\mathcal{T}_{\tilde{w}}$, $t_{\tilde{w}} \leq t'_{\tilde{w}}$ if and only if $|t_{\tilde{w}}| = |t'_{\tilde{w}}|$ and, for each $1 \leq i \leq |t_{\tilde{w}}|$, $t_{\tilde{w}}[i] \subseteq t'_{\tilde{w}}[i]$.

The counters of the VASS will be indexed by the pairs $(\tilde{w}, t_{\tilde{w}})$, where \tilde{w} is a class word and $t_{\tilde{w}} \in \mathcal{T}_{\tilde{w}}$ is one of its possible temporary fulfilments. A rough upper bound to the number C of counters of the VASS is thus given by $2^{2 \cdot (|\varphi|+2) \cdot (|\varphi|^2 - |\varphi| + 1) \cdot 2^{|\varphi|+2}}$ (the length of the class word, and consequently that of the temporary fulfilment, is bounded by $(|\varphi|^2 - |\varphi| + 1) \cdot 2^{|\varphi|+2}$ and the number of different values that can occur at each position of the class word, and consequently of the temporary fulfilment, is bounded by $2^{|\varphi|+2}$).

Let x be the point introduced in the current state of the VASS and let R be the set of its requests. The class of x , the class of its immediate predecessor, and the classes of the points which are used to satisfy the past requests in $\mathbf{req}_{\overline{A}}(x)$, whose number is bounded by $|\varphi|$, are dealt with by the control of the automaton, as we shall show later. All the other classes are updated by making use of the following 3-argument relation:

$t_{\tilde{w}} \xrightarrow{R} t'_{\tilde{w}}$ if and only if (i) $t_{\tilde{w}} \leq t'_{\tilde{w}}$; (ii) there exist $m(= |t_{\tilde{w}}|)$ interval tuples $(\mathbf{req}(\tilde{w}[i]), F_i, R)$ such that for each $1 \leq i \leq m$, $t'_{\tilde{w}}[i] = t_{\tilde{w}}[i] \cup \{\langle A \rangle \psi \in t_{\tilde{w}}[i] : \psi \in F_i\}$ and $\text{len}_1, \sim \notin F_i$; (iii) there exists no a temporary fulfilment $t''_{\tilde{w}}$, with $t'_{\tilde{w}} < t''_{\tilde{w}}$, that satisfies both (i) and (ii).

Property (iii) guarantees a maximality condition with respect to the satisfied $\langle A \rangle$ -requests for meaningful points in the update of the temporary fulfilment. From the point of view of the model, one can always find a LIS that guarantees such a property in a pointwise manner.

Let \mathbb{D} be a finite linear order and $x \in D$, with $x \neq \min(\mathbb{D})$. We denote by $\text{pre}(x)$ the predecessor of x . We introduce now the notion of *saturated well-match*.

Definition 1 Let $(\mathbf{L}, \mathbf{ff}_{\mathbf{L}})$, with $\mathbf{L} = (\mathbb{D}, \mathcal{L})$, be a well-match. We say that $(\mathbf{L}, \mathbf{ff}_{\mathbf{L}})$ is *saturated* if and only if for each pair $x, x' \in D$, with $x' \notin \bigcup_{x'' \in \mathbf{ff}_{\mathbf{L}}^{\overline{A}}(x) \cup \{x, \text{pre}(x)\}} [x'']_{\sim}$ and $x' < x$, it holds that, for every well-match $(\mathbf{L}', \mathbf{ff}_{\mathbf{L}'})$, with $\mathbf{L}' = (\mathbb{D}, \mathcal{L}')$ and $\mathbf{ff}_{\mathbf{L}'}^{\overline{A}} = \mathbf{ff}_{\mathbf{L}}^{\overline{A}}$, such that $\mathcal{L}'([\bar{x}', \bar{x}]) = \mathcal{L}([\bar{x}', \bar{x}])$, with $\bar{x}' \neq x'$ or $\bar{x} \neq x$, it holds that $t_{\tilde{w}[x']_{\sim}, x} \prec t'_{\tilde{w}[x']_{\sim}, x}$ for the temporary fulfilments $t_{\tilde{w}[x']_{\sim}, x}$ in \mathbf{L} and $t'_{\tilde{w}[x']_{\sim}, x}$ in \mathbf{L}' .

The intuitive meaning of saturation is that for any pair of points x, x' , with $x' < x$, in a saturated well-match $(\mathbf{L}, \mathbf{ff}_{\mathbf{L}})$, if x' has no role in the fulfilling of the past requests of x , then the temporary fulfilment $t_{\tilde{w}[x']_{\sim}, x}$ cannot be improved, that is, it is not possible to increase the set of future requests that x' fulfills before the introduction of x by changing any labeling in \mathbf{L} under the constraint that $\mathbf{ff}_{\mathbf{L}}^{\overline{A}}$ remains the same.

The next lemma shows that we can safely restrict our attention to saturated well-matches.

Lemma 5 *Let φ be a satisfiable MPNL \sim formula and let $(\mathbf{L}, ff_{\mathbf{L}})$, with $\mathbf{L} = (\mathbb{D}, \mathcal{L})$, be a well-match for it. Then, there is a saturated well-match $(\mathbf{L}' = (\mathbb{D}, \mathcal{L}'), ff_{\mathbf{L}'})$ for it.*

Proof The proof is quite straightforward. Once more, it exploits an iterative procedure that builds a sequence of well-matches that progressively fixes the defects with respect to the saturation property. As we shall see, one of its distinctive features is that it does not change the function $ff_{\mathbf{L}}^{\bar{A}}$, that is, it enforces $ff_{\mathbf{L}_i}^{\bar{A}} = ff_{\mathbf{L}_{i+1}}^{\bar{A}}$ for all i , as it updates some labelings without affecting it.

Let $\mathbf{L}_0 = (\mathbb{D}, \mathcal{L}_0) = (\mathbb{D}, \mathcal{L})$. By hypothesis, the LIS \mathbf{L}_0 satisfies the conditions of Lemma 4. Consider now the well-match built at the i -th iteration. Let x, x' , with $x' < x$, be the minimum pair of elements in D (according to a lexicographical order on the pairs (x, x')) such that the saturation condition is violated in \mathbf{L}_i . Then, there exists a well-match $(\mathbf{L}'_i, ff_{\mathbf{L}'_i})$, with $\mathbf{L}'_i = (\mathbb{D}, \mathcal{L}'_i)$ and $ff_{\mathbf{L}'_i}^{\bar{A}} = ff_{\mathbf{L}_i}^{\bar{A}}$, such that for each pair \hat{x}', \hat{x} , with $\hat{x}' \neq x'$ or $\hat{x} \neq x$, it holds that $\mathcal{L}_i([\hat{x}', \hat{x}]) = \mathcal{L}'_i([\hat{x}', \hat{x}])$ and $t_{\tilde{w}_{[x']\sim, x}} < t'_{\tilde{w}_{[x']\sim, x}}$ for the temporary fulfillments $t_{\tilde{w}_{[x']\sim, x}}$ in \mathbf{L}_i and $t_{\tilde{w}'_{[x']\sim, x}}$ in \mathbf{L}'_i . Since temporary fulfillments are built only on meaningful points, it holds that $pos_{\sim}(x') \leq |\varphi|^2 - |\varphi|$ or $x' = \max(\mathbf{req}(x'), [x']\sim)$. Moreover, since $x' < x$ is a violation of the saturation property, it holds that $x' \notin \bigcup_{x'' \in ff_{\mathbf{L}}^{\bar{A}}(x) \cup \{x, pre(x)\}} [x'']\sim$. As a direct consequence of the latter condition, it holds that $\{\psi \in \mathbf{req}|_A(x') : \exists x'' \leq x, \psi \in \mathcal{L}_i([x', x''])\} \subset \{\psi \in \mathbf{req}|_A(x') : \exists x'' \leq x, \psi \in \mathcal{L}'_i([x', x''])\}$ and since $\mathcal{L}_i([x', x'']) = \mathcal{L}'_i([x', x''])$, for each $x'' \neq x$, it holds that $\mathcal{L}_i([x', x]) \cap \mathbf{req}|_A(x') \subset \mathcal{L}'_i([x', x]) \cap \mathbf{req}|_A(x')$.

Since $x' \notin ff_{\mathbf{L}}^{\bar{A}}(x)$, we can safely put $\mathcal{L}_{i+1}([x', x]) = \mathcal{L}'_i([x', x])$, leaving the labeling of all the other intervals and $ff_{\mathbf{L}}^{\bar{A}}$ unchanged. It is easy to check that $t_{\tilde{w}_{[x']\sim, x}} < t'_{\tilde{w}'_{[x']\sim, x}}$ holds for the temporary fulfillments $t_{\tilde{w}_{[x']\sim, x}}$ in \mathbf{L}_i and $t_{\tilde{w}'_{[x']\sim, x}}$ in \mathbf{L}_{i+1} .

To complete the proof, it suffices to observe that, for all $x' \in D$, at most $|\mathbf{req}|_A(x')| \leq |\varphi|$ pairs of points (x', x) , with $x' < x$, are taken into consideration by the procedure. Since at each iteration $i + 1$ the added point x is used to fulfill (in \mathbf{L}_{i+1}) at least one request $\psi \in \mathbf{req}|_A(x')$, with $x' < x$, which in \mathbf{L}_i was fulfilled with a point $x'' > x$, it immediately follows that after at most k steps, with $k \leq |\varphi| \cdot |D|$, we obtain a saturated LIS $\mathbf{L}' = \mathbf{L}_k$ for φ . \square

The saturation property will be exploited in the proof of the completeness of the reduction: starting from a saturated well-match $(\mathbf{L}, ff_{\mathbf{L}})$, with $\mathbf{L} = (\mathbb{D}, \mathcal{L})$, one can indeed prove the existence of a 0-0 computation for the VASS.

Let us now focus on the computation of the VASS. Whenever we introduce a (new occurrence of a) set of requests R , it may happen that the second component of a pair $(\tilde{w}, t_{\tilde{w}})$ needs to be updated. More precisely, for all pairs $(\tilde{w}, t_{\tilde{w}})$ such that there exists $t'_{\tilde{w}}$ with $t_{\tilde{w}} \xrightarrow{R} t'_{\tilde{w}}$, we force the VASS to simultaneously decrease the counter $c_{\tilde{w}, t_{\tilde{w}}}$ by 1 and to increase the counter $c_{\tilde{w}, t'_{\tilde{w}}}$ by 1 (value transfer). However, to guarantee that such an update does not violate the soundness of the construction, we must successfully cope with two non-trivial problems:

- (P1) Any given occurrence of a set R cannot be used more than one time, that is, when we transfer values from a counter $c_{\tilde{w}, t_{\tilde{w}}}$ to a counter $c_{\tilde{w}, t'_{\tilde{w}}}$, with $t_{\tilde{w}} \xrightarrow{R} t'_{\tilde{w}}$,

we must guarantee that all the transitions that transfer values from $c_{\tilde{w}, t'_{\tilde{w}}}$ to another counter $c_{\tilde{w}, t''_{\tilde{w}}}$ are not fired until a new set R' is introduced. If that were not the case, we would be using a single occurrence of a set R (associated with a given point x) to solve more requests than it can handle and this would compromise the soundness of the construction.

(P2) Whenever a transition performing a value transfer from a counter $c_{\tilde{w}, t_{\tilde{w}}}$ to a counter $c_{\tilde{w}, t'_{\tilde{w}}}$, with $t_{\tilde{w}} \xrightarrow{R} t'_{\tilde{w}}$, is fired, we can only conclude that $c_{\tilde{w}, t_{\tilde{w}}}$ was greater than zero at that time. The VASS, indeed, cannot test $c_{\tilde{w}, t_{\tilde{w}}}$ for being 0 and, if this is not the case, fire all possible transitions until it reaches 0 (this is the feature that makes reachability problems for such machines decidable). Then, in a computation, it may happen that $c_{\tilde{w}, t_{\tilde{w}}}$ remains greater than zero when we introduce a new request R' after R even in those cases in which it was possible to transfer the value of $c_{\tilde{w}, t_{\tilde{w}}}$ to some $c_{\tilde{w}, t'_{\tilde{w}}}$, with $t_{\tilde{w}} \xrightarrow{R} t'_{\tilde{w}}$. This does not compromise the fulfilling of $\langle A \rangle$ requests, as their fulfillment is simply postponed. However, if $c_{\tilde{w}, t_{\tilde{w}}}$ never changes its value, that is, no transition that executes the decrement of $c_{\tilde{w}, t_{\tilde{w}}}$ is ever fired, two explanations are possible: either $c_{\tilde{w}, t_{\tilde{w}}} = 0$ or $c_{\tilde{w}, t_{\tilde{w}}} > 0$ and $t_{\tilde{w}} \xrightarrow{R} t'_{\tilde{w}}$ for any $t'_{\tilde{w}}$. While the first case is perfectly acceptable in a computation, the second one is sufficient to cause the termination of the computation with failure (again, the problem arises because we cannot test whether or not $c_{\tilde{w}, t_{\tilde{w}}}$ is equal to 0).

Problem (P1) can be solved by using maximality of the ternary relation $t_{\tilde{w}} \xrightarrow{R} t'_{\tilde{w}}$, which guarantees that if we fix R , then the binary relation \xrightarrow{R} is anti-transitive. Then, it is possible to arrange the pairs (class word, temporary fulfilment) in an arbitrary order $<_R$ such that $(\tilde{w}, t_{\tilde{w}}) <_R (\tilde{w}, t'_{\tilde{w}})$ if (and only if) $t_{\tilde{w}} \xrightarrow{R} t'_{\tilde{w}}$. Thus, once we leave the pair $(\tilde{w}, t_{\tilde{w}})$, updates involving $t_{\tilde{w}}$ cannot be considered anymore, unless we introduce another set of requests R' . As a matter of fact, the order $<_R$ depends on the chosen set R . Hence, we assume that $|\mathcal{R}|$ total orders $<_R$ have been initially fixed (one for each $R \in \mathcal{R}_\varphi$), all satisfying the above conditions.

Problem (P2) can be dealt with by exploiting the distinctive properties of LISs.

The next lemma gives necessary and sufficient conditions for the expansion of a (not necessarily fulfilling) LIS with a new point.

Lemma 6 *Let φ be an MPNL \sim formula, $\mathbf{L} = (\mathbb{D}, \mathcal{L})$ be a LIS, and R be a set of requests. Then, there is a LIS $\mathbf{L}' = (\mathbb{D}', \mathcal{L}')$, with $D' = D \cup \{x\}$, for some $x \notin D$, $\mathbb{D}' = \mathbb{D} \cup \{x' < x : x' \in D\}$, $\mathcal{L}'([x', x'']) = \mathcal{L}([x', x''])$, for all $x' \leq x'' \in D$, and $\mathbf{req}(x) = R$, if and only if there is $\hat{D} \subseteq D$, with either $\hat{D} = \emptyset$ or $\hat{D} = [x']_{\sim}$, for some $x' \in D$, such that, for each $x'' \in D$, there is an interval tuple $(\mathbf{req}(x''), F, R)$, for some atom F , with $\sim \in F$ if and only if $x'' \in \hat{D}$ and $len_1 \in F$ if and only if $x'' = \max(D)$.*

Proof The proof is straightforward (the claim follows from the properties of LISs).

As for the left-to-right direction, since \mathbf{L}' is a LIS, $(\mathbf{req}(x'), \mathcal{L}'([x', x]), \mathbf{req}(x))$ is an interval tuple, for each $x' \in D'$. We must distinguish two cases. If $|[x]_{\sim}| = 1$ (x is the only element in $[x]_{\sim}$), then $\sim \notin \mathcal{L}'([x'', x])$, for each $x'' \in D$, and $len_1 \in \mathcal{L}'([x'', x])$ if and only if $x'' = \max(D)$. If $|[x]_{\sim}| > 1$, then $\sim \in \mathcal{L}'([x', x])$, for all $x' \in [x]_{\sim}$, $\sim \notin \mathcal{L}'([x', x])$, for all $x' \in D \setminus [x]_{\sim}$, and $len_1 \in \mathcal{L}'([x', x])$ if and only if $x' = \max(D)$.

As for the right-to-left direction, suppose that there exist a set of requests R and a set of points $\hat{D} \subseteq D$, with either $\hat{D} = \emptyset$ or $\hat{D} = [x']_{\sim}$, for some $x' \in D$,

such that, for each $x'' \in D$, there is an interval tuple $(\mathbf{req}(x''), F, R)$, for some atom F , with $\sim \in F$ if and only if $x'' \in \hat{D}$ and $len_1 \in F$ if and only if $x'' = \max(D)$. We let $D' = D \cup \{x\}$, for some $x \notin D$, and $\mathbb{D}' = \mathbb{D} \cup \{x' < x : x' \in D\}$, and we define the labeling function \mathcal{L}' , over the $\mathbb{I}(\mathbb{D}')$, as follows. First of all, we let $\mathcal{L}'([x', x'']) = \mathcal{L}([x', x''])$ for all $x', x'' \in D$, with $x' \leq x''$. Then, we distinguish two cases. Case (i): x belongs to a class $\hat{D} = [x']_{\sim}$, for some $x' \in D$. By hypothesis, for each $x'' \in D$, either $x'' \notin [x']_{\sim}$ and there exists an interval tuple $(\mathbf{req}(x''), F, R)$ such that $\sim \in F$ and $len_1 \in F$ if and only if $x'' = \max(D)$ or $x'' \in [x']_{\sim}$ and there exists an interval tuple $(\mathbf{req}(x''), F, R)$ such that $\sim \in F$ and $len_1 \in F$ if and only if $x'' = \max(D)$. In both cases, we let $\mathcal{L}([x'', x]) = F$. Case (ii): $\hat{D} = \emptyset$. Then, by hypothesis, for each $x'' \in D$, $x'' \notin [x']_{\sim}$ and there exists an interval tuple $(\mathbf{req}(x''), F, R)$ such that $\sim \notin F$ and $len_1 \in F$ if and only if $x'' = \max(D)$. We let $\mathcal{L}([x'', x]) = F$. It is immediate to check that the resulting pair $\mathbf{L}' = (D', \mathcal{L}')$ is a LIS. \square

According to Lemma 6, a point x , with $\mathbf{req}(x) = R$, can be added at a certain computation step if and only if one of the following conditions holds:

- for all x' introduced so far, there is an interval tuple $(\mathbf{req}(x'), F, R)$, with $\sim \notin F$;
- there exist $m (\geq 1)$ points x_1, \dots, x_m , among those introduced so far, such that (i) $x_i \sim x_j$, for all $1 \leq i, j \leq m$, and $x_i \not\sim x'$, for all $1 \leq i \leq m$ and $x' \in D \setminus \{x_1, \dots, x_m\}$; (ii) for all $1 \leq i \leq m$, there exists an interval tuple $(\mathbf{req}(x_i), F, R)$, with $\sim \in F$, and if $x_i = \max(D)$, then $len_1 \in F$; (iii) for at least one $1 \leq i \leq m$, there exists no interval tuple $(\mathbf{req}(x_i), F, R)$, with $\sim \notin F$.

Notice that, while in the second case we are forced to insert x into the class of x_1, \dots, x_m , in the first case we can either introduce a new class and add x to it or insert x in one existing class, provided that for each x' in such a class there is an interval tuple $(\mathbf{req}(x'), F, R)$, with $\sim \in F$, and if $x' = \max(D)$, then $len_1 \in F$.

Formally, this can be done by means of a simple function f_{\sim} that keeps track of the number of classes featuring at least one point x with $\mathbf{req}(x) = R$, for each $R \in \mathcal{R}$. As a matter of fact, it is not necessary to know the exact number of classes, but it suffices to distinguish among the following cases: (i) R does not occur in any class yet, (ii) R occurs in exactly one class, or (iii) R occurs in more than one class. Accordingly, we introduce a function $f_{\sim} : \mathcal{R} \rightarrow \{0, 1, +\}$, called class witness function. We specify how to update such a function when moving from one state to another by means of the control of the VASS.

In the control of the VASS, we put together the set R , the pair (class word, temporary fulfillment) of R , the function f_{\sim} , and a multi-set of at most $|\varphi|$ pairs of the form $(\tilde{w}, t_{\tilde{w}})$, denoted by P_R , called *preemption*. Intuitively, such a multi-set represents the classes that have been used to satisfy those $\langle \bar{A} \rangle$ -requests of R which have been satisfied at a distance greater than 1 (at most $|\varphi| - 1$) plus the pair (class word, temporary fulfillment) of the predecessor of R if (and only if) R and its predecessor have been assigned to different classes. In addition, we include an index m , that allows us to order (according to $<_R$) the updates of the counters.

Finally, given a class word \tilde{w} , a temporary fulfillment $t_{\tilde{w}}$ for it, and a set of requests R , we say that R is *visible* in $(\tilde{w}, t_{\tilde{w}})$ if and only if $\tilde{w}[[t_{\tilde{w}}]] = R$ and $t_{\tilde{w}}[[t_{\tilde{w}}]] = \emptyset$, and we say that R is *invisible* in $(\tilde{w}, t_{\tilde{w}})$ if and only if $t_{\tilde{w}}[[t_{\tilde{w}}]] \neq \emptyset$, $|\{i : i \leq |t_{\tilde{w}}| \wedge \tilde{w}[i] = R\}| = |\varphi|^2 - |\varphi|$, and $|\{i : \tilde{w}[i] = R\}| = |\varphi|^2 - |\varphi| + 1$.

We are now ready to define states and transitions of the VASS.

A state of the VASS is a tuple $(R, \tilde{w}, t_{\tilde{w}}, f_{\sim}, P_R, m)$, where R can be either visible or invisible in $(\tilde{w}, t_{\tilde{w}})$ and $1 \leq m \leq M$, M being the number of possible different pairs $(\tilde{w}, t_{\tilde{w}})$. The set of initial states Q_i contains those states $(R, \tilde{w}, t_{\tilde{w}}, f_{\sim}, \emptyset, 1)$ such that $\langle A \rangle \psi \notin R$, for all $\langle A \rangle \psi \in Tr_{\varphi}$, $\tilde{w}[1] = R$, $|t_{\tilde{w}}| = 1$, $t_{\tilde{w}}[1] = \emptyset$, $f_{\sim}(R) = 1$, and $f_{\sim}(R') = 0$, for all $R' \neq R$. The set of final states Q_f consists of those states $(R, \tilde{w}, t_{\tilde{w}}, f_{\sim}, P_R, M)$ such that $\tilde{w}[t_{\tilde{w}}] = R$, $\langle A \rangle \psi \notin R$, for all $\langle A \rangle \psi \in Tr_{\varphi}$, and the pair $(\tilde{w}, t_{\tilde{w}})$ as well as all pairs $(\tilde{w}', t_{\tilde{w}'}) \in P_R$ are final.

The transition relation is the set union $\Delta_{M,1} \cup \bigcup_{1 \leq k \leq M} \Delta_{k,k} \cup \bigcup_{1 \leq k < M} \Delta_{k,k+1} \cup \Delta_{M,M}^f$, denoted by Δ . The sets $\Delta_{M,1}$, $\Delta_{k,k}$, $\Delta_{k,k+1}$, and $\Delta_{M,M}^f$ consists of triples of the form (q, U, q') , which are defined as follows.

Transitions in $\Delta_{M,1}$ model the addition of a new point at position $x+1$, labelled with a set of requests R' , as a successor of the point at position x , labelled with R . These transitions can change the current state of the automaton $(R, \tilde{w}, t_{\tilde{w}}, f_{\sim}, P_R, M)$ into a new state $(R', \tilde{w}', t_{\tilde{w}'}, f'_{\sim}, P'_R, 1)$ (all the components of the state may change) on the basis of the application of some update U .

Formally, a triple $((R, \tilde{w}, t_{\tilde{w}}, f_{\sim}, P_R, M), U, (R', \tilde{w}', t_{\tilde{w}'}, f'_{\sim}, P'_R, 1))$ belongs to $\Delta_{M,1}$ if and only if it satisfies the following four conditions:

1. First, we check whether a point labelled with R' can be inserted as the immediate successor of a point labelled with R as well as a (not immediate) successor of all predecessors of x belonging to $[x]_{\sim}$ (if any), that is, we check whether there exists a way of labelling intervals $[x', x+1]$, with $x' \leq x$ and $x' \in [x]_{\sim}$.

Formally, let $n = |t_{\tilde{w}}|$, if R is visible; $n = |t_{\tilde{w}}| + 1$ otherwise. There must exist n interval tuples $(\tilde{w}[1], F_1^{pre}, R'), \dots, (\tilde{w}[n-1], F_{n-1}^{pre}, R'), (R, F_n^{pre}, R')$ such that (i) for each $1 \leq i \leq n-1$, $len_1 \notin F_i^{pre}$, (ii) for each $1 \leq i \leq n$, $\sim \in F_i^{pre}$ if and only if $\langle A \rangle (len_1 \wedge \sim) \in R$, and (iii) $len_1 \in F_n^{pre}$.

We distinguish two cases. Case (i): $x+1 \in [x]_{\sim}$ (i.e., $\langle A \rangle (len_1 \wedge \sim) \in R$). We let $\tilde{w} = \tilde{w}'$ and, for each $1 \leq i \leq |t_{\tilde{w}}|$, $t'_{\tilde{w}}[i] = t_{\tilde{w}}[i] \cup \{\langle A \rangle \psi \in Tr_{\varphi} : \psi \in F_i^{pre}\}$. If R' is visible in $(\tilde{w}', t_{\tilde{w}'})$, then $|t_{\tilde{w}'}| = |t_{\tilde{w}}| + 1$; $|t_{\tilde{w}'}| = |t_{\tilde{w}}|$ otherwise. Case (ii): $x+1 \notin [x]_{\sim}$ (i.e., $\langle A \rangle (len_1 \wedge \sim) \notin R$). Let $n' = |t_{\tilde{w}'}| - 1$, if R' is visible in $(\tilde{w}', t_{\tilde{w}'})$; $n' = |t_{\tilde{w}'}|$ otherwise. There must exist a temporary fulfillment $t'_{\tilde{w}'}$ for \tilde{w}' , with $|t'_{\tilde{w}'}| \leq |t_{\tilde{w}'}|$, and n' interval tuples $(\tilde{w}'[1], F'_1, R'), \dots, (\tilde{w}'[n'], F'_{n'}, R')$ such that, for each $1 \leq i \leq n'$, $\sim \in F'_i$ and $t'_{\tilde{w}'}[i] = t'_{\tilde{w}'}[i] \cup \{\langle A \rangle \psi \in Tr_{\varphi} : \psi \in F'_i\}$.

2. Then, we check whether the past requests of the new point can be satisfied. In fact, the satisfaction of some of them comes for free from the fulfillment of condition 1. We must guarantee the satisfaction of the remaining ones.

Let $R^{\bar{A}} = \bigcup_{1 \leq i \leq n} \{\langle \bar{A} \rangle \psi : \psi \in F_i^{pre}\}$, if $x+1 \in [x]_{\sim}$; $R^{\bar{A}} = \bigcup_{1 \leq i \leq n} \{\langle \bar{A} \rangle \psi : \psi \in F_i^{pre}\} \cup \bigcup_{1 \leq i \leq n'} \{\langle \bar{A} \rangle \psi : \psi \in F'_i\}$ otherwise ($x+1 \notin [x]_{\sim}$). There must exist $m \leq |\varphi| - 1$ pairs $(\tilde{w}_1^p, t_{\tilde{w}_1^p}), \dots, (\tilde{w}_m^p, t_{\tilde{w}_m^p})$ (at most one for each past request not belonging to $R^{\bar{A}}$) such that (i) for each $1 \leq i \leq m$, there exist $|t_{\tilde{w}_i^p}|$ interval tuples $(\tilde{w}_i^p[j], F_{i,j}^p, R')$ such that $len_1, \sim \notin F_{i,j}^p$, for each $1 \leq j \leq |t_{\tilde{w}_i^p}|$ (consistency conditions), and (ii) for each $\langle \bar{A} \rangle \psi \in R^{\bar{A}}$, there exist i, j such that $\psi \in F_{i,j}^p$ (fulfilling conditions).

For each $1 \leq i \leq m$, let $t'_{\tilde{w}_i^p}$ be a temporary fulfillment such that $t_{\tilde{w}_i^p} \leq t'_{\tilde{w}_i^p}$ and, for each $1 \leq j \leq |t_{\tilde{w}_i^p}|$, $t'_{\tilde{w}_i^p}[j] = t_{\tilde{w}_i^p}[j] \cup \{\langle A \rangle \psi \in Tr_{\varphi} : \psi \in F_{i,j}^p\}$. We put $P_R = \{(\tilde{w}_i^p, t'_{\tilde{w}_i^p}) : 1 \leq i \leq m\}$ if $x+1 \in [x]_{\sim}$; $P_R = \{(\tilde{w}_i^p, t'_{\tilde{w}_i^p}) : 1 \leq i \leq$

- $m\} \cup \{(\tilde{w}, t_{\tilde{w}_{pre}})\}$, where $|t_{\tilde{w}_{pre}}| = |t_{\tilde{w}}|$ and, for each $1 \leq i \leq |t_{\tilde{w}}|$, $t_{\tilde{w}_{pre}}[i] = t_{\tilde{w}}[i] \cup \{\langle A \rangle \psi \in Tr_{\varphi} : \psi \in F_i^{pre}\}$ otherwise ($x+1 \notin [x]_{\sim}$).
3. Next, we guarantee that the fuction f_{\sim} can be consistently updated. For all $R'' \in \mathcal{R}_{\varphi}$, if $R'' \neq R'$, then $f'_{\sim}(R'') = f_{\sim}(R'')$. Otherwise, if $R'' = R'$, $f'_{\sim}(R') = 1$ if either $f_{\sim}(R') = 0$ or $f_{\sim}(R') = 1$ and there exists $1 \leq i < |t_{\tilde{w}'}|$ such that $\tilde{w}'[i] = R'$; $f'_{\sim}(R') = +$ otherwise. Moreover, for all $R'' \in \mathcal{R}_{\varphi}$, if $f_{\sim}(R'') = +$ or $f_{\sim}(R'') = 1$, $R'' \neq R$, and, for each $1 \leq i < |t_{\tilde{w}'}|$, $\tilde{w}'[i] \neq R''$, then there exists an interval tuple (R'', F, R') such that both $len_1 \notin F$ and $\sim \notin F$.
 4. Finally, we suitably constrain the decrease and the increase of the values of the involved counters. This is done by means of the preemption multi-set. Once more, we distinguish two cases. If $x+1 \in [x]_{\sim}$, then $U = \{c_{\tilde{w}'', t_{\tilde{w}''}} + P_R(\tilde{w}'', t_{\tilde{w}''}) : c_{\tilde{w}'', t_{\tilde{w}''}} \in P_R\} \cup \{c_{\tilde{w}'', t_{\tilde{w}''}} - P'_R(\tilde{w}'', t_{\tilde{w}''}) : c_{\tilde{w}'', t_{\tilde{w}''}} \in P'_R\}$; otherwise, $U = \{c_{\tilde{w}'', t_{\tilde{w}''}} + P_R(\tilde{w}'', t_{\tilde{w}''}) : c_{\tilde{w}'', t_{\tilde{w}''}} \in P_R\} \cup \{c_{\tilde{w}, t_{\tilde{w}}} + 1\} \cup \{c_{\tilde{w}'', t_{\tilde{w}''}} - P'_R(\tilde{w}'', t_{\tilde{w}''}) : c_{\tilde{w}'', t_{\tilde{w}''}} \in P'_R\} \cup \{c_{\tilde{w}', t_{\tilde{w}'}} - 1\}$ (since P_R is a multi-set, we use the standard notation $N(a)$ for such objects, where N is a multiset and a an object, which returns the number of elements a contained in N).

The transitions in $\Delta_{k,k}$ model counter updates which are not directly dealt with by the transitions in $\Delta_{M,1}$. A triple $((R, \tilde{w}, t_{\tilde{w}}, f_{\sim}, P_R, k), U, (R, \tilde{w}, t_{\tilde{w}}, f_{\sim}, P_R, k))$ belongs to $\Delta_{k,k}$ if and only if $U = \{c_{\tilde{w}', t_{\tilde{w}'}} - 1, c_{\tilde{w}', t_{\tilde{w}'}} + 1\}$, $(\tilde{w}', t_{\tilde{w}'})$ is the k -th element in the order $<_R$, and $t_{\tilde{w}'} \xrightarrow{R} t'_{\tilde{w}'}$.

The transitions in $\Delta_{k,k+1}$ are simply used to make one step forward with respect to the ordering $<_R$. Formally, $\Delta_{k,k+1}$ consists of all and only the triples of the form $((R, \tilde{w}, t_{\tilde{w}}, f_{\sim}, P_R, k), \emptyset, (R, \tilde{w}, t_{\tilde{w}}, f_{\sim}, P_R, k+1))$.

Finally, the transitions in $\Delta_{M,M}^f$, together with the acceptance conditions of the automaton, are used to guarantee that, at the end of the computation, all classes are final, and thus the formula is satisfiable. A triple $((R, \tilde{w}, t_{\tilde{w}}, f_{\sim}, P_R, M), U, (R, \tilde{w}, t_{\tilde{w}}, f_{\sim}, P_R, M))$ belongs to $\Delta_{M,M}^f$ if and only if $[A] \neg len_1 \in R$, $(\tilde{w}, t_{\tilde{w}})$ as well as all $(\tilde{w}', t_{\tilde{w}'}) \in P_R$ are final, and $U = \{c_{\tilde{w}'', t_{\tilde{w}''}} - 1\}$, where $(\tilde{w}'', t_{\tilde{w}''})$ is final.

The next two propositions respectively prove the soundness and the completeness of the proposed reduction. Their proofs are given in the appendix.

Proposition 1 *Let φ be an MPNL \sim -formula. If \mathcal{V}_{φ} has a 0-0 computation, then φ is satisfiable over the class of finite linear orders.*

Proposition 2 *Let φ be an MPNL \sim -formula. If φ is satisfiable over the class of finite linear orders, then \mathcal{V}_{φ} has a 0-0 computation.*

Decidability of finite satisfiability for MPNL \sim is formally stated by the following theorem, which is an immediate consequence of the two previous propositions.

Theorem 3 *Let φ be an MPNL \sim -formula. The problem of deciding whether φ is satisfiable over the class of finite linear orders is decidable.*

5 Conclusions

The paper consists of two main parts. In the first one, we proved that the satisfiability problem for PNL, over finite linear orders, extended with one equivalence

relation \sim is still NEXPTIME-complete. First, we showed that the size of every equivalence class of a minimal model of a PNL formula φ is exponentially bounded in its length $|\varphi|$; then, we provided an exponential bound to the number of equivalence classes in a minimal model. The existence of an exponential upper bound to the cardinality of a small model for $\text{PNL}\sim$ easily follows. Since the polynomial reductions from PNL to $\text{FO}^2[<]$ and from $\text{FO}^2[<]$ to PNL can be easily extended to analogous reductions between $\text{PNL}\sim$ and $\text{FO}^2[<, \sim]$, decidability and NEXPTIME-completeness of $\text{FO}^2[<, \sim]$ immediately follow (notice that $\text{FO}^2[<, \sim]$ has the same complexity as weaker fragments like FO^2 , $\text{FO}^2[\sim]$, and $\text{FO}^2[<]$). In the second part of the paper, we showed that decidability is preserved if a metric component is added to $\text{PNL}\sim$. However, proof techniques turn out to be much more complex and computational complexity bounds increase significantly.

We are currently working at the solution of the last missing tile of the puzzle, namely, the satisfiability problem for PNL extended with two equivalence relations \sim_1 and \sim_2 , or, equivalently, $\text{FO}^2[<, \sim_1, \sim_2]$, over finite linear orders (the undecidability of the extension of MPNL with two equivalence relations, or, equivalently, that of $\text{FO}^2[<, +1, \sim_1, \sim_2]$, immediately follows from the undecidability of finite satisfiability for $\text{FO}^2(<, +1, \sim_1, \sim_2)$ proved by Bojańczyk et al. in [2]).

References

1. M. Bojańczyk. Weak MSO with the unbounding quantifier. *Theory of Computing Systems*, 48(3):554–576, 2011.
2. M. Bojańczyk, C. David, A. Muscholl, T. Schwentick, and L. Segoufin. Two-variable logic on data words. *ACM Transactions on Computational Logic*, 12(4):27, 2011.
3. D. Bresolin, D. Della Monica, V. Goranko, A. Montanari, and G. Sciavicco. Metric propositional neighborhood logics on natural numbers. *Software and System Modeling*, 12(2):245–264, 2013.
4. D. Bresolin, V. Goranko, A. Montanari, and G. Sciavicco. Propositional interval neighborhood logics: Expressiveness, decidability, and undecidable extensions. *Annals of Pure and Applied Logic*, 161(3):289–304, 2009.
5. D. Bresolin, A. Montanari, and P. Sala. An optimal tableau-based decision algorithm for propositional neighborhood logic. In *Proc. of the 24th STACS*, volume 4393 of LNCS, pages 549–560. Springer, 2007.
6. D. Bresolin, A. Montanari, and G. Sciavicco. An optimal decision procedure for Right Propositional Neighborhood Logic. *Journal of Automated Reasoning*, 38(1-3):173–199, 2007.
7. Davide Bresolin, Angelo Montanari, Pietro Sala, and Guido Sciavicco. Optimal decision procedures for MPNL over finite structures, the natural numbers, and the integers. *Theoretical Computer Science*, 493:98–115, 2013.
8. S. Demri and R. Lazic. LTL with the freeze quantifier and register automata. *ACM Transactions on Computational Logic*, 10(3), 2009.
9. V. Goranko, A. Montanari, and G. Sciavicco. Propositional interval neighborhood temporal logics. *Journal of Universal Computer Science*, 9(9):1137–1167, 2003.
10. E. Grädel, P. G. Kolaitis, and M. Y. Vardi. On the decision problem for two-variable first-order logic. *Bulletin of Symbolic Logic*, 3(1):53–69, 1997.
11. J. E. Hopcroft and J.-J. Pansiot. On the reachability problem for 5-dimensional vector addition systems. *Theoretical Computer Science*, 8(2):135–159, 1979.
12. R. M. Karp and R. E. Miller. Parallel program schemata. *Journal of Computer and System Sciences*, 3(2):147–195, 1969.
13. E. Kieronski, J. Michaliszyn, I. Pratt-Hartmann, and L. Tendera. Two-variable first-order logic with equivalence closure. In *Proc. of the 27th LICS*, pages 431–440. IEEE, 2012.
14. E. Kieronski and M. Otto. Small substructures and decidability issues for first-order logic with two variables. In *Proc. of the 20th LICS*, pages 448–457. IEEE, 2005.

15. E. Kieronski and L. Tendera. On finite satisfiability of two-variable first-order logic with equivalence relations. In *Proc. of the 24th LICS*, pages 123–132. IEEE, 2009.
16. S. R. Kosaraju. Decidability of reachability in vector addition systems (preliminary version). In *Proc. of the 14th STOC*, pages 267–281. ACM, 1982.
17. R. J. Lipton. The reachability problem requires exponential space. Technical Report 62, Department of Computer Science, Yale University, 1976.
18. A. Montanari, G. Puppis, P. Sala, and G. Sciavicco. Decidability of the interval temporal logic $AB\bar{B}$ on natural numbers. In *Proc. of the 27th STACS*, pages 597–608, 2010.
19. A. Montanari and P. Sala. Adding an equivalence relation to the interval logic $AB\bar{B}$: Complexity and expressiveness. In *Proc. of the 28th LICS*, pages 193–202. IEEE, 2013.
20. M. Mortimer. On languages with two variables. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 21:135–140, 1975.
21. M. Otto. Two variable first-order logic over ordered domains. *Journal of Symbolic Logic*, 66(2):685–702, 2001.
22. J. Ouaknine and J. Worrell. On the decidability and complexity of metric temporal logic over finite words. *Logical Methods in Computer Science*, 3(1), 2007.
23. D. Scott. A decision method for validity of sentences in two variables. *Journal of Symbolic Logic*, 27(4):377, 1962.
24. Y. Venema. A modal logic for chopping intervals. *Journal of Logic and Computation*, 1(4):453–476, 1991.

Appendix

The polynomial reduction from $\text{FO}^2[<]$ to PNL

In the following, we briefly show how to turn the exponential reduction from $\text{FO}^2[<]$ to PNL given in [4] into a polynomial one. We recall that a *polynomial reduction from a logic L to a logic L'* is a polynomial-time procedure that receives a formula α of L as input and returns a formula β of L' as output such that α is (finitely) satisfiable if and only if β is (finitely) satisfiable. In our case, the resulting formula $\beta \equiv \sigma_s(\alpha)$ is obtained by the direct translation of the formula α by means of the transformation χ described in the table below. The final formula used in the reduction will be presented in Theorem 4.

Basic formulas	Non-basic formulas
$\chi_{x,y}(x = x) := \chi_{x,y}(y = y) = \top$	$\chi_{x,y}(\neg\alpha) := \neg\chi_{x,y}(\alpha)$
$\chi_{x,y}(x = y) := \chi_{x,y}(y = x) = \pi$	$\chi_{x,y}(\phi \vee \psi) := \chi_{x,y}(\phi) \vee \chi_{x,y}(\psi)$
$\chi_{x,y}(y < x) := \perp$	$\chi_{x,y}(\exists x\psi) := \langle A \rangle \chi_{y,x}(\psi) \vee [A] \langle \bar{A} \rangle \chi_{x,y}(\psi)$
$\chi_{x,y}(P(x, x)) := \langle \bar{A} \rangle (\pi \wedge p^{\leq} \wedge p^{\geq})$	$\chi_{x,y}(\exists y\psi) := \langle \bar{A} \rangle \chi_{y,x}(\psi) \vee [\bar{A}] \langle A \rangle \chi_{x,y}(\psi)$
$\chi_{x,y}(P(y, y)) := \langle A \rangle (\pi \wedge p^{\leq} \wedge p^{\geq})$	
$\chi_{x,y}(P(x, y)) := p^{\leq}$	
$\chi_{x,y}(P(y, x)) := p^{\geq}$	

The translation of $\text{FO}^2[<]$ formulas into PNL ones.

We now introduce a model transformation ζ that univocally translates each model of $\text{FO}^2[<]$ into a model of PNL.

Definition 2 Let $\mathcal{A} = \langle \mathbb{D}, V_{\mathcal{A}} \rangle$ be a relational first-order model on a vocabulary $\tau \cup \{\leq\}$ (without loss of generality, we assume that τ contains only symbols with arity 2), where $\leq^{\mathcal{A}}$ is a linear order on \mathbb{D} . The corresponding interval model $\zeta(\mathcal{A})$ on $\mathcal{P}rop = \{p^{\leq}, p^{\geq} : P \in \tau\}$ is a pair $\langle \mathbb{I}(\mathbb{D}), V_{\zeta(\mathcal{A})} \rangle$ such that, for any binary relation $P \in \tau$ and any interval $[a, b]$, $[a, b] \in V_{\zeta(\mathcal{A})}(p^{\leq})$ if and only if $(a, b) \in V_{\mathcal{A}}(P)$ and $[a, b] \in V_{\zeta(\mathcal{A})}(p^{\geq})$ if and only if $(b, a) \in V_{\mathcal{A}}(P)$.

The next lemma proves the correctness of the transformation χ using the model transformation ζ .

Lemma 7 *For every $\text{FO}^2[\prec]$ -formula $\alpha(x, y)$, every $\text{FO}^2[\prec]$ -model $\mathcal{A} = \langle \mathbb{D}, V_{\mathcal{A}} \rangle$, and every pair $a, b \in D$, with $a \leq b$, the following hold: (i) $\mathcal{A} \models \alpha(a, b)$ if and only if $\zeta(\mathcal{A}), [a, b] \models \chi_{x,y}(\alpha)$ and (ii) $\mathcal{A} \models \alpha(b, a)$ if and only if $\zeta(\mathcal{A}), [a, b] \models \chi_{y,x}(\alpha)$.*

Proof The proof is a straightforward induction on the complexity of the formula α . \square

Let the formula *sync* be defined as $\text{sync} \equiv \bigwedge_{p \leq, p \geq} [U](\pi \rightarrow (p \leq \leftrightarrow p \geq))$, where $[U]\psi$ is the *universal modality*, which can be defined in PNL as follows:

$$[U]\varphi ::= [A][A][\bar{A}]\varphi \wedge [A][\bar{A}][\bar{A}]\varphi \wedge [\bar{A}][\bar{A}][A]\varphi \wedge [\bar{A}][A][A]\varphi.$$

The constraint *sync* will be exploited by the formula $\sigma_s(\alpha)$ to force each model of its to agree (be *synchronized*) on the value of every interval point (for every pair of symbols $p \leq, p \geq$ in *Prop*). This is needed to prove the right-to-left implication of the next theorem since, otherwise, there could be models of $\sigma_s(\alpha)$ that would not be mappable to an $\text{FO}^2[\prec]$ -model according to inverse transformation of ζ .

Theorem 4 *For every $\text{FO}^2[\prec]$ -formula α , α is (finitely) satisfiable if and only if the PNL-formula*

$$\sigma_s(\alpha) \equiv \text{sync} \wedge (\chi_{x,y}(\varphi) \vee \chi_{y,x}(\varphi))$$

is (finitely) satisfiable.

Proof The claim is a direct consequence of Lemma 7. \square

Note that the above reduction is exponential because of the computational explosion caused by nested quantifiers. To turn it into a polynomial reduction, we can apply first the Scott's normal form to the formula α (in polynomial time) to obtain a formula α' , which is equi-satisfiable with α and whose level of nested quantifiers does not exceed two, and then apply σ_s to α' .

Proposition 1. *Let φ be an MPNL \sim -formula. If \mathcal{V}_φ has a 0-0 computation, then φ is satisfiable over the class of finite linear orders.*

Proof Let $C = (q_0, \bar{0}) \rightarrow \dots \rightarrow (q_n, \bar{0})$ be a 0-0 computation of the VASS \mathcal{V}_φ , where $q_0 \in Q_i$ and $q_n \in Q_f$. According to the definition of Δ , the computation can be splitted in two parts. In the first part (prefix), it alternates between transitions in $\Delta^{M,1}$ and (sequences of) transitions in $\Delta^{k,k}$ and $\Delta^{k,k+1}$ until it reaches a final state $q_f = (R, \tilde{w}, t_{\tilde{w}}, f_{\sim}, P_R, M)$ for the first time. Since $[A]\text{-len}_1 \in R$ (by definition of final state), no further transitions in $\Delta_{M,1}$ can be fired. In the second part (suffix), the computation is thus forced to remain in q_f and only two types of transition can be executed. Let $(\tilde{w}', t_{\tilde{w}'})$ be the M -th pair in the order $<_R$. By construction, either a transition $(q_f, \{c_{\tilde{w}', t_{\tilde{w}'}} - 1, c_{\tilde{w}', t_{\tilde{w}'}} + 1\}, q_f)$ in $\Delta_{M,M}$ or a transition $(q_f, \{c_{\tilde{w}_f, t_{\tilde{w}_f}}\} - 1, q_f) \in \Delta_{M,M}^f$, for some final pair $(\tilde{w}_f, t_{\tilde{w}_f})$, can be fired in q_f . Since the computation ends with the values of counters equal to $\bar{0}$, for each fired transition $(q_f, \{c_{\tilde{w}', t_{\tilde{w}'}} - 1, c_{\tilde{w}', t_{\tilde{w}'}} + 1\}, q_f)$, the pair $(\tilde{w}', t_{\tilde{w}'})$ is final. Moreover, for all pairs $(\tilde{w}, t_{\tilde{w}})$ such that there exists no transition of the form

$(q_f, \{c_{\tilde{w}, t_{\tilde{w}}} - 1, c_{\tilde{w}, t_{\tilde{w}}} + 1\}, q_f)$ in $\Delta_{M,M}^f$, either $c_{\tilde{w}, t_{\tilde{w}}} = 0$ the first time we reach q_f or the pair $(\tilde{w}, t_{\tilde{w}})$ is final, as only counters relative to final pairs are decreased and counters reach the value $\bar{0}$ at the end of the computation.

Now, let $C_{pre} = (q_0, \bar{0}) \rightarrow \dots \rightarrow (q_m, v_m)$ be the computation prefix and let (q_k, v_k) be the result of the last application of a transition in $\Delta_{k,k}$ in C_{pre} . We build a LIS $\mathbf{L} = (\mathbb{D}, \mathcal{L})$ from C_{pre} by a suitable iterative process. \mathbf{L} is obtained as the result of a number of iteration steps, being $\mathbf{L}_i = (\mathbb{D}_i, \mathcal{L}_i)$, where \mathcal{L}_i is not necessarily defined for all points in D_i , the output of the i -th step. Let D_i^\sim be a minimal subset of points in D_i such that for each $x \in D_i$ there exists $x' \in D_i^\sim$ with $x \in [x']_\sim$. Minimality of D_i^\sim guarantees that in D_i^\sim there is exactly one witness for each class in D_i .

In order to define an invariant for the iterative construction, we introduce the following functions:

- $ff_{\mathbf{L}_i}^{\bar{A}} : D_i \rightarrow 2^{D_i}$, that maps each $x \in D_i$ into a set of points $ff_{\mathbf{L}_i}^{\bar{A}}(x)$ such that, for each $\psi \in \mathbf{req}_{\bar{A}}(x)$, there exists $x' < x$ in $ff_{\mathbf{L}_i}^{\bar{A}}(x)$ with $\psi \in \mathcal{L}([x', x])$;
- $f_{\mathcal{W}}^i : D_i^\sim \rightarrow \tilde{\mathcal{W}}_\varphi$, that maps each $x \in D_i^\sim$ into its class word $\tilde{w}_{[x]_\sim}$;
- $f_{\mathcal{T}}^i : D_i^\sim \rightarrow \mathcal{T}$, that maps each $x \in D_i^\sim$ into its temporary fulfillment at step i ;
- $f_{C_{pre}}^i : D_i^\sim \rightarrow \mathcal{T}$, that maps each $x \in D_i^\sim$ into its temporary fulfillment in the computation C_{pre} .

We would like to briefly explain the role of the last two functions, namely, $f_{\mathcal{T}}^i$ and $f_{C_{pre}}^i$, that map a given equivalence class (that is, an element of D_i^\sim) into the LIS \mathbf{L}_i and the computation C_{pre} , respectively. Intuitively, the execution of a transition in $\Delta_{M,1}$ corresponds to the insertion of a new point x , with $\mathbf{req}(x) = R$, in \mathbf{L}_i as the successor of the largest point of \mathbb{D}_i . As we already pointed out, the control of the automaton only provides the labeling for the subset of intervals that end at points belonging to $[x]_\sim$ (the class of x), to the class of the predecessor of x (if any), or to the classes $[x']_\sim$ ($\neq [x]_\sim$) which are used to satisfy the \bar{A} requests in R . As for the points belonging to the remaining classes, it may happen that the automaton executes some, but not all, transitions in $\Delta_{k,k}$, that is, some “active” (that is, executable) transitions in $\Delta_{k,k}$ are possibly not executed (recall that the control of the automaton cannot execute any zero-test operation). However, the fact that not all active transitions are actually executed by the automaton does not introduce any inconsistency in \mathbf{L}_i , as the function f_\sim guarantees the existence of a consistent labeling of the intervals beginning at these points and ending at x . What happens is simply that the addition of a point x , which turns out to completely invisible to the computation, can force an expansion of the temporary words of some classes in \mathbf{L}_i . It immediately follows that, at each step i , the temporary words in \mathbf{L}_i are supersets of those in the computation, that is, for all $x \in D_i$, $f_{C_{pre}}^i(x) \leq f_{\mathcal{T}}^i(x)$. However, this is not a problem, because being the computation successful guarantees that all the temporary words in the automaton will be sooner or later saturated and thus all the requests in the resulting LIS \mathbf{L} will be satisfied.

Let $q_i = (R^i, \tilde{w}^i, t_{\tilde{w}^i}, f_\sim^i, P_{R^i}^i, k^i)$. We show that, at each iteration of the construction process, the truth of the following invariant is guaranteed:

1. for all $x \in D_i$, it holds that $f_{\mathcal{W}}^i(x) = f_{\mathcal{W}}^{i+1}(x)$ and $ff_{\mathbf{L}_i}^{\bar{A}}(x) = ff_{\mathbf{L}_{i+1}}^{\bar{A}}(x)$;
2. for all $x, x' \in D_i$, with $x' < x$ and $x \neq \max(D_i)$, $\mathcal{L}_i([x', x])$ is defined and, for each $\langle \bar{A} \rangle \psi \in \mathbf{req}_{\bar{A}}(x)$, there exists $x'' \in D_i$ such that $\psi \in \mathcal{L}([x'', x])$;

3. for all $x \in D_i^\sim$, it holds that $f_{C_{pre}}^i(x) \leq f_{\mathcal{T}}^i(x)$;
4. for all $(\tilde{w}, t_{\tilde{w}})$, it holds that $|\{x \in D_i^\sim : f_{C_{pre}}^i(x) = t_{\tilde{w}} \wedge [x]_\sim \cap \overline{ff_{\mathbf{L}_i}^A}(\max(D_i)) = \emptyset\}| = c_{\tilde{w}, t_{\tilde{w}}}$;
5. for all $x \in D_i^\sim$, either $\mathcal{L}_i([x', \max(D_i^\sim)])$ is defined for all $x' \sim x$ in D_i or it is not defined for any $x' \sim x$ in D_i ;
6. for all $x \in D_i^\sim$ such that $[x]_\sim \cap \overline{ff_{\mathbf{L}_i}^A}(\max(D_i)) = \emptyset$, if $\mathcal{L}_i([x, \max(D_i)])$ is defined and $(\tilde{w}', t_{\tilde{w}'})$ is the k^i -th element in the order $<_{R^i}$, then $(f_{\mathcal{W}}^i(x), f_{C_{pre}}^i(x)) <_{R^i} (\tilde{w}', t_{\tilde{w}'})$;
7. for all $x \in D_i$, with $x \sim x'$, for some $x' \in D_i^\sim$ with $f_{\mathcal{W}}^i(x') = \tilde{w}$ and $f_{\mathcal{T}}^i(x') = t_{\tilde{w}}$, if $pos_\sim(x) > |\varphi^2| - |\varphi|$ and $\tilde{w}[j] \neq \mathbf{req}(x)$ for all $j > |t_{\tilde{w}}|$, then, for all $x'' \in D_i$, with $x'' > \max(\mathbf{req}(x), [x']_\sim)$, $\mathcal{L}_i([x, x'']) = \mathcal{L}_i([\max(\mathbf{req}(x), [x']_\sim), x''])$;
8. for all $R \in \mathcal{R}_\varphi$, it holds that $f_{\sim}^i(R) = \#_R^i$, if $\#_R^i \leq 1$, and $f_{\sim}^i(R) = +$ otherwise, where $\#_R^i = |\{x \in D_i^\sim : \exists x' \sim x(\mathbf{req}(x') = R)\}|$;
9. for all $(\tilde{w}, t_{\tilde{w}})$, it holds that $P_R^i(\tilde{w}, t_{\tilde{w}}) = CP_R^i(\tilde{w}, t_{\tilde{w}})$, if $(\tilde{w}, t_{\tilde{w}}) \neq (\tilde{w}^i, t_{\tilde{w}^i})$, and $P_R^i(\tilde{w}, t_{\tilde{w}}) = CP_R^i(\tilde{w}, t_{\tilde{w}}) + 1$ otherwise, where $CP_R^i(\tilde{w}, t_{\tilde{w}}) = |\{[x']_\sim : x' \in \overline{ff_{\mathbf{L}_i}^A}(\max(D_i)) \wedge f_{\mathcal{W}}^i(x') = \tilde{w} \wedge f_{C_{pre}}^i(x') = t_{\tilde{w}}\}|$.

Let $q_0 = (f_\sim, R, \tilde{w}, t_{\tilde{w}}, \emptyset, 1)$ be the initial state of the computation. We put $D_0 = \{x_0\}$, $\mathbf{req}(x_0) = R$, $f_{\mathcal{W}}^0(x_0) = \tilde{w}$, $f_{C_{pre}}^0(x_0) = f_{\mathcal{T}}^0(x_0) = t_{\tilde{w}}$, and $\overline{ff_{\mathbf{L}_0}^A}(x_0) = \emptyset$. It is easy to check that it satisfies the invariant conditions.

Let us now consider the generic step $i + 1$. Let $q_i = (R^i, \tilde{w}^i, t_{\tilde{w}^i}, f_{\sim}^i, P_R^i, k^i)$ and assume that \mathbf{L}_i satisfies the invariant conditions 1–9. We show how to build \mathbf{L}_{i+1} in such a way that it satisfies them as well. Let $((R^i, \tilde{w}^i, t_{\tilde{w}^i}, f_{\sim}^i, P_R^i, k^i), U_i, (R^{i+1}, \tilde{w}^{i+1}, t_{\tilde{w}^{i+1}}, f_{\sim}^{i+1}, P_R^{i+1}, k_{i+1}))$ be the fired transition. We must distinguish among three cases, depending on its type.

- Case (i): the fired transition belongs to $\Delta_{M,1}$.

We first complete the labeling of the intervals ending in $\max(D_i)$. Let the labeling $\mathcal{L}_i([x, \max(D_i)])$, for some $x \in D_i$, with $x < \max(D_i)$, be not defined. By condition 5 of the invariant, the labeling $\mathcal{L}_i([x', \max(D_i)])$ is not defined for all $x' \sim x$. Let $\{\bar{x}_1, \dots, \bar{x}_m\}$ be the set of all points in $[x]_\sim$ belonging to D_i . By definition of f_{\sim}^i and by condition 8 of the invariant (that links it to the current LIS), for each $1 \leq j \leq m$, there exists an interval tuple $(\mathbf{req}(\bar{x}_j), F_j, \mathbf{req}(\max(D_i)))$, with $\text{len}_1, \sim \notin F_j$. It is not difficult to check that, for all $1 \leq j \leq j' \leq m$, if $\mathbf{req}(\bar{x}_j) = \mathbf{req}(\bar{x}_{j'})$, then we can safely assume $F_j = F_{j'}$. This guarantees that condition 7 of the invariant is preserved. We define $\mathcal{L}_{i+1}([\bar{x}_j, \max(D_i)]) = F_j$, for all $1 \leq j \leq m$.

We proceed in this way until we define the labeling $\mathcal{L}_{i+1}([x', \max(D_i)])$ for all x' for which it was not already defined.

Now, we extend D_i with a new point \hat{x} , that is, we put $D_{i+1} = D_i \cup \{\hat{x}\}$, with $\hat{x} > x$ for all $x \in D_i$. Let $\mathbf{req}(\hat{x}) = R^{i+1}$. The labeling \mathcal{L}_{i+1} for the intervals ending in \hat{x} and beginning at some point belonging to $[max(D_i)]_\sim$ (case 1), $[\hat{x}]_\sim$ (case 2), or the classes of the past far friends of \hat{x} (case 3) is defined as follows.

1. Let $n = |t_{\tilde{w}^i}|$, if R^i is visible in $(\tilde{w}^i, t_{\tilde{w}^i})$; $n = |t_{\tilde{w}^i}| + 1$, otherwise. By definition of $\Delta_{M,1}$, there exist n interval tuples $(\tilde{w}^i[1], F_1^{pre}, R^{i+1}), \dots, (\tilde{w}^i[n-1], F_{n-1}^{pre}, R^{i+1}), (R^i, F_n^{pre}, R^{i+1})$ such that (i) for all $1 \leq j \leq n$, $\sim \in F_j^{pre}$ if and only if $\langle A \rangle(\text{len}_1 \wedge \sim) \in R^i$, (ii) $\text{len}_1 \in F_n^{pre}$, and (iii) $\text{len}_1 \notin F_j^{pre}$, for all

- $1 \leq j < n$. By the invariant conditions 4 and 9, it holds that $f_{\mathcal{W}}^i(\max(D_i)) = \tilde{w}^i$ and $f_{\mathcal{T}}^i(\max(D_i)) = t_{\tilde{w}^i}$. Moreover, for all $x'' \in [\max(D_i)]_{\sim}$, if x'' is meaningful, then there exists $1 \leq j \leq n$ such that $j = |\{x''' \in [\max(D_i)]_{\sim} : x''' < x'' \wedge x''' \text{ is meaningful}\}|$ and we put $\mathcal{L}_{i+1}([x'', \hat{x}]) = F_j^{pre}$. If x'' is not meaningful, two cases are possible. If $\tilde{w}^i[j'] \neq \mathbf{req}(x'')$ for all $j' > |t_{\tilde{w}^i}|$, then the point x''' that, at the end of the construction, will turn out to be the largest point in $[\max(D_i)]_{\sim}$ with set of requests equal to $\mathbf{req}(x'')$ has been already introduced in D_i . Since x'' is not meaningful, it holds that $x''' > x''$. Then, we put $\mathcal{L}_{i+1}([x'', \hat{x}]) = \mathcal{L}_{i+1}([x''', \hat{x}])$, since we have already defined \mathcal{L}_{i+1} for meaningful points. On the contrary, if there exists $j' > |t_{\tilde{w}^i}|$ such that $\tilde{w}^i[j'] = \mathbf{req}(x'')$, we can conclude that the above-defined point x''' has not been introduced yet. Since x'' is not meaningful, there are exactly $|\varphi|^2 - |\varphi|$ meaningful points $x''' < x''$ in $[\max(D_i)]_{\sim}$, with $\mathbf{req}(x''') = \mathbf{req}(x'')$. We take one of these points x''' and we put $\mathcal{L}_{i+1}([x'', \hat{x}]) = \mathcal{L}_{i+1}([x''', \hat{x}])$, since we have already defined \mathcal{L}_{i+1} for meaningful points. Finally, being $\max(D_i)$ meaningful or not, we put $\mathcal{L}([\max(D_i), \hat{x}]) = F_n^{pre}$.
2. If $|t_{\tilde{w}^{i+1}}| = 1$, then \hat{x} is the first point in its class, and then we skip this step. Similarly, if $\langle A \rangle(\text{len}_1 \wedge \sim) \in R^i$, then the labeling of the intervals that begin at a point in $[\hat{x}]_{\sim}$ and end at \hat{x} has been already defined (case 1), and thus we skip this step as well. Let $n = |t_{\tilde{w}^{i+1}}| - 1$, if R^{i+1} is visible in $(\tilde{w}^{i+1}, t_{\tilde{w}^{i+1}})$, that is, R^{i+1} has been added to the temporary word (at step $i+1$); $n = |t_{\tilde{w}^{i+1}}|$ otherwise, that is, R^{i+1} has not be added to the temporary word as it is not visible. By case (ii) of condition 1 of the definition of $\Delta_{M,1}$, there exist a temporary fulfillment $t'_{\tilde{w}^{i+1}}$ for \tilde{w}^{i+1} and n interval tuples $(\tilde{w}^{i+1}[1], F_1^{i+1}, R^{i+1}), \dots, (\tilde{w}^{i+1}[n], F_n^{i+1}, R^{i+1})$ such that for each $1 \leq j \leq n$, $-\text{len}_1, \sim \in F_j^{i+1}$ and $t_{\tilde{w}^{i+1}}[j] = t'_{\tilde{w}^{i+1}}[j] \cup \{\langle A \rangle \psi \in TR(\varphi) : \psi \in F_j^{i+1}\}$. Then, by condition 4 or condition 9 of the invariant, there exists x' in D_i with $f_{\mathcal{W}}^i(x') = \tilde{w}^{i+1}$ and $f_{C_{pre}}^i(x') = t'_{\tilde{w}^{i+1}}$. More precisely, if there exists a point $x'' \in \text{ff}_{L_i}^{\bar{A}}(\max(D_i))$, with $f_{\mathcal{W}}^i(x'') = \tilde{w}^{i+1}$ and $f_{C_{pre}}^i(x'') = t'_{\tilde{w}^{i+1}}$, then we take $x' = x''$ (condition 9 of the invariant); otherwise, the existence of such a point x' is guaranteed by condition 4 of the invariant. For each $x'' \in [x']_{\sim}$ if x'' is meaningful, then there exists $1 \leq j \leq n$ such that $j = |\{x''' \in [x']_{\sim} : x''' < x'' \wedge x''' \text{ is meaningful}\}|$, and we put $\mathcal{L}_{i+1}([x'', \hat{x}]) = F_j^{i+1}$. If x'' is not meaningful, there are two possible situations (as in case 1). If $\tilde{w}^{i+1}[j'] \neq \mathbf{req}(x'')$ for all $j' > |t_{\tilde{w}^{i+1}}|$, then the point x''' that, at the end of the construction, will turn out to be the largest point in $[x']_{\sim}$ with set of requests equal to $\mathbf{req}(x'')$ has been already introduced in D_i . Since x'' is not meaningful, it holds that $x''' > x''$. Then, we put $\mathcal{L}_{i+1}([x'', \hat{x}]) = \mathcal{L}_{i+1}([x''', \hat{x}])$, since we have already defined \mathcal{L}_{i+1} for meaningful points. On the contrary, if there exists $j' > |t_{\tilde{w}^{i+1}}|$ such that $\tilde{w}^{i+1}[j'] = \mathbf{req}(x'')$, we can conclude that the above-defined point x''' has not been introduced yet. Since x'' is not meaningful, there are exactly $|\varphi|^2 - |\varphi|$ meaningful points $x''' < x''$ in $[x']_{\sim}$ with $\mathbf{req}(x''') = \mathbf{req}(x'')$. We take one of these points x''' and we put $\mathcal{L}_{i+1}([x'', x]) = \mathcal{L}_{i+1}([x''', x])$, since we have already defined \mathcal{L}_{i+1} for meaningful points.
3. Let $n = |t_{\tilde{w}^{i+1}}| - 1$, if R^{i+1} is visible in $(\tilde{w}^{i+1}, t_{\tilde{w}^{i+1}})$, $n = |t_{\tilde{w}^{i+1}}|$ otherwise, and let $n' = |t_{\tilde{w}^i}|$, if R^i is visible in $(\tilde{w}^i, t_{\tilde{w}^i})$, $n' = |t_{\tilde{w}^i}| + 1$, otherwise.

Moreover, let $R^{\bar{A}} = \bigcup_{1 \leq j \leq n} \{\langle \bar{A} \rangle \psi : \psi \in F_j^{pre}\}$, if $\langle A \rangle (\text{len}_1 \wedge \sim) \in R^i$, and $R^{\bar{A}} = \bigcup_{1 \leq j \leq n} \{\langle \bar{A} \rangle \psi : \psi \in F_j^{pre}\} \cup \bigcup_{1 \leq j \leq n'} \{\langle \bar{A} \rangle \psi : \psi \in F_j^{i+1}\}$ otherwise.

By condition 2 of the definition of $\Delta_{M,1}$, there are $m \leq |\varphi| - 1$ pairs $(\tilde{w}_1^p, t_{\tilde{w}_1^p}), \dots, (\tilde{w}_m^p, t_{\tilde{w}_m^p})$ such that (i) for each $1 \leq i' \leq m$, there are $|t_{\tilde{w}_{i'}^p}|$ interval tuples $(\tilde{w}_{i'}^p[j], F_{i',j}^p, R^{i+1})$ with $\text{len}_1, \sim \notin F_{i',j}^p$ for each $1 \leq j \leq |t_{\tilde{w}_{i'}^p}|$ (consistency conditions), and (ii) for each $\langle \bar{A} \rangle \psi \in R^{i+1} \setminus R^{\bar{A}}$, there are i', j such that $\psi \in F_{i',j}^p$ (fulfilling conditions). By the same condition, it also holds that $P_{R^{i+1}} = \{(\tilde{w}_j^p, t_{\tilde{w}_j^p}) : 1 \leq j \leq m\}$, if $\langle A \rangle (\text{len}_1 \wedge \sim) \in R^i$; $P_{R^{i+1}} = \{(\tilde{w}_j^p, t_{\tilde{w}_j^p}) : 1 \leq j \leq m\} \cup \{(\tilde{w}^i, t_{\tilde{w}^i})\}$, with $|t_{\tilde{w}^i}| = |t_{\tilde{w}^i}|$ and $t_{\tilde{w}^i}[j'] = t_{\tilde{w}^i}[j'] \cup \langle A \rangle \psi \in TR(\varphi) : \psi \in F_{j'}^{pre}\}$, for all $1 \leq j' \leq |t_{\tilde{w}^i}|$, otherwise.

By condition 4 of the definition of $\Delta_{M,1}$ (definition of U), $U_i = \{c_{\tilde{w}''} + P_{R^i}(\tilde{w}'', t_{\tilde{w}''}) : c_{\tilde{w}''} \in P_{R^i}\} \cup \{c_{\tilde{w}''} - P_{R^{i+1}}(\tilde{w}'', t_{\tilde{w}''}) : c_{\tilde{w}''} \in P_{R^{i+1}}\}$, if $\langle A \rangle (\text{len}_1 \wedge \sim) \in R^i$; $U_i = \{c_{\tilde{w}''} + P_{R^i}(\tilde{w}'', t_{\tilde{w}''}) : c_{\tilde{w}''} \in P_{R^i}\} \cup \{c_{\tilde{w}^i} + 1\} \cup \{c_{\tilde{w}''} - P_{R^{i+1}}(\tilde{w}'', t_{\tilde{w}''}) : c_{\tilde{w}''} \in P_{R^{i+1}}\} \cup \{c_{\tilde{w}^{i+1}} - 1\}$ otherwise. Since we have already dealt with the class of \hat{x} and the class of its predecessor, we can restrict our attention to the update $\{c_{\tilde{w}''} + P_{R^i}(\tilde{w}'', t_{\tilde{w}''}) : c_{\tilde{w}''} \in P_{R^i}\} \cup \{c_{\tilde{w}''} - P_{R^{i+1}}(\tilde{w}'', t_{\tilde{w}''}) : c_{\tilde{w}''} \in P_{R^{i+1}}\}$. By a simple counting argument, which makes use of conditions 4 and 9 of the invariant (that hold for \mathbf{L}_i by the inductive hypothesis), we can conclude that, for each pair $(\tilde{w}, t_{\tilde{w}})$, $|\{j : (\tilde{w}_j^p, t_{\tilde{w}_j^p}) = (\tilde{w}, t_{\tilde{w}})\}| \leq |\{x' \in D_i^{\sim} : f_{C_{pre}}^i(x') = t_{\tilde{w}} \wedge f_{\mathcal{W}}^i(x') = \tilde{w}\}| = c_{\tilde{w}, t_{\tilde{w}}}^i + |P_{R^i}(\tilde{w}, t_{\tilde{w}})|$.

Now, we can choose $\bar{x}_1, \dots, \bar{x}_m$ distinct points in D_i^{\sim} such that, for each $1 \leq j \leq m$, $f_{\mathcal{W}}^i(\bar{x}_j) = \tilde{w}_j$, $f_{C_{pre}}^i(\bar{x}_j) = t_{\tilde{w}_j}$, $\bar{x}_j \not\prec \hat{x}$, and $\bar{x}_j \not\prec \max(D_i)$. In analogy with cases 1 and 2, we can define the labeling of the intervals that begin at points belonging to $[\bar{x}_1]_{\sim}, \dots, [\bar{x}_m]_{\sim}$ iteratively. The labeling procedure consists of two nested loops, the external one is indexed by j , with $1 \leq j \leq m$, and it ranges over the classes; the internal one is indexed by j' , with $1 \leq j' \leq |t_{\tilde{w}_j}|$, and it ranges over the meaningful elements of $[\bar{x}_j]_{\sim}$. Let us consider the j -th iteration step of the external loop. At each step j' , we take the j' -th meaningful point $x' \in [\bar{x}_j]_{\sim}$, that is, $j' = |\{x'' \in [\bar{x}_j]_{\sim} : x'' < x' \wedge x'' \text{ is meaningful}\}|$, and we put $\mathcal{L}_{i+1}([x', \hat{x}]) = F_{j,j'}^p$. At the end of the external loop, we only need to define the labeling of the non-meaningful points $x' \in [\bar{x}_j]_{\sim}$, for all $1 \leq j \leq m$. To this end, a single loop on the classes $[\bar{x}_1]_{\sim}, \dots, [\bar{x}_m]_{\sim}$ suffices. Once more, for each non-meaningful point $x' \in [\bar{x}_j]_{\sim}$, two cases are possible. If $\tilde{w}^j[j'] \neq \mathbf{req}(x')$ for all $j' > |t_{\tilde{w}^j}|$, then the point x'' that, at the end of the construction, will turn out to be the largest point in $[x_j]_{\sim}$ with set of requests equal to $\mathbf{req}(x')$ has been already introduced in D_i . Since x' is not meaningful, it holds that $x'' > x'$. Then, we put $\mathcal{L}_{i+1}([x', \hat{x}]) = \mathcal{L}_{i+1}([x'', \hat{x}])$, since we have already defined \mathcal{L}_{i+1} for the pair (x'', \hat{x}) . On the contrary, if there exists $j' > |t_{\tilde{w}^j}|$ such that $\tilde{w}^j[j'] = \mathbf{req}(x')$, we can conclude that the above-defined point x'' has not been introduced yet. Since x' is not meaningful, there are exactly $|\varphi|^2 - |\varphi|$ meaningful points $x'' < x'$ in $[x_j]_{\sim}$, with $\mathbf{req}(x'') = \mathbf{req}(x')$. We take one of these points x'' and we put $\mathcal{L}_{i+1}([x', \hat{x}]) = \mathcal{L}_{i+1}([x'', \hat{x}])$, since we have already defined \mathcal{L}_{i+1} for the pair (x'', \hat{x}) .

It can be easily checked that, at the end of this step, the invariant conditions are satisfied by the LIS \mathbf{L}_{i+1} .

- Case (ii): the fired transition belongs to $\Delta_{k,k}$, for some $1 \leq k \leq M$.
By definition of $\Delta_{k,k}$, $R^i = R^{i+1}$, $\tilde{w}^i = \tilde{w}^{i+1}$, $t_{\tilde{w}^i} = t_{\tilde{w}^{i+1}}$, $f_{\tilde{\sim}}^i = f_{\tilde{\sim}}^{i+1}$, $P_R^i = P_R^{i+1}$, $k_i = k_{i+1}$, and $U_i = \{c_{\tilde{w}, t_{\tilde{w}}} - 1, c_{\tilde{w}, t'_{\tilde{w}}} + 1\}$ for some word \tilde{w} and some pair of its temporary fulfillments $t_{\tilde{w}}$ and $t'_{\tilde{w}}$. Moreover, by definition of $t_{\tilde{w}} \xrightarrow{R} t_{\tilde{w}'}$, $t_{\tilde{w}} \leq t_{\tilde{w}'}$ and, for each $1 \leq j \leq |t_{\tilde{w}}|$, there is an interval tuple $(\tilde{w}[j], F_j, R^i)$ such that $len_1, \sim \notin F_j$ and $t'_{\tilde{w}}[j] = t_{\tilde{w}}[j] \cup \{\langle A \rangle \psi \in \tilde{w}[j] : \psi \in F_j\}$. By conditions 4-6 of the invariant, there is $x \in D_i^{\sim}$ such that (i) for all $x' \sim x$, $x' \notin \text{ff}_{\mathbf{L}_i^{\bar{A}}}(\max(D_i))$ and the labeling $\mathcal{L}_i([x', \max(D_i)])$ is undefined, (ii) $f_{\mathcal{W}}^i(x) = \tilde{w}$, and (iii) $f_{C_{pre}}^i(x) = t_{\tilde{w}}$. Let $\bar{x}_0 < \dots < \bar{x}_m$ be all the points in $[x]_{\sim}$. For each $1 \leq j \leq m$, we proceed as follows:
 - (i) if $pos_{\sim}(\bar{x}_j) \leq |\varphi|^2 - |\varphi|$, then we put $\mathcal{L}_{i+1}([\bar{x}_j, \max(D_i)]) = F_{j'}$ where j' is the minimal index satisfying $R^{j'} = \mathbf{req}(\bar{x}_j)$ and $|\{j'' \leq j' : \tilde{w}[j''] = \mathbf{req}(\bar{x}_j)\}| = pos_{\sim}(\bar{x}_j)$;
 - (ii) if $pos_{\sim}(\bar{x}_j) \geq |\varphi|^2 - |\varphi| + 1$, $\tilde{w}[j''] \neq \mathbf{req}(\bar{x}_j)$, for all $j'' > |t_{\tilde{w}}|$, and $\bar{x}_j = \max(\{\bar{x}_{j'} : \mathbf{req}(\bar{x}_{j'}) = \mathbf{req}(\bar{x}_j)\})$, then we put $\mathcal{L}_{i+1}([\bar{x}_j, \max(D_i)]) = F_{j'}$, where j' is the maximal index for which $R^{j'} = \mathbf{req}(\bar{x}_j)$;
 - (iii) if $pos_{\sim}(\bar{x}_j) \geq |\varphi|^2 - |\varphi| + 1$ and there is $j'' > |t_{\tilde{w}}|$ such that $\tilde{w}[j''] = \mathbf{req}(\bar{x}_j)$, then we take any $\bar{x}_{j'} < \bar{x}_j$, with $pos_{\sim}(\bar{x}_{j'}) \leq |\varphi|^2 - |\varphi|$ and $\mathbf{req}(\bar{x}_{j'}) = \mathbf{req}(\bar{x}_j)$, and we put $\mathcal{L}_{i+1}([\bar{x}_j, \max(D_i)]) = \mathcal{L}_{i+1}([\bar{x}_{j'}, \max(D_i)])$ (notice that the labeling $\mathcal{L}_{i+1}([\bar{x}_{j'}, \max(D_i)])$ has been defined in (i));
 - (iv) if $pos_{\sim}(\bar{x}_j) \geq |\varphi|^2 - |\varphi| + 1$, $\tilde{w}[j''] \neq \mathbf{req}(\bar{x}_j)$, for all $j'' > |t_{\tilde{w}}|$, and $\bar{x}_j \neq \max(\{\bar{x}_{j'} : \mathbf{req}(\bar{x}_{j'}) = \mathbf{req}(\bar{x}_j)\})$, then we take $\bar{x}_{j''} = \max(\{\bar{x}_{j'} : \mathbf{req}(\bar{x}_{j'}) = \mathbf{req}(\bar{x}_j)\})$, and we put $\mathcal{L}_{i+1}([\bar{x}_j, \max(D_i)]) = \mathcal{L}_{i+1}([\bar{x}_{j''}, \max(D_i)])$ (notice that the labeling $\mathcal{L}_{i+1}([\bar{x}_{j''}, \max(D_i)])$ has been defined in (ii)).
 At the end of the procedure, it holds that $f_{C_{pre}}^{i+1}(x') = f_{C_{pre}}^i(x')$ and $f_{\mathcal{T}}^{i+1}(x') = f_{\mathcal{T}}^i(x')$ for all $x' \neq x$ in D_i^{\sim} , and thus condition 3 of the invariant is satisfied by these points. It remains to check that such a condition is satisfied by x as well. By definition of $f_{C_{pre}}^i$, it holds that $f_{C_{pre}}^i(x) = t_{\tilde{w}}$ and $f_{C_{pre}}^{i+1}(x) = t'_{\tilde{w}}$. Moreover, by definition of $\Delta_{k,k}$, for each $1 \leq j \leq |t_{\tilde{w}}|$, $t'_{\tilde{w}}[j] = t_{\tilde{w}}[j] \cup \{\langle A \rangle \psi \in \tilde{w}[j] : \psi \in F_j\}$. By the above construction, it holds that $f_{\mathcal{T}}^{i+1}(x)[j] = f_{\mathcal{T}}^i(x)[j] \cup \{\langle A \rangle \psi \in \tilde{w}[j] : \psi \in F_j\}$ for all $1 \leq j \leq |t_{\tilde{w}}|$. Finally, by condition 3 of the invariant, it holds that $t_{\tilde{w}} \leq f_{\mathcal{T}}^i(x)$ and thus $t'_{\tilde{w}} = f_{C_{pre}}^{i+1}(x) \leq f_{\mathcal{T}}^{i+1}(x)$. This allows us to conclude that all the conditions of the invariant are satisfied.
- Case (iii): the fired transition belongs to $\Delta_{k,k+1}$, for some $1 \leq k < M$.
We simply put $\mathbf{L}_{i+1} = \mathbf{L}_i$, thus leaving everything unchanged.

This basically concludes the proof, because at step $|C_{pre}|$, by the invariant conditions, it holds that $\mathbf{L}_{|C_{pre}|}$ is a consistent LIS for which the \bar{A} -requests of each point are satisfied. Moreover, the A -requests of meaningful points are satisfied thanks to condition 3 of the invariant and to the fact that after step $|C_{pre}|$ only counters $c_{(\tilde{w}, t_{\tilde{w}})}$, where $(\tilde{w}, t_{\tilde{w}})$ is final, are decreased. Finally, the fulfillment of A -requests for the points in $D^{|C_{pre}|}$, which are not meaningful, is guaranteed by condition 7 of the invariant. \square

Proposition 2. *Let φ be an MPNL \sim formula. If φ is satisfiable over the class of finite linear orders, then \mathcal{V}_φ has a 0-0 computation.*

Proof Since φ is satisfiable, by Lemma 5, there exists a saturated well-match $(\mathbf{L}, ff_{\mathbf{L}})$, with $\mathbf{L} = (\mathbb{D}, \mathcal{L})$, for it. We show how to obtain a 0-0 computation $C = (q_0, \bar{0}) \rightarrow \dots \rightarrow (q_n, \bar{0})$, with $q_0 \in Q_i$ and $q_n \in Q_f$, that mimics the behavior of \mathbf{L} . Given a point $x \in D$, let $pre(x)$ (resp., $next(x)$) be the immediate predecessor (resp., successor) of x in the finite linear order \mathbb{D} . Moreover, let $D_x = \{x' \in D : x' \leq x\}$.

For each $x \in D$, we define the function

$$f_{\sim}^x(R) = \begin{cases} + & \exists x' < x'' \leq x \left(\mathbf{req}(x') = \mathbf{req}(x'') = R \wedge x' \not\sim x'' \right) \\ 1 & \exists x' < x \left(\mathbf{req}(x') = R \wedge \forall x'' \leq x \left(\mathbf{req}(x'') = R \rightarrow x'' \sim x' \right) \right) \\ 0 & \text{otherwise.} \end{cases}$$

We build the 0-0 computation C of the VASS \mathcal{V}_φ by the following iterative procedure. Let (q_i, v_i) be the i -th configuration of C , with $q_i = (R^i, \tilde{w}^i, t_{\tilde{w}^i}, f_{\sim}^i, P_{R^i}, k)$, and let $t(i)$ be the number of transitions in $\Delta_{M,1}$ executed in C , restricted to the steps $0, \dots, i-1$. Now, let $t(i) = x$, that is, let x be the $t(i)$ -th element in the linear order \mathbb{D} . For each $0 \leq i \leq n$, we choose a minimal set $Out_i \subseteq D_{t(i)}$ that satisfies the condition: $\forall x' \in D_{t(i)} (\exists x'' \in Out_i (x' \sim x'') \vee x' \in [x]_{\sim} \cup [pre(x)]_{\sim} \cup \bigcup_{x'' \in ff_{\mathbf{L}}^{\bar{A}}(x)} [x'']_{\sim})$. The minimality constraint forces $Out_i \cap ([x]_{\sim} \cup [pre(x)]_{\sim} \cup \bigcup_{x'' \in ff_{\mathbf{L}}^{\bar{A}}(x)} [x'']_{\sim})$ to be empty, which amounts to say that the class of x , the class of the predecessor of x , and the classes of all the past far friends of x are not witnessed in Out_i .

We associate two sets \widehat{Out}_i and \widetilde{Out}_i , with $\widehat{Out}_i \cup \widetilde{Out}_i = Out_i$ and $\widehat{Out}_i \cap \widetilde{Out}_i = \emptyset$, with each step i and we guarantee the following three invariant conditions:

1. Let $P = ff_{\mathbf{L}}^{\bar{A}}(x) \setminus [x]_{\sim} = \{\bar{x}_1, \dots, \bar{x}_h\}$ (observe that $h < |\varphi|$). It holds that $R^i = \mathbf{req}(x)$, $\tilde{w}^i = \tilde{w}_{[x]_{\sim}}$, $t_{\tilde{w}^i} = t_{\tilde{w}_{[x]_{\sim}}}$, $f_{\sim}^i = f_{\sim}^x$, and $P_{R^i} = \{(\tilde{w}_{[\bar{x}_1]_{\sim}}, t_{\tilde{w}_{[\bar{x}_1]_{\sim}, x}}), \dots, (\tilde{w}_{[\bar{x}_h]_{\sim}}, t_{\tilde{w}_{[\bar{x}_h]_{\sim}, x}})\}$, if $x = \min(D)$ or $pre(x) \not\sim x$; $P_{R^i} = \{(\tilde{w}_{[\bar{x}_1]_{\sim}}, t_{\tilde{w}_{[\bar{x}_1]_{\sim}, x}}), \dots, (\tilde{w}_{[\bar{x}_h]_{\sim}}, t_{\tilde{w}_{[\bar{x}_h]_{\sim}, x}})\} \cup \{(\tilde{w}_{[pre(x)]_{\sim}}, t_{\tilde{w}_{[pre(x)]_{\sim}, x}})\}$ otherwise.
2. For all $x' \in \widehat{Out}_i$, it holds that $t_{\tilde{w}_{[x']_{\sim}, pre(x)}} \neq t_{\tilde{w}_{[x']_{\sim}, x}}$ and $k \leq k'$, where k' is the position of the pair $(\tilde{w}_{[x']_{\sim}}, t_{\tilde{w}_{[x']_{\sim}, pre(x)}})$ in the order $<_{R^i}$.
3. For all pairs $(\tilde{w}, t_{\tilde{w}})$, $v_i[(\tilde{w}, t_{\tilde{w}})] = |\{x' \in \widehat{Out}_i : (\tilde{w}^{x'}, t_{\tilde{w}_{[x']_{\sim}, pre(x)}}) = (\tilde{w}, t_{\tilde{w}})\}| + |\{x' \in \widetilde{Out}_i : (\tilde{w}^{x'}, t_{\tilde{w}_{[x']_{\sim}, x}}) = (\tilde{w}, t_{\tilde{w}})\}|$.

We define the initial configuration of the computation C as the pair $(q_0, \bar{0})$, where $q_0 = (\mathbf{req}(\min(D)), \tilde{w}_{[\min(D)]_{\sim}}, t_{\tilde{w}_{[\min(D)]_{\sim}, \min(D)}}, f_{\sim}^{\min(D)}, \emptyset, 1)$, and we initialize both \widehat{Out}_0 and \widetilde{Out}_0 to \emptyset . It can be easily checked that it satisfies the invariant conditions.

Let us consider now the step $i+1$. Let (q_i, v_i) , with $q_i = (R^i, \tilde{w}^i, t_{\tilde{w}^i}, P_{R^i}, f_{\sim}^i, k)$, be the current configuration, that satisfies the invariant conditions. We proceed as follows.

1. If $\widehat{Out}_i = \emptyset$, we directly move to the next step. Otherwise, let $x' \in \widehat{Out}_i$ be such that $(\tilde{w}_{[x']_{\sim}}, t_{\tilde{w}_{[x']_{\sim}, pre(x)}})$ is the smallest element in $\{(\tilde{w}_{[x'']_{\sim}}, t_{\tilde{w}_{[x'']_{\sim}, pre(x)}}) : x'' \in \widehat{Out}_i\}$ with respect to $<_{R^i}$. W.l.o.g., we can assume $(\tilde{w}_{[x']_{\sim}}, t_{\tilde{w}_{[x']_{\sim}, pre(x)}})$

to be the k -th element in the order $<_{R^i}$. As a matter of fact, if it was not the case, by condition 2 of the invariant, it would not be the k' -th element in the order, for some $k' < k$ (and thus, $k' > k$). We execute a number of transitions in $\Delta_{k,k+1}$ until we reach a state $q_{i'} = (R^i, \tilde{w}^i, t_{\tilde{w}^i}, P_{R^i}, f_{\sim}^i, k')$, which coincides with q_i apart from the replacement of k by k' . Since all these transitions do nothing, the invariant is trivially preserved.

Since $(\mathbf{L}, ff_{\mathbf{L}})$ is a saturated well-match, $t_{\tilde{w}_{[x']\sim, pre(x)}} \xrightarrow{R} t_{\tilde{w}_{[x']\sim, x}}$ and thus the transition $((R^i, \tilde{w}^i, t_{\tilde{w}^i}, P_{R^i}, f_{\sim}^i, k), \{c_{\tilde{w}_{[x']\sim, t_{\tilde{w}_{[x']\sim, pre(x)}}} - 1, c_{\tilde{w}_{[x']\sim, t_{\tilde{w}_{[x']\sim, x}} + 1}\}, (R^i, \tilde{w}^i, t_{\tilde{w}^i}, P_{R^i}, f_{\sim}^i, k))$ belongs $\Delta_{k,k}$. We execute such a transition. Condition 1 of the invariant is trivially satisfied by the $i + 1$ -th configuration. To satisfy conditions 2 and 3 of the invariant, it suffices to let $\widehat{Out}_{i+1} = \widehat{Out}_i \setminus \{x'\}$ and $\widehat{Out}_{i+1} = \widehat{Out}_i \cup \{x'\}$. If $\widehat{Out}_{i+1} \neq \emptyset$, we repeat this step; otherwise, ($\widehat{Out}_i = \emptyset$), we move to the next one.

2. If $t(i) = \max(D)$, we terminate the procedure. Otherwise, ($t(i) \neq \max(D)$), we proceed as follows. W.l.o.g., we can assume that $k = M$. If this was not case, as in step 1, we execute a number of transitions in $\Delta_{k,k+1}$ until we reach a state $q_{i'} = (R^i, \tilde{w}^i, t_{\tilde{w}^i}, P_{R^i}, f_{\sim}^i, M)$. Since all these transitions do nothing, the invariant is trivially preserved. Let $x = next(t(i))$.

We can execute a transition (q_i, U_i, q_{i+1}) , where $q_{i+1} = (\mathbf{req}(x), \tilde{w}_{[x]\sim} t_{\tilde{w}_{[x]\sim, x}}, P_{R^{i+1}}, f_{\sim}^i, 1)$, with $P_{R^{i+1}}(\tilde{w}, t_{\tilde{w}}) = |\{[x']\sim : \tilde{w} = \tilde{w}_{[x']\sim} \wedge t_{\tilde{w}} = t_{\tilde{w}_{[x']\sim, x}} \wedge x' \neq x \wedge (pre(x) \in [x']\sim \vee [x']\sim \cap ff_{\mathbf{L}}^{\bar{A}}(x) \neq \emptyset)\}|$ for each pair $(\tilde{w}, t_{\tilde{w}})$. By the definition of transitions in $\Delta_{M,1}$, the values of $R^i, \tilde{w}^i, t_{\tilde{w}^i}, P_{R^i}, R^{i+1}, \tilde{w}^{i+1}, t_{\tilde{w}^{i+1}}$, and $P_{R^{i+1}}$ univocally determine the update U . It is easy to check that condition 1 of the invariant is satisfied. In order to satisfy conditions 2 and 3 of the invariant, we put $\widehat{Out}_{i+1} = \{x' \in Out_{i+1} : t_{\tilde{w}_{[x']\sim, pre(x)}} = t_{\tilde{w}_{[x']\sim, x}}\}$ and $\widehat{Out}_{i+1} = Out_{i+1} \setminus \widehat{Out}_{i+1}$, where, by definition, Out_{i+1} is a minimal set that satisfies the condition $\forall x' \in D_{t(i+1)} (\exists x'' \in Out_{i+1} (x' \sim x'') \vee x' \in [x]\sim \cup [pre(x)]\sim \cup \bigcup_{x'' \in ff_{\mathbf{L}}^{\bar{A}}(x)} [x'']\sim)$. Finally, we increment i by 1 and we go back to step 1.

The procedure terminates in a configuration (q_i, v_i) , with $\widehat{Out}_i = \emptyset$ and $t(i) = \max(D)$. W.l.o.g., we can assume that $k = M$. Once more, if this was not the case, we execute a number of transitions in $\Delta_{k,k+1}$ until we reach a state $q_{i'} = (R^i, \tilde{w}^i, t_{\tilde{w}^i}, P_{R^i}, f_{\sim}^i, M)$. Since all these transitions do nothing, the invariant is trivially preserved. By condition 3 of the invariant, for each pair $(\tilde{w}, t_{\tilde{w}})$, it holds that $v_i[(\tilde{w}, t_{\tilde{w}})] = |\{[x]\sim : x \in D \wedge \tilde{w}_{[x]\sim} = \tilde{w} \wedge t_{\tilde{w}_{[x]\sim, max(D)}} = t_{\tilde{w}}\}|$. Since \mathbf{L} is fulfilling, for each $x \in D$, $(\tilde{w}_{[x]\sim}, t_{\tilde{w}_{[x]\sim, max(D)}})$ is final. The only components of v_i that can be greater than 0 are those relative to pairs (class word, temporary fulfilment) which are final. If there is any such pair, we execute suitable transitions in $\Delta_{M,M}^f$ until we reach the configuration $(q_i, \bar{0})$. \square